

TESIS DOCTORAL

AÑO 2019

LABORATORIO DUAL DE ROBÓTICA EDUCATIVA

Autor:

PEDRO PLAZA MERINO

INGENIERÍA INDUSTRIAL

TECNOLOGÍAS INDUSTRIALES

Director:

Dr. ELIO SAN CRISTÓBAL RUIZ

Codirector:

Dr. GERMÁN CARRO FERNÁNDEZ

TESIS DOCTORAL

AÑO 2019

LABORATORIO DUAL DE ROBÓTICA EDUCATIVA



Autor:

PEDRO PLAZA MERINO

INGENIERÍA INDUSTRIAL

TECNOLOGÍAS INDUSTRIALES

Director:

Dr. ELIO SAN CRISTÓBAL RUIZ

Codirector:

Dr. GERMÁN CARRO FERNÁNDEZ

ÍNDICE GENERAL

Índice general.....	v
Lista de símbolos, abreviaturas y siglas	xiii
Lista de tablas y figuras.....	xvii
Lista de tablas	xvii
Lista de figuras	xxiii
Agradecimientos	xxix
Resumen	xxx
Summary	xxxiii
Prefacio	xxxv
1. Introducción.....	1
1.1. Mejora de la calidad educativa de la ciencia y la tecnología en España	2
1.2. Aprendizaje e Internet	5
1.3. Internet of Things - IoT.....	6
1.4. Tipos de plataformas de desarrollo	7
1.5. Kits de desarrollo basados en FPGA.....	8
1.6. Comunicaciones Wireless en la educación	9
1.7. Innovaciones que aporta este trabajo	10
1.8. Organización de este trabajo	11
2. Estado del arte	15
2.1. La educación STEM.....	21
2.2. Robótica educativa como potenciador de STEM	24
2.3. Iniciativas y herramientas en torno a STEM.....	29
2.3.1. Arduino	29
2.3.2. Raspberry Pi	34
2.3.3. Scratch	36
2.3.4. Crumble.....	39
2.3.5. Otras herramientas en torno a STEM	43
2.4. Laboratorios remotos aplicados a STEM.....	47
2.5. Conclusiones	48
3. Plataformas de Desarrollo y su uso	53

3.1.	Plataformas para aplicaciones IoT basadas en Arduino.....	55
3.1.1.	Arduino WiFi Shield.....	55
3.1.2.	Genuino Yún Shield.....	57
3.1.3.	Arduino Yún Linino OS	58
3.1.4.	Genuino MKR1000.....	59
3.2.	NodeMCU ESP8266	61
3.3.	Onion OMEGA	62
3.4.	Particle PHOTON	63
3.5.	Comparativa de acuerdo con el hardware de procesamiento.....	64
3.6.	Comparativa de acuerdo con el sistema de alimentación	66
3.7.	Comparativa de acuerdo con la memoria disponible	67
3.8.	Comparativa de acuerdo con las comunicaciones	68
3.9.	Comparativa de acuerdo con el método de programación	71
3.10.	Comparativa de acuerdo con las dimensiones.....	72
3.11.	Comparativa de acuerdo con el coste.....	74
3.12.	Comparativa de acuerdo con el soporte de las comunidades	74
3.13.	Uso de las plataformas por la comunidad científica	75
3.14.	Plataformas de robótica educativa Wireless	76
3.15.	Conclusiones	81
4.	Escenarios educativos y pruebas con alumnos.....	83
4.1.	Actividades realizadas en 2016/2017	83
4.1.1.	Taller de Crumble.....	83
4.1.2.	Curso anual de introducción a la robótica 2016/2017	84
4.2.	Actividades realizadas en 2017/2018	89
4.2.1.	Taller Scratch Day 2017	89
4.2.2.	Taller Crumble Day 2017.....	91
4.2.3.	Taller de robótica educativa multi-plataforma 2017.....	93
4.2.4.	Talleres de robótica educativa multi-plataforma 2018	95
4.2.5.	Curso anual de introducción a la robótica 2017/2018	98
4.2.6.	Taller de robótica educativa para la integración y la inclusión EDUCON 2018	119
4.3.	Actividades realizadas en 2018/2019	120
4.3.1.	Resultados obtenidos en Academia PIXEL para Scratch básico	120

4.3.2.	Resultados obtenidos en Academia PIXEL para Crumble básico	122
5.	Arquitectura de la plataforma	127
5.1.	Herramienta Colaborativa Robótica Educativa	127
5.1.1.	Funcionalidades	129
5.2.	Metodologías de desarrollo	130
5.2.1.	Metodología del Desarrollo Rápido de Aplicaciones	131
5.2.2.	Metodología del Proceso Unificado de Desarrollo de Software.....	132
5.2.3.	Metodología de Kendall y Kendall	134
5.2.4.	Metodología Orientada a Objetos	136
5.2.5.	Metodología de Sistemas Blandos.....	138
5.2.6.	Metodología del Modelo en V	140
5.2.7.	Metodología Extreme Programming	142
5.2.8.	Metodología SCRUM.....	143
5.3.	Metodología de desarrollo de la plataforma	144
5.3.1.	Primera etapa: especificaciones de la plataforma.....	146
5.3.2.	Segunda etapa: diseño de la arquitectura	147
5.3.3.	Tercera etapa: descripción de los componentes	149
5.3.4.	Cuarta etapa: implementación	149
5.3.5.	Quinta etapa: pruebas de integración	150
5.3.6.	Sexta etapa: pruebas de sistema	150
5.3.7.	Séptima etapa: pruebas con alumnos	150
6.	Desarrollo hardware	151
6.1.	Metodología de desarrollo hardware	151
6.1.1.	Paso 1: especificaciones hardware	153
6.1.2.	Paso 2: elementos hardware	155
6.1.3.	Paso 3: diseño de los esquemáticos	158
6.1.4.	Paso 4: diseño físico del PCB.....	159
6.1.5.	Paso 5: archivos de fabricación.....	159
6.1.6.	Paso 6: ensamblado de la PBA.....	161
6.1.7.	Paso 7: pruebas de integridad	162
6.2.	Primer prototipo, versión 1.0.....	162
6.3.	Segundo prototipo, versión 1.1.....	170

6.4.	Tercer prototipo, versión 1.2	178
6.5.	Cuarto prototipo, versión 1.3.....	187
7.	Desarrollo firmware	201
7.1.	Metodología de desarrollo firmware	201
7.1.1.	Paso 1: especificaciones firmware	206
7.1.2.	Paso 2: bloques funcionales.....	210
7.1.3.	Paso 3: descripción firmware.....	215
7.1.4.	Paso 4: simulación	228
7.1.5.	Paso 5: integración en FPGA.....	236
7.1.6.	Paso 6: pruebas de integración hardware-firmware	245
7.2.	Primera versión de firmware, versión 1.....	246
7.3.	Segunda versión de firmware, versión 2.....	250
7.4.	Tercera versión de firmware, versión 3	254
8.	Desarrollo software	265
8.1.	Metodología de desarrollo software	265
8.1.1.	Paso 1: especificaciones software	266
8.1.2.	Paso 2: bloques funcionales.....	270
8.1.3.	Paso 3: codificación del software	270
8.1.4.	Paso 4: verificación del software	279
8.1.5.	Paso 6: pruebas de integración hardware-firmware-software.....	282
8.2.	Primera versión de software, versión 1.0	282
8.3.	Segunda versión de software, versión 1.1	283
9.	Pruebas con alumnos de la plataforma	287
9.1.	Pruebas de sistema de la plataforma.....	287
9.1.1.	Dron Educativo.....	288
9.1.2.	Prototipado creativo.....	289
9.1.3.	Rotación de luz con LED.....	289
9.1.4.	Red de comunicaciones LiFi	289
9.1.5.	Red de semáforos	289
9.1.6.	Control de parking	289
9.1.7.	Entorno Educativo con Smart Devices	289
9.1.8.	Entorno remoto con Internet	290

9.1.9.	Escenario para las pruebas de sistema	290
9.2.	Pruebas con alumnos	291
10.	Costes de los desarrollos	297
10.1.	Costes del desarrollo de la arquitectura	298
10.2.	Costes del desarrollo hardware	300
10.3.	Costes del desarrollo firmware	302
10.4.	Costes del desarrollo software.....	304
10.5.	Costes de las pruebas con alumnos	306
10.5.1.	Costes del curso anual de introducción a la robótica 2016/2017 – Módulo 1	306
10.5.2.	Costes del curso anual de introducción a la robótica 2016/2017 – Módulo 2	308
10.5.3.	Costes del curso anual de introducción a la robótica 2016/2017 – Módulo 3	309
10.5.4.	Taller Scratch Day 2017	311
10.5.5.	Taller Crumble Day 2017.....	312
10.5.6.	Taller de robótica educativa multi-plataforma 2017.....	313
10.5.7.	Curso anual de introducción a la robótica 2017/2018 – Scratch básico.....	314
10.5.8.	Curso anual de introducción a la robótica 2017/2018 – Scratch intermedio	316
10.5.9.	Curso anual de introducción a la robótica 2017/2018 – Scratch avanzado.....	317
10.5.10.	Curso anual de introducción a la robótica 2017/2018 – Crumble básico.....	317
10.5.11.	Talleres de robótica educativa multi-plataforma 2018	318
10.5.12.	Taller de robótica educativa multi-plataforma EDUCON 2018.....	319
10.6.	Costes totales de desarrollo de la herramienta de robótica educativa	320
10.7.	Conclusiones	321
11.	Lecciones aprendidas	325
11.1.	Seguimiento del avance de la investigación	326
11.1.1.	Descripción de las condiciones, causas o factores	327
11.1.2.	Descripción de la solución	328
11.1.3.	Análisis de los resultados.....	329
11.2.	Canales de comunicación.....	329
11.1.1.	Descripción de las condiciones, causas o factores	330
11.1.2.	Descripción de la solución	330
11.1.3.	Análisis de los resultados.....	330
11.3.	Arquitectura del sistema	330

11.3.1.	Descripción de las condiciones, causas o factores.	331
11.3.2.	Descripción de la solución	331
11.3.3.	Análisis de los resultados.....	331
11.4.	Desarrollo firmware	331
11.4.1.	Descripción de las condiciones, causas o factores.	332
11.4.2.	Descripción de la solución	332
11.4.3.	Análisis de los resultados.....	332
11.5.	Gestión de contenidos educativos.....	333
11.5.1.	Descripción de las condiciones, causas o factores.	333
11.5.2.	Descripción de la solución	333
11.5.3.	Análisis de los resultados.....	333
11.6.	Análisis de resultados en acciones educativas.....	334
11.6.1.	Descripción de las condiciones, causas o factores.	334
11.6.2.	Descripción de la solución	334
11.6.3.	Análisis de los resultados.....	334
11.7.	Gestión de marketing y publicidad	335
11.7.1.	Descripción de las condiciones, causas o factores.	335
11.7.2.	Descripción de la solución	335
11.7.3.	Análisis de los resultados.....	336
12.	Conclusiones y líneas de investigación futuras	337
12.1.	Conclusiones y aportaciones de este trabajo.....	338
12.2.	Publicaciones realizadas durante el trabajo de investigación	340
12.3.	Desarrollos y aplicaciones futuras.....	343
12.4.	Observaciones finales	344
13.	Currículum Vitae.....	347
13.1.	Datos personales.....	347
13.2.	Extracto:	347
13.3.	Experiencia laboral:.....	347
13.4.	Educación:	348
13.5.	Habilidades.....	349
14.	Bibliografía.....	351
Anexo 1:	Contenido del DVD-ROM	385

Anexo 1.1. Diseño hardware.....	385
Anexo 1.2. Diseño firmware	385
Anexo 1.3. Diseño software	388
Anexo 1.4. Material educativo.....	388
Anexo 2: Otros méritos.....	389
Anexo 2.1. Colaboración en conferencias internacionales.....	389
Anexo 2.2. Colaboración en revistas científicas internacionales.....	391
Anexo 2.3. Cargos en la Rama de Estudiantes del IEEE de la UNED	392
Anexo 2.4. Colaboración con grupos de afinidad del IEEE.....	392
Anexo 2.5. Promoción y divulgación mediante la impartición de cursos y talleres	392

LISTA DE SÍMBOLOS, ABREVIATURAS Y SIGLAS

ADOO	Análisis y Diseño Orientado a Objetos
AFO	Ortesis Activa de Tobillo y Pie
AOO	Análisis Orientado a Objetos
ARM	Advanced Risc Machine
BEMS	Building Energy Management System
BLE	Bluetooth Low Energy
BOYD	Bring Your Own Device
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CASE	Computer Aided Software Engineering
CDR	Comité de las Regiones Europeo
CDTI	Centro para el Desarrollo Tecnológico Industrial
CI	Computational Intelligence
CLI	Command-Line Interface
CWI	Centrum Wiskunde & Informatica
DC	Direct Current
DDF	Detección de Fallos y Diagnóstico
DGBL	Digital Games Based Learning
DIECTQAI	Departamento de Ingeniería Eléctrica, Electrónica, Control, Telemática y Química Aplicada a la Ingeniería
DIY	Do It Yourself
DRC	Design Rule Check
EDA	Electronic Design Automation
EDUCON	Global Engineering Education Conference
ERF	European Robotics Forum
ERW	European Robotics Week
FIE	Frontiers in Education
FPGA	Field Programmable Gate Array
FTL	Flexible Thinking in Learning
FW	Firmware
GPIO	General Purpose Input Output
GPU	Graphics Processing Unit
GUI	Graphic User Interface
H2H	Human-to-Human
HDL	Hardware Description Language
HDMI	High-Definition Multimedia Interface
HIPO	Hierarchy-Input-Process-Output
HW	Hardware
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
IC	Integrated Circuit

ICL	Conference on Interactive Collaborative Learning
ICPS	Industrial Cyberphysical Systems
ICSP	In-Circuit Serial Programming
IDE	Interface Development Environment
IDII	Instituto de Diseño de Interacción Ivrea
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IoET	Internet of Educational Things
IoT	Internet of Things
IoTDS	IoT Devices
IP	Intelectual Property
ISCED	International Standard Classification of Education
IVA	Impuesto sobre el Valor Añadido
JAD	Joint Application Design
LAMP	Linux, Apache, MySQL y PHP
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitter Diode
Li-Fi	Light Fidelity
LiPo	Lithium Polymer
LOE	Ley Orgánica de Educación
LOMCE	Ley Orgánica para la Mejora de la Calidad Educativa
LPM	Library Parameters Modules
LWMOOCS	Learning with MOOCS
MCU	Microcontroller Unit
MIPS	Microprocessor without Interlocked Pipeline Stages
MISO	Multiple-Input Single-Output
MIT	Massachusetts Institute of Technology
MOOC	Massive Open Online Course
MOOCS	Massive Open Online Courses
MOSI	Master Out Slave In
MRP	Multi-Role Project
MSA	Modelo Sistema de Actividades
OS	Operating System
PBA	Printed Board Assembly
PBL	Project Based Learning
PC	Personal Computer
PCB	Printed Circuit Board
PIXEL	Pi Improved Xwindows Environment, Lightweight
POO	Programación Orientada a Objetos
POVDs	Power Oriented Device
PVP	Precio de Venta al Público
PWM	Pulse Width Modulation
QSPI	Quad Serial Peripheral Interface

RAD	Rapid Application Development
RAM	Random-Access Memory
REV	Remote Engineering and Virtual Instrumentation
RGB	Red, Green and Blue
ROI	Return of Investment
RTC	Real Time Clock
RTOS	Real Time Operating System
RUP	Rational Unified Process
SBC	System Board Computer
SCE	Siemens Automation Cooperates with Education
SDC	Synopsys Design Constraint
SDK	Software Development Kit
SDLC	Systems Development Life Cycle
SEIDI	Secretaría de Estado de Investigación, Desarrollo e Innovación
SETSI	Secretaría de Estado para las Telecomunicaciones y para la Sociedad de la Información
SiPM	Silicon Photomultiplier
SMD	Surface Mount Device
SoC	System on Chip
SPI	Serial Peripheral Interface
SQA	Signal Quality-Aware
SRAM	Static Random-Access Memory
SSL	Secure Sockets Layer
SSM	Soft Systems Methodology
STEAM	STEM + the Arts
STEM	Science, Technology, Engineering and Mathemathis
SW	Software
TAAE	Technology, Learning and Teaching of Electronics
TALE	International Conference on Teaching, Assessment and Learning for Engineering
TEL	Technology Enhanced Learning
TFT	Thin Film Transistor
TIC	Tecnologías de la Información y la Comunicación
UART	Universal Asynchronous Receiver/Transmitter
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UMI	Ubiquitous Computing, Mobile Computing and the Internet of Things
UML	Unified Modeling Language
UNED	Universidad Nacional de Educación a Distancia
USB	Universal Serial Bus
VESLL	Laboratorio Virtual de Aprendizaje de Ciencias de Ingeniería
VGA	Video Graphics Array
WICED	Wireless Internet Connectivity for Embedded Devices
WIE	Women in Engineering

XP Extreme Programming
YP Young Professionals

LISTA DE TABLAS Y FIGURAS

Lista de tablas

Tabla 1.	Dispositivo de procesamiento de tarjeta de desarrollo IoT.....	65
Tabla 2.	Alimentación y puertos de interfaz de tarjeta de desarrollo IoT.....	66
Tabla 3.	Recursos de memoria volátil y memoria no volátil de tarjeta de desarrollo IoT.....	67
Tabla 4.	Puertos serie y puertos discretos de tarjeta de desarrollo IoT.....	69
Tabla 5.	Herramientas y lenguajes de programación de tarjeta de desarrollo IoT.....	71
Tabla 6.	Dimensiones de tarjeta de desarrollo IoT.....	73
Tabla 7.	Coste de tarjeta de desarrollo IoT.....	74
Tabla 8.	Nivel académico de los estudiantes.....	85
Tabla 9.	Nivel académico de los estudiantes.....	90
Tabla 10.	Nivel académico de los estudiantes.....	92
Tabla 11.	Nivel académico de los estudiantes.....	94
Tabla 12.	Nivel académico de los estudiantes.....	95
Tabla 13.	Respuestas del pretest.....	96
Tabla 14.	Resultados de las actividades de las sesiones de la segunda parte.....	96
Tabla 15.	Resultados de las actividades de las sesiones de la tercera parte.....	96
Tabla 16.	Resultados de las actividades de las sesiones de la cuarta parte.....	97
Tabla 17.	Opinión de los asistentes acerca del taller.....	97
Tabla 18.	Respuestas del post-test.....	98
Tabla 19.	Nivel académico de los estudiantes.....	109
Tabla 20.	Respuestas del pretest.....	109
Tabla 21.	Resultados de las actividades de las sesiones de la primera parte.....	109
Tabla 22.	Resultados de las actividades de las sesiones de la segunda parte.....	110
Tabla 23.	Resultados de las actividades de las sesiones de la tercera parte.....	110
Tabla 24.	Resultados de las actividades de las sesiones de la cuarta parte.....	110
Tabla 25.	Opinión de los asistentes acerca del curso.....	111
Tabla 26.	Respuestas del post-test.....	111
Tabla 27.	Respuestas del pretest.....	112
Tabla 28.	Resultados de las actividades de las sesiones de la primera parte.....	112
Tabla 29.	Resultados de las actividades de las sesiones de la segunda parte.....	112

Tabla 30.	Resultados de las actividades de las sesiones de la tercera parte.....	113
Tabla 31.	Resultados de las actividades de las sesiones de la cuarta parte.....	113
Tabla 32.	Opinión de los asistentes acerca del curso.....	113
Tabla 33.	Respuestas del post-test.....	114
Tabla 34.	Nivel académico de los estudiantes.....	114
Tabla 35.	Respuestas del pretest.....	114
Tabla 36.	Resultados de las actividades de las sesiones de la primera parte.....	115
Tabla 37.	Resultados de las actividades de las sesiones de la segunda parte.....	115
Tabla 38.	Resultados de las actividades de las sesiones de la tercera parte.....	115
Tabla 39.	Resultados de las actividades de las sesiones de la cuarta parte.....	115
Tabla 40.	Opinión de los asistentes acerca del curso.....	116
Tabla 41.	Respuestas del post-test.....	116
Tabla 42.	Nivel académico de los estudiantes.....	116
Tabla 43.	Respuestas del pretest.....	117
Tabla 44.	Resultados de las actividades de las sesiones de la primera parte.....	117
Tabla 45.	Resultados de las actividades de las sesiones de la segunda parte.....	117
Tabla 46.	Resultados de las actividades de las sesiones de la tercera parte.....	118
Tabla 47.	Resultados de las actividades de las sesiones de la cuarta parte.....	118
Tabla 48.	Opinión de los asistentes acerca del curso.....	118
Tabla 49.	Respuestas del post-test.....	119
Tabla 50.	Nivel académico de los estudiantes.....	120
Tabla 51.	Respuestas del pretest.....	120
Tabla 52.	Resultados de las actividades de las sesiones de la primera parte.....	121
Tabla 53.	Resultados de las actividades de las sesiones de la segunda parte.....	121
Tabla 54.	Resultados de las actividades de las sesiones de la tercera parte.....	121
Tabla 55.	Resultados de las actividades de las sesiones de la cuarta parte.....	121
Tabla 56.	Opinión de los asistentes acerca del curso.....	122
Tabla 57.	Respuestas del post-test.....	122
Tabla 58.	Respuestas del pretest.....	123
Tabla 59.	Resultados de las actividades de las sesiones de la primera parte.....	123
Tabla 60.	Resultados de las actividades de las sesiones de la segunda parte.....	123
Tabla 61.	Resultados de las actividades de las sesiones de la tercera parte.....	124

Tabla 62.	Resultados de las actividades de las sesiones de la cuarta parte.	124
Tabla 63.	Opinión de los asistentes acerca del curso.	124
Tabla 64.	Respuestas del post-test.	125
Tabla 65.	Resumen de las especificaciones de la plataforma.....	146
Tabla 66.	Relación entre las señales del módulo principal y el conector P_MM.	153
Tabla 67.	Relación entre las E/S de la FPGA y el conector P2.....	156
Tabla 68.	Relación entre las E/S de la FPGA y el conector P2.....	156
Tabla 69.	Relación entre las E/S de la FPGA y el conector P3.....	156
Tabla 70.	Relación entre las E/S de la FPGA y el conector P4.....	156
Tabla 71.	Relación entre las E/S de Arduino y el conector de puertos digitales 1.	157
Tabla 72.	Relación entre las E/S de Arduino y el conector de puertos digitales 2.	157
Tabla 73.	Relación entre las E/S de Arduino y el conector de puertos analógicos.....	157
Tabla 74.	Relación entre las E/S de Arduino y el conector de alimentación.	157
Tabla 75.	Relación entre las E/S de Arduino y el conector BT_COM.....	158
Tabla 76.	Relación entre las señales del módulo principal y el conector P_MM.	189
Tabla 77.	Puertos de la FPGA conectados al módulo Bluetooth.	206
Tabla 78.	Puertos de la FPGA conectados al Shield de Arduino.	207
Tabla 79.	Puertos de la FPGA conectados al módulo secundario 1.....	207
Tabla 80.	Puertos de la FPGA conectados al módulo secundario 2.....	208
Tabla 81.	Puertos de la FPGA conectados al módulo secundario 3.....	208
Tabla 82.	Puertos de la FPGA conectados al módulo secundario 4.....	209
Tabla 83.	Puertos de la funcionalidad UART Bluetooth.....	211
Tabla 84.	Puertos de la funcionalidad UART Shield.....	212
Tabla 85.	Puertos de la comunicación discreta Shield.....	212
Tabla 86.	Puertos de la funcionalidad UART Arduino.....	213
Tabla 87.	Puertos de la monitorización PWM.	213
Tabla 88.	Puertos de la comunicación discreta Arduino.	214
Tabla 89.	Señales de entrada/salida del bloque MAIN_PROCESSOR.	219
Tabla 90.	Señales de entrada/salida del bloque UART_BLUETOOTH.	221
Tabla 91.	Señales de entrada/salida del bloque UART_SHIELD.....	222
Tabla 92.	Señales de entrada/salida del bloque DISCRETE_COMM_SHIELD.....	222
Tabla 93.	Señales de entrada/salida del bloque UART_ARDUINO.	224

Tabla 94.	Señales de entrada/salida del bloque PWM_MONITORING_ARDUINO.....	225
Tabla 95.	Señales de entrada/salida del bloque DISCRETE_COMM_ARDUINO.	226
Tabla 96.	Señales de entrada/salida del bloque UART_RX_MACHINE.	227
Tabla 97.	Señales de entrada/salida del bloque UART_TX_MACHINE.	227
Tabla 98.	Resumen de uso de recursos de Análisis y Síntesis.	236
Tabla 99.	Resumen de uso de recursos por entidad de Análisis y Síntesis.....	237
Tabla 100.	Relación entre los puertos E/S del FW y pines de la FPGA para el grupo de HWE. .	238
Tabla 101.	Relación entre los puertos E/S del FW y pines FPGA para el Shield.	238
Tabla 102.	Relación entre los puertos E/S del FW y pines FPGA para Arduino 1.....	239
Tabla 103.	Relación entre los puertos E/S del FW y pines FPGA para Arduino 2.....	240
Tabla 104.	Relación entre los puertos E/S del FW y pines FPGA para Arduino 3.....	241
Tabla 105.	Relación entre los puertos E/S del FW y pines FPGA para Arduino 4.....	242
Tabla 106.	Relación entre los puertos E/S del FW y pines FPGA para No Usado.	243
Tabla 107.	Resumen de uso de recursos de Análisis y Síntesis, versión 1 de firmware.	244
Tabla 108.	Uso de recursos por entidad de Análisis y Síntesis, versión 1 de firmware.....	245
Tabla 109.	Valores de configuración de velocidad de transmisión del módulo HC-06.	248
Tabla 110.	Relación entre los puertos E/S del FW y pines FPGA para Arduino 2.....	251
Tabla 111.	Relación entre los puertos E/S del FW y pines FPGA para No Usado.	252
Tabla 112.	Resumen de uso de recursos de Análisis y Síntesis, versión 2 de firmware.	252
Tabla 113.	Uso de recursos por entidad de Análisis y Síntesis, versión 2 de firmware.....	253
Tabla 114.	Puertos de la FPGA conectados al módulo secundario 1.	255
Tabla 115.	Puertos de la FPGA conectados al módulo secundario 2.	255
Tabla 116.	Puertos de la FPGA conectados al módulo secundario 3.	255
Tabla 117.	Puertos de la FPGA conectados al módulo secundario 4.	256
Tabla 118.	Señales de entrada/salida del bloque MAIN_PROCESSOR, versión 3 de firmware.	258
Tabla 119.	Resumen de uso de recursos de Análisis y Síntesis, versión 3 de firmware.	261
Tabla 120.	Uso de recursos por entidad de Análisis y Síntesis, versión 3 de firmware.....	262
Tabla 121.	Puertos de comunicación serie de las Arduino.....	267
Tabla 122.	Puertos PWM de las Arduino.....	267
Tabla 123.	Puertos de la comunicación discreta Arduino.	267
Tabla 124.	Constantes para definir las salidas PWM.....	271
Tabla 125.	Constantes relacionadas con la generación de los pulsos PWM.	271

Tabla 126.	Variables de propósito general.....	271
Tabla 127.	Constantes para definir las entradas/salidas discretas.	271
Tabla 128.	Grados asignados a cada uno de los puertos PWM de la Arduino 1.	272
Tabla 129.	Valores de los puertos discretos en función del tiempo de la Arduino 1.	273
Tabla 130.	Grados asignados a cada uno de los puertos PWM de la Arduino 2.	274
Tabla 131.	Valores de los puertos discretos en función del tiempo de la Arduino 2.	275
Tabla 132.	Grados asignados a cada uno de los puertos PWM de la Arduino 3.	275
Tabla 133.	Valores de los puertos discretos en función del tiempo de la Arduino 3.	276
Tabla 134.	Grados asignados a cada uno de los puertos PWM de la Arduino 4.	277
Tabla 135.	Valores de los puertos discretos en función del tiempo de la Arduino 4.	278
Tabla 136.	Conexiones entre la Arduino y el LED RGB.	279
Tabla 137.	Secuencia de iluminación del LED RGB durante la prueba de verificación.....	280
Tabla 138.	Secuencia de giro del servo durante la prueba de verificación para la Arduino 1.	280
Tabla 139.	Secuencia de giro del servo durante la prueba de verificación para la Arduino 2.	280
Tabla 140.	Secuencia de giro del servo durante la prueba de verificación para la Arduino 3.	281
Tabla 141.	Secuencia de giro del servo durante la prueba de verificación para la Arduino 4.	281
Tabla 142.	Nivel académico de los estudiantes.....	292
Tabla 143.	Respuestas del pretest.....	293
Tabla 144.	Resultados de las actividades de las sesiones de la primera parte.	294
Tabla 145.	Resultados de las actividades de las sesiones de la segunda parte.....	294
Tabla 146.	Opinión de los asistentes acerca del curso.....	295
Tabla 147.	Respuestas del post-test.....	295
Tabla 148.	Listado de los costes del desarrollo de la arquitectura.	300
Tabla 149.	Listado de los costes del desarrollo hardware.....	302
Tabla 150.	Listado de los costes del desarrollo firmware.	304
Tabla 151.	Listado de los costes del desarrollo software.....	305
Tabla 152.	Listado de los costes del módulo de Scratch 2017.	308
Tabla 153.	Listado de los costes del módulo de Crumble 2017.	309
Tabla 154.	Listado de los costes del módulo de Arduino 2017.	311
Tabla 155.	Listado de los costes del taller Scratch Day 2017.	312
Tabla 156.	Listado de los costes del taller Crumble Day 2017.	313
Tabla 157.	Listado de los costes del taller de robótica educativa multi-plataforma 2017.....	314

Tabla 158.	Listado de los costes del módulo de Scratch básico.	316
Tabla 159.	Listado de los costes del módulo de Scratch intermedio.	316
Tabla 160.	Listado de los costes del módulo de Scratch básico.	318
Tabla 161.	Listado de los costes de los talleres de robótica educativa 2018.	319
Tabla 162.	Listado de los costes del taller de robótica educativa EDUCON.	320
Tabla 163.	Resumen de los costes totales de desarrollo de la herramienta.	321
Tabla 164.	Costes totales de desarrollo de la plataforma y de las actividades formativas.	321
Tabla 165.	PVP y número de unidades de la plataforma para cubrir los costes.	323
Tabla 166.	Revisiones en conferencias internacionales para el año 2017.	389
Tabla 167.	Revisiones en conferencias internacionales para el año 2018.	390
Tabla 168.	Revisiones en conferencias internacionales para el año 2019.	390

Lista de figuras

Figura 1.	Puertos de la plataforma Arduino Nano.	32
Figura 2.	Ejemplo de programa con el IDE de Arduino.	33
Figura 3.	Ejemplo de robot con Arduino.	34
Figura 4.	Raspberry Pi 3 Modelo B.	35
Figura 5.	Importar proyectos en Scratch.	37
Figura 6.	Entorno y paneles de Scratch.	38
Figura 7.	Puertos de la tarjeta Crumble.	40
Figura 8.	Ejemplo de programa con Crumble.	41
Figura 9.	Ejemplo de robot con Crumble.	42
Figura 10.	Shield WiFi de Arduino.	56
Figura 11.	Genuino Yún Shield.	57
Figura 12.	Arduino Yún Linino OS.	59
Figura 13.	Genuino MKR1000.	60
Figura 14.	NodeMCU ESP8266.	62
Figura 15.	Onion OMEGA.	63
Figura 16.	Particle Photon.	64
Figura 17.	Plataformas Shield Yún de Arduino y NodeMCU.	66
Figura 18.	El Shield de Arduino Yún controla un robot de 4 ruedas.	73
Figura 19.	NodeMCU controlando un brazo robot de 6 grados de libertad.	73
Figura 20.	Kit Bluetooth DFRobotShop Rover v2.	78
Figura 21.	Kit de iniciación a la robótica Makeblock.	79
Figura 22.	Makeblock mBot – Kit robótico STEM para niños.	79
Figura 23.	Robot móvil autónomo con red Wireless.	79
Figura 24.	Conexión Bluetooth LEGO MIDSTORMS.	80
Figura 25.	Robot de cuatro ruedas basado en NI myRIO.	81
Figura 26.	Kit robótico basado en Crumble.	84
Figura 27.	Ejemplo de robot con Crumble.	86
Figura 28.	Ejemplo de robot con Arduino.	88
Figura 29.	Configuración del aula del taller.	90
Figura 30.	Ejemplo de kit robótico con Crumble.	93

Figura 31.	Matriz de contenidos del curso.....	99
Figura 32.	Transiciones para cada módulo.	108
Figura 33.	Diagrama de bloques de la plataforma y sus periféricos.....	128
Figura 34.	Herramienta Robótica Educativa.	129
Figura 35.	Diagrama de Modelo en V.	141
Figura 36.	Modelo en V del diseño de la plataforma.....	146
Figura 37.	Componentes de la plataforma a desarrollar.	147
Figura 38.	Arquitectura de la plataforma.	148
Figura 39.	Comunicaciones de la plataforma.....	149
Figura 40.	Shields Arduino apilados.....	154
Figura 41.	Soldador de estaño.	162
Figura 42.	Esquemáticos de la versión 1.0 del módulo principal.....	163
Figura 43.	Esquemáticos de la versión 1.0 del módulo secundario.....	164
Figura 44.	Layout de la versión 1.0 del módulo principal.	165
Figura 45.	Layout de la versión 1.0 del módulo secundario.	165
Figura 46.	Versión 1.0 del PCB del módulo principal.....	167
Figura 47.	Versión 1.0 del PCB del módulo secundario.	167
Figura 48.	Versión 1.0 del módulo principal.	168
Figura 49.	Versión 1.0 del módulo secundario.	169
Figura 50.	Diagrama de bloques del primer cambio hardware.	170
Figura 51.	Layout de la versión 1.1 del módulo principal.	171
Figura 52.	Layout de la versión 1.1 del módulo secundario.	171
Figura 53.	Versión 1.1 del PCB del módulo principal.	173
Figura 54.	Versiones 1.0 (arriba) y 1.1 (abajo) del PCB del módulo principal.....	173
Figura 55.	Versión 1.1 del PCB del módulo secundario.	174
Figura 56.	Versiones 1.0 (arriba) y 1.1 (abajo) del PCB del módulo secundario.....	174
Figura 57.	Versión 1.1 del módulo principal.	175
Figura 58.	Versiones 1.0 (arriba) y 1.1 (abajo) de la PBA del módulo principal.....	175
Figura 59.	Versión 1.1 del módulo secundario.	176
Figura 60.	Versiones 1.1 (arriba) y 1.0 (abajo) de la PBA del módulo secundario.....	176
Figura 61.	Versión 1.1 de la Plataforma.....	177
Figura 62.	Diagrama de bloques del segundo cambio hardware.	178

Figura 63.	Esquemáticos de la versión 1.2 del módulo principal.....	180
Figura 64.	Layout de la versión 1.2 del módulo principal.	181
Figura 65.	Layout de la versión 1.2 del módulo secundario.	181
Figura 66.	Versión 1.2 del PCB del módulo principal.....	182
Figura 67.	Versiones 1.1 (arriba) y 1.2 (abajo) del PCB del módulo principal.....	183
Figura 68.	Versión 1.2 del PCB del módulo secundario.	183
Figura 69.	Versiones 1.1 (arriba) y 1.2 (abajo) del PCB del módulo secundario.....	184
Figura 70.	Versión 1.2 del módulo principal.	184
Figura 71.	Versiones 1.1 (arriba) y 1.2 (abajo) de la PBA del módulo principal.....	185
Figura 72.	Versión 1.2 del módulo secundario.	185
Figura 73.	Versiones 1.1 (arriba) y 1.2 (abajo) de la PBA del módulo secundario.....	186
Figura 74.	Versión 1.2 de la Plataforma.....	187
Figura 75.	Diagrama de bloques del tercer cambio hardware.	188
Figura 76.	Esquemáticos de la versión 1.3 del módulo principal.....	189
Figura 77.	Esquemáticos de la versión 1.3 del módulo principal.....	190
Figura 78.	Esquemáticos de la versión 1.3 del módulo secundario.....	191
Figura 79.	Layout de la versión 1.3 del módulo principal.	192
Figura 80.	Layout de la versión 1.3 del módulo secundario.	192
Figura 81.	Versión 1.3 del PCB del módulo principal.....	193
Figura 82.	Versiones 1.2 (arriba) y 1.3 (abajo) del PCB del módulo principal.....	194
Figura 83.	Versión 1.3 del PCB del módulo secundario.	194
Figura 84.	Versiones 1.2 (arriba) y 1.3 (abajo) del PCB del módulo secundario.....	195
Figura 85.	Estación de soldadura.	196
Figura 86.	Microscopio electrónico.....	196
Figura 87.	Resultado de la soldadura de los componentes ST2378ETTR.....	196
Figura 88.	Versión 1.3 del módulo principal.	197
Figura 89.	Versiones 1.2 (arriba) y 1.3 (abajo) de la PBA del módulo principal.....	197
Figura 90.	Versión 1.3 del módulo secundario.	198
Figura 91.	Versiones 1.2 (arriba) y 1.3 (abajo) de la PBA del módulo secundario.....	198
Figura 92.	Versión 1.3 de la Plataforma.....	199
Figura 93.	Flujo de verificación con Modelsim.	202
Figura 94.	Flujo de diseño usando QUARTUS II.	204

Figura 95.	QUARTUS II web edition 13.0SP1.....	205
Figura 96.	Conexiones entre los diferentes componentes hardware.....	210
Figura 97.	Diagrama de bloques de los bloques funcionales.....	211
Figura 98.	Diseño jerárquico Top-Down.	216
Figura 99.	Diagrama de bloques del sistema completo.....	217
Figura 100.	Diagrama de secuencia para la comunicación con UART_BLUETOOTH.	218
Figura 101.	Verificación jerárquica Down-Top.	228
Figura 102.	Diagrama de flujo del proceso de verificación.....	229
Figura 103.	Carga del firmware en la FPGA.	247
Figura 104.	LED de estado de la FPGA.	247
Figura 105.	Integración del módulo Bluetooth.....	248
Figura 106.	Código Arduino para la configuración del módulo Bluetooth.	249
Figura 107.	Captura de Windows de detección del módulo Bluetooth.....	249
Figura 108.	Respuesta del módulo Bluetooth cuando se le envía un caracter.	250
Figura 109.	Diagrama de bloques del primer cambio firmware.....	251
Figura 110.	Diagrama de bloques del segundo cambio firmware.	254
Figura 111.	Bloques funcionales de la versión 3 de firmware.	256
Figura 112.	Diseño jerárquico Top-Down de la versión 3 de firmware.	257
Figura 113.	Diagrama de bloques del sistema completo de la versión 3 de firmware.....	258
Figura 114.	Verificación jerárquica Down-Top de la versión 3 de firmware.	259
Figura 115.	Interfaz de desarrollo de Arduino.....	266
Figura 116.	Esquema de funcionalidades de las Arduino.	269
Figura 117.	Diagrama de flujo de la comunicación con señales discretas.....	270
Figura 118.	Diagrama de flujo PWM del hilo de ejecución de la Arduino 1.	273
Figura 119.	Respuesta temporal de señales discretas del hilo de ejecución de la Arduino 1.	273
Figura 120.	Diagrama de flujo PWM del hilo de ejecución de la Arduino 2.	274
Figura 121.	Diagrama de flujo PWM del hilo de ejecución de la Arduino 3.	276
Figura 122.	Diagrama de flujo PWM del hilo de ejecución de la Arduino 4.	278
Figura 123.	Diagrama de bloques del primer cambio software.	284
Figura 124.	Diagrama de bloques de pruebas de sistema.....	287
Figura 125.	Robot de 4 ruedas y brazo manipulador para fines educativos.	288
Figura 126.	Dron Educativo.....	288

Figura 127.	Montaje de la plataforma para pruebas de sistema.....	291
Figura 128.	Diagrama de bloques de pruebas con alumnos.....	292
Figura 129.	Montaje de la plataforma con solución semafórica.	294

AGRADECIMIENTOS

Para comenzar debo expresar mi más profundo agradecimiento a mi director de Tesis, el profesor Dr. Elio San Cristóbal Ruiz, a mi Codirector de Tesis, el Dr. Germán Carro Fernández, y a nuestro asesor incondicional e inagotable, el profesor Dr. Manuel-Alonso Castro Gil, sin los cuales la culminación de este trabajo no hubiera sido posible. Mi realización científica se debe en gran medida a su apoyo incondicional y su capacidad para motivarme en los momentos difíciles.

También doy gracias a mi familia (esposa, padres, hermana, sobrina, suegros, cuñados, sobrinos políticos, tíos, primos y amigos cercanos), que ha soportado mis agobios y mi exceso de trabajo acumulado durante estos años de realización de esta tesis. En especial, a mi compañera, amiga y esposa, Mónica, que verá por fin que pasamos bastante más tiempo juntos disfrutando de nuestra compañía en un mayor grado. A mi cuñado y prácticamente hermano Javi, y sus dos hijos Alejandro y César que han sido un apoyo imprescindible a lo largo de esta aventura robótico-educativa. No puedo dejar de sin mencionar a mi suegro, Paco, sin tus mecanizados “caseros” y tus opiniones de montajes no sé lo que habríamos hecho en los talleres y cursos.

Otra figura importante para mí y para esta tesis es mi abuela que, a pesar de no comprender lo que yo tenía entre manos, siempre me ha animado a terminar esta tesis doctoral, casi desde los primeros meses de haber empezado este trabajo de investigación. Gracias Manolita, siempre te llevo en mi corazón, te recuerdo y te quiero, aunque ya no estés entre nosotros.

A mis amigos en general. Quizás debería enumeraros, pero sois tantos los que habéis esperado que llegue este momento, que no quiero correr el riesgo de olvidar el nombre de nadie.

Además, quiero agradecer uno por uno a mis compañeros y colaboradores de la UNED por sus aportaciones a este trabajo. Gracias Manuel Blázquez, Clara Pérez, Elena Ruiz, Antonio Menacho y Félix García-Loro.

Para finalizar, pero no por ello menos importante, agradecer a Manuel Ballesteros, Julia Merino, Aruquia Peixoto, Ramón Carrasco y Jesús Fraile Ardanuy por dejarme ser parte de vuestras vidas en forma de colaboraciones relacionadas con el IEEE, Jóvenes Profesionales y Mujer en la Ingeniería.

RESUMEN

En las últimas décadas la industria ha ido evolucionando a lo largo de varias revoluciones. La última iteración de la industria apareció hace algunos años cuando la robótica fue incluida como parte de la industria. La educación ha sido influenciada por las diferentes revoluciones industriales, pero en estos momentos la necesidad de la industria requiere que haya un cambio en el perfil de los estudiantes. A corto y medio plazo el tejido industrial demanda una gran cantidad de profesionales con conocimientos en las áreas de la ciencia, la tecnología, la ingeniería y las matemáticas, las denominadas STEM (Science, Technology, Engineering and Mathematics). Por otro lado, la robótica educativa surge como una prometedora impulsora de la educación STEM. En la actualidad hay multitud de soluciones para aplicar robótica educativa en contextos STEM, pero los contenidos educativos no están muy desarrollados o bien, no facilitan la colaboración entre los estudiantes. Asimismo, hay soluciones que no son abiertas lo que dificulta su uso o la flexibilidad para desarrollar diferentes tipos de aplicaciones robóticas en contextos educativos. De ahí que se haya elegido dispositivo lógico programable como el núcleo de la plataforma que se ha desarrollado como parte de esta tesis doctoral. Los dispositivos lógicos programables proporcionan flexibilidad durante el ciclo de diseño porque las iteraciones de diseño son simplemente una cuestión de cambiar el archivo de programación. Además, se pueden reprogramar incluso mientras están trabajando. También, la lógica programable permite el procesamiento paralelo que puede utilizarse para supervisar tareas o evitar problemas de comunicación como son los casos de retardos indeseados por cuello de botella. Por otro lado, Arduino permite una plataforma de prototipado fácil y rápida. Asimismo, es una plataforma de software y hardware extensible y de código abierto. Este trabajo describe la combinación de poderosos recursos en una única herramienta. La Herramienta Colaborativa para Robótica Educativa (HeCRE) combina hardware, firmware y software abierto totalmente integrados en la misma plataforma. De este modo, se garantiza que las modificaciones se puedan aplicar fácilmente en la plataforma con el objetivo de adaptarla al programa de aprendizaje o a las sesiones de aprendizaje que lo requieran. También se incluye una metodología completa de desarrollo, las líneas de trabajo futuro y cómo se planea utilizarlas. La propuesta lleva la robótica al proceso de aprendizaje, desde simples objetos de aprendizaje hasta proyectos complejos. Escalabilidad, modularidad, posibilidades de reconfiguración y compatibilidad son las principales características de su diseño. De esta manera, la plataforma se incluye como parte de los escenarios educativos dentro del contexto de la robótica educativa.

SUMMARY

In recent decades the industry has evolved over several revolutions. The last iteration of the industry appeared a few years ago when robotics was included as part of the industry. Education has been influenced by different industrial revolutions, but now the need for industry requires a change in the profile of students. In the short and medium term, the industrial fabric demands a large number of professionals with knowledge in the areas of science, technology, engineering and mathematics, the so-called STEM (Science, Technology, Engineering and Mathematics). On the other hand, educational robotics emerges as a promising driver of STEM education. In today's world, there are many solutions for applying educational robotics in STEM contexts, but the educational contents are not very developed or do not facilitate the queueing between students. Likewise, there are solutions that are not open, which makes it difficult to use them or the flexibility to develop different types of robotic applications in educational contexts. Hence, the logical device program-ble has been chosen as the core of the platform that has been developed as part of this doctoral thesis. Programmable logic devices provide flexibility during the design cycle because design iterations are simply a matter of changing the programming file. In addition, they can be reprogrammed even while they are working. Also, programmable logic allows for parallel processing that can be used to monitor tasks or avoid communication problems such as unwanted bottleneck delays. On the other hand, Ar-duino allows an easy and fast prototyping platform. Arduino is also an open source, extensible hardware and software platform. This paper describes the combination of powerful resources in a single tool. The Educational Robotics Collaborative Tool (HeCRE) combines hardware, firmware and open source software fully integrated into the same platform. In this way, it is guaranteed that the modifications can be easily applied in the platform with the objective of adapting it to the learning program or to the learning sessions that require it. Also included is a complete development methodology, future lines of work and how they are planned to be used. The proposal takes robotics to the learning process, from simple learning objects to complex projects. Scalability, modularity, reconfiguration possibilities and compatibility are the main characteristics of its design. In this way, the platform is included as part of the educational scenarios within the context of educational robotics.

PREFACIO

Cinco años no es un periodo de tiempo exageradamente grande, pero para mí ha sido un período de tiempo cargado de ilusiones, experiencias, inquietudes y resultados que es difícil de detallar en la extensión que se merece, cuando se trata de plasmar todas las vivencias que he disfrutado a lo largo del desarrollo de esta tesis doctoral. A lo largo de este documento, sintetizo la parte académica del proceso que ha significado para mí y para mi desarrollo académico esta etapa de mi vida que ha sido la realización de esta tesis doctoral.

Aprovecho este espacio para detallar los motivos que me condujeron a iniciar esta tesis, y algunos detalles y vivencias más destacables que ocurrieron durante estos cuatro años.

Tuve la oportunidad de desarrollar un trabajo fin de máster sobre robótica basada en FPGA con fines educativos, pero no pude profundizar con este trabajo en la parte educativa tanto como me hubiese gustado. Además, la idea de desarrollar una tesis doctoral con el apoyo de un equipo humano tan fabuloso como el constituido por Elio, Germán y Manuel fue un aliciente más. Desde el principio de esta tesis, me lo he tomado como una afición. ¿Por qué? Porque trabajo los temas que me interesan y me apasionan, tengo total libertad sobre lo que hago, cómo lo hago y cuándo lo hago y, además, siempre he recibido el apoyo que he necesitado, en la medida que me ha hecho falta, y en ningún momento me han agobiado.

En las fases más tempranas de esta tesis Mónica, Javi, César y Alejandro fueron mis “co-nejillos de indias” con mis experimentos de robótica educativa con aquellos robots tan graciosos basados en Arduino y en Crumble. Hasta montamos unas piernas robóticas y un brazo robótico, ambos basados en servos.

En el verano de 2015 nació Plaza Robótica, “un lugar de encuentro para los amantes de la robótica”. Desde entonces hasta la fecha siempre ha tenido una gran vinculación con esta tesis doctoral.

Esta tesis también se ha nutrido de cursos en el Centro Internacional de Tecnologías Avanzadas de Peñaranda de Bracamonte y en Complubot de Alcalá de Henares. Asimismo, hemos dado cursos anuales y talleres en La Estera (Camarma de Esteruelas), en el centro comercial El Círculo (Torrejón de Ardoz), en la Academia PÍXEL (Rivas), en el centro asociado de la UNED en Las Tablas. Hemos colaborado con Science World en Alcalá de Henares e incluso hemos tenido experiencias de robótica educativa en la Parroquia Nuestra Señora del Rosario en Torrejón de Ardoz.

CAPÍTULO 1

1. INTRODUCCIÓN

Las tecnologías evolucionan día a día con una gran rapidez. Ahora, los estudiantes tienen que estar familiarizados con la tecnología y la manera de adquirir conocimientos relativos a programación y al uso de los tipos de dispositivos que serán claras oportunidades en su futuro profesional.

Siguiendo la estela del software abierto, en los últimos años, ha surgido el llamado hardware abierto. Su objetivo es diseñar dispositivos informáticos de forma abierta, de manera que todas las personas puedan acceder, como mínimo, a los planos de construcción de dichos diseños hardware.

Lejos de ser una novedad, esta corriente enlaza directamente con la década de los años 70, cuando los primeros aficionados a los ordenadores construían sus propios equipos en los garajes con piezas compradas a diferentes fabricantes y creaban sus propias implementaciones. El gran impulso que ha recibido el Hardware Abierto ha sido gracias a iniciativas como Arduino [Arduino, 2018] y Raspberry Pi [Raspberry, 2018]. Tras estas dos innovaciones, diversos fabricantes de tarjetas se sumaron a la creación de nuevas tarjetas de desarrollo de propósito general. Un ejemplo es Beaglebone [Beagleboard, 2018] la cual es proporcionada por la empresa Beagleboard. Otro ejemplo es la tarjeta Intel Edison que es una aportación de Intel.

A lo largo de este capítulo se introducen las bases sobre las que se sustenta esta tesis doctoral. También se incluyen las motivaciones que han conducido a la realización de esta tesis doctoral. Además, se muestran las necesidades que han promovido el desarrollo de esta tesis doctoral. Asimismo, se detallan los aspectos innovadores conseguidos a lo largo de la investigación. Finalmente se exponen las etapas a través de las cuales se han logrado los resultados mostrados.

Una de las principales motivaciones de esta tesis doctoral es lograr una herramienta que proporcione una mejora en el proceso de aprendizaje en el marco de los conocimientos relacionados con las áreas de ciencia, tecnología, ingeniería y matemáticas, también denominado STEM.

1.1. Mejora de la calidad educativa de la ciencia y la tecnología en España

Desde 2006, el aprendizaje en la educación española está regulado por la Ley Orgánica de Educación (LOE). En 2013, esta ley ha sido mejorada con la elaboración de una nueva Ley Orgánica para la Mejora de la Calidad Educativa (LOMCE) que promueve, como parte de otros cambios importantes, la ciencia y la tecnología como principales competencias a incluir en los programas educativos españoles. Estos programas de educación se enmarcan en la Educación Primaria Obligatoria, la Educación Secundaria Obligatoria, la Etapa de Bachillerato y la Educación Profesional, todas ellas clasificadas como de grado.

La comunidad científica está potenciando la educación científica [Rodríguez, 2014], la educación tecnológica [Blázquez, 2013] y la educación STEM (Ciencia, Tecnología, Ingeniería y Matemáticas) [Plaza, 2016]. La robótica se ha convertido en una herramienta importante para que los estudiantes se involucren en las áreas de ciencia, tecnología y educación STEM. Esto ocurre debido a la presencia de kits didácticos robóticos como soluciones asequibles que pueden ser fácilmente introducidas en el aula. Además, la Robótica está siendo utilizada como modernización y mejora de la mayoría de los procesos. Esto ocurre como resultado de que los robots pueden integrarse fácilmente en los procesos industriales actuales [Carro, 2012b].

En la actualidad están surgiendo dos fenómenos tecnológicos, IoT y Big Data. En el contexto de la IoT, las cosas pueden intercambiar información para llevar a cabo la toma de decisiones por sí mismas [Rodríguez, 2014]. Por otro lado, Big Data es la ciencia que analiza una gran cantidad de datos adquiridos con el objetivo de tomar decisiones sobre el campo en que se han recogido esos datos (educativos, médicos, etc.). La industria actual se está transformando bajo el nombre de Industria 4.0, la Cuarta Revolución Industrial. La Industria 4.0 se basa en la digitalización, IoT y Big Data con el objetivo de proporcionar sus productos manufacturados desde la fábrica hasta el hogar del cliente. Todo este proceso debe tener una duración similar al tiempo que tardaría el cliente en recibir el producto, en el caso que el producto ya estuviese fabricado y listo para ser enviado en el almacén del fabricante. Por lo tanto, para hacer realidad esta transformación, las empresas necesitan evolucionar hacia empresas digitales. De esta manera, en los próximos años será necesario contar con personal cualificado en materia de ciencia y tecnología. Además, la educación española debe alinearse con la estrategia de crecimiento Europa 2020 [Europa, 2018] para la próxima década. En el siglo XXI, sin embargo, la programación de computadoras es vista como una habilidad crítica y necesaria que todos deben aprender [Shim, 2016]. Así, Europa inició un viaje de uso e implantación de

las TIC con el aval de la Comisión Europea en 2010. Se materializó con la Agenda Digital [Agenda, 2018] cuyo principal objetivo era el desarrollo de un mercado digital único que condujera a Europa a un crecimiento inteligente, sostenible e integrador. Las acciones que debía llevar a cabo la Unión Europea eran:

- Introducir el objetivo de un mercado digital único.
- Abrir un acceso legal en línea a los contenidos simplificando las autorizaciones.
- Facilitar los pagos y la facturación por vía electrónica.
- Unificar los servicios de telecomunicaciones.
- Reforzar la interoperabilidad y las normas.
- Consolidar la confianza y seguridad en línea.
- Promover un acceso rápido a Internet.
- Invertir en investigación e innovación.
- Fomentar la cultura, las competencias y el sector de las TIC.
- Aprovechar el uso ingenioso de la tecnología por parte de la sociedad.

Las herramientas de robótica educativa son oportunidades prometedoras para ser usadas en la educación STEM (Ciencia, Tecnología, Ingeniería y Matemáticas). A pesar de esto, todavía es necesario lograr herramientas educativas STEM que integren estas cuatro áreas de desarrollo en las instituciones de educación formal. El uso de la mecánica de juego (actualmente denominada gamificación) tuvo un efecto positivo en la motivación de los estudiantes mientras estos realizaban tareas de alto grado de dificultad. También, tal y como detallan los autores en [Hew, 2015], el efecto de la gamificación logró que los estudiantes desarrollaran construcciones de gran calidad. La solución adecuada debería reducir los esfuerzos económicos, de formación y de comprensión. Las conclusiones que se exponen en [Christensen, 2015] son probablemente la aplicación de varios tipos diferentes de actividades prácticas de participación STEM fomenten o mantengan una predisposición positiva a aplicar STEM en los niveles de la escuela de educación primaria y secundaria. Los niños australianos utilizan las computadoras en la escuela de manera muy similar, practicando principalmente habilidades específicas de aprendizaje (por ejemplo, matemáticas y alfabetización) y rara vez participan en actividades creativas [Hatzigianni, 2016]. Pero la robótica enmarcada en la educación STEM potencia la creatividad cuando los niños usan computadoras. Además, una herramienta educativa STEM promueve las siguientes competencias de los estudiantes:

- Trabajo en equipo,
- resolución de problemas,
- resiliencia

- y comunicación.

Además, la creatividad también puede reforzarse con prototipos de ciencia ficción dentro de un marco educativo. Por lo tanto, se pueden diseñar y crear construcciones educativas robóticas disruptivas con el objetivo de mejorar las sesiones de aprendizaje [Johnson, 2013]. Para validar los contenidos de aprendizaje, el FTL (Flexible Thinking in Learning) puede integrarse en objetos educativos robóticos para ayudar en la evaluación de la inclinación de los alumnos a pensar de manera flexible o para explicar las diferencias individuales con respecto a la utilización de nuevos métodos de aprendizaje [Barak, 2016]. El método MRP (Multi-Role Project) ayuda a los estudiantes a adquirir importantes conocimientos y habilidades profesionales, a experimentar realidades profesionales del mundo real y a desarrollar sus habilidades para trabajar en equipo y de forma autónoma [Warin, 2016]. "Involucrar a los futuros ingenieros" es un tema que está en el centro de las conversaciones cotidianas sobre la enseñanza de la ingeniería [Deanne, 2018]. [Adams, 2011] ilustra con el ejemplo el proceso, los principios subyacentes y el valor de tomar la metodología de perspectivas múltiples para desafiar las suposiciones previas y configurar un sistema de meta-investigación que revela nuevas áreas de erudición e innovación.

Además, algunas iniciativas han surgido con el objetivo de proporcionar una plataforma robótica rentable para la educación STEM, como [Susilo, 2016]. Asimismo, el proyecto Thymio se describe en [Mondada, 2017] como una solución educativa robótica de código abierto y acceso libre que puede ser utilizada fácilmente por científicos, profesores y diseñadores de la educación. Otro ejemplo es el proyecto Duckietown, una plataforma abierta, económica y flexible para la educación e investigación autónoma. La plataforma comprende pequeños vehículos autónomos ("Duckiebots") construidos a partir de componentes estándar, y ciudades ("Duckietowns") completas con carreteras, señalización, semáforos, obstáculos y ciudadanos (duckies) que necesitan transporte [Paull, 2017]. Como se identificó en [Bethke, 2013], las unidades curriculares de ciencias basadas en el diseño de ingeniería pueden apoyar el conocimiento del contenido científico de los estudiantes de primaria, a la vez que ayudan a los estudiantes a aprender a diseñar, construir y probar soluciones a problemas de ingeniería. Por otro lado, las artes pueden contribuir y ser una parte integral de la educación STEAM. [Sochacka, 2016] promueve que el enfoque STEAM ofrece muchas más posibilidades.

Las leyes españolas están actuando sobre la enseñanza de la ciencia y la tecnología desde la Educación Infantil, la Educación Primaria Obligatoria, para proporcionar conocimiento a las generaciones futuras.

Como bien es sabido, desde hace ya algunos años el proceso de aprendizaje ha estado bastante ligado al uso de Internet como herramienta de apoyo a los contenidos educativos. Por este motivo, hemos considerado interesante hacer un análisis de las relaciones que hay entre el aprendizaje y el uso de Internet en por parte del profesorado y de los alumnos.

1.2. Aprendizaje e Internet

La combinación del proceso de aprendizaje con Internet promueve la innovación de las prácticas educativas. Esto se debe, en parte, al desarrollo de nuevos medios y materiales educativos. De esta manera, se abren nuevos espacios y tiempos de aprendizaje. Gracias al uso de las tecnologías de la información y de la comunicación se facilita el rediseño de las metodologías educativas. Por todo esto, es posible adquirir conocimientos incrementando el grado de experimentación a lo largo del proceso de aprendizaje.

A través de Internet es posible aplicar diferentes metodologías educativas (aprendizaje basado en tareas, inquiry learning spaces tal y como se describe en [Gillet, 2017] y de más ejemplos relacionados). La colaboración viene dada por las herramientas de colaboración que se apliquen al escenario de aprendizaje (foros, chats, espacios comunes de trabajo, etc.). De acuerdo con [Pumipuntu, 2008], esta metodología ayuda a profesores y alumnos participar en una clase en cualquier lugar y a cualquier hora.

Hay multitud de ejemplos de uso de sistemas educativos basados en Internet tal y como puede verse en los trabajos de los autores [Farkhadov, 2016], [Perva, 2016], [Martínez-Mediano, 2016], [Valdez, 2016], [Bovier, 2016], [Torre, 2016], [Irawati, 2016] y [Wei, 2016]. Cabe destacar un ejemplo que es el empleado para salvar las barreras de la distancia que se describe en [Johnson, 2010]. Este trabajo muestra un claro ejemplo de aprendizaje basado en Internet en un contexto transatlántico. Otro ejemplo es [Mustafa, 2014], donde los autores apuestan por el aprendizaje basado en Internet para alcanzar población rural en zonas donde no existen demasiados recursos. Por otro lado, en [Rojas, 2014] combinan el concepto de Gamificación con educación relacionada al área de las profesiones de la salud.

Por otra parte, estas relaciones llevan a cuestionar los roles y modelos de aprendizaje y de uso de tecnologías de información y de comunicaciones tradicionales. El cambio en el entorno educativo se vuelve una necesidad, si es que se quiere tener éxito en la creación de comunidades virtuales que aprenden y que lo hacen aprovechando las nuevas oportunidades de colaboración, interacción, exploración, experimentación y presentación que están disponibles actualmente. Para ello, es necesaria una interacción entre las

metodologías tradicionales con las nuevas corrientes metodológicas que han surgido en los últimos años, como se señala en [Delgado, 2015].

Como herramientas emergentes de la enseñanza basadas en Internet cabe destacar el gran avance de los laboratorios remotos, ver [Carro, 2012a], siendo una tecnología en continua evolución. En [Carro, 2015b] se puede comprobar como la mecatrónica y la robótica son herramientas que ayudan a la motivación, siendo fácil su inclusión en entornos de laboratorios remotos. En [Carro, 2014] se presenta un laboratorio remoto sencillo, está basado en Arduino, lo cual permite a profesores desarrollar nuevos aspectos prácticos de la teoría de la percepción y un amplio rango de disciplinas relacionadas, tanto con la física, como con la ingeniería. En [Tawfik, 2013] se puede observar otro claro ejemplo de laboratorio remoto para la experimentación con circuitos electrónicos.

Para poder aprovechar todo el potencial de los laboratorios remotos, quizás sea conveniente explorar las oportunidades que puede ofrecer Internet of Things (IoT) al entorno educativo.

1.3. Internet of Things - IoT

Cada día más personas están usando Internet para comunicarse con otras para acceder a un amplio rango de información, para compartir sus ideas o conectar dispositivos que pueden ser accedidos desde cualquier lugar y en cualquier momento.

De acuerdo con [Cui, 2012], en la esfera del Internet of Things (IoT), las Cosas pueden identificarse y conectarse a través de Internet. Lo que permite que dichas cosas intercambien información y puedan tomar decisiones por ellas mismas.

Actualmente, es posible encontrar multitud de aplicaciones basadas en IoT las cuales están siendo desarrolladas en muy diversos campos. Un ejemplo es [Xu, 2014], que se enmarca en el campo de los servicios médicos, donde se demuestra cómo recoger, integrar, e interactuar con los datos proporcionados por los dispositivos IoT que dan soporte a los servicios médicos de emergencias. Además, [Nastic, 2013] propone un novedoso modelo de programación que facilita un modo eficiente e intuitivo para desarrollar aplicaciones IoT basadas en una Plataforma en la Nube.

También es posible encontrar aplicaciones Educativas basadas en IoT. Como se indica en [Pruet, 2015] y [Carro, 2015a]. En la primera referencia, usando dispositivos de bajo coste, los estudiantes ubicados en zonas rurales desfavorecidas pueden acceder a un aprendizaje interactivo, a cualquier hora, y en cualquier sitio. La segunda remarca cómo

laboratorios remotos sencillos pueden estar fácilmente conectados como elementos IoT, de manera que implementan herramientas educativas.

Llegados a este punto, hemos considerado oportuno identificar las plataformas que hay en la actualidad para poder desarrollar una herramienta enmarcada en el contexto de educación STEM, la robótica educativa y que se pueda fácilmente integrar en un entorno compatible con IoT.

1.4. Tipos de plataformas de desarrollo

Hoy en día hay infinidad de plataformas de desarrollo basadas en distintos tipos de dispositivos. Las actuales plataformas se pueden catalogar en cuatro grandes grupos:

- Basadas en Microcontroladores que integran microcontroladores, comunicación serie y puertos de entrada/salida digitales y analógicos. Ejemplos de esta categoría son Arduino y Crumble [Crumble, 2018].
- Basadas en Microprocesadores que están formadas por un microprocesador, memoria RAM, procesador gráfico y diferentes tipos de comunicación, como puede ser USB y Ethernet. Raspberry Pi o Beaglebone Black son claros ejemplos de esta categoría.
- Basadas en FPGA: usan la FPGA como procesador principal. Dependiendo del uso para el que son concebidos, pueden incluir elementos usados en las plataformas basadas en microcontroladores, y existen alternativas que incluyen memoria RAM al estar pensadas para ser programadas con un microcontrolador embebido. Estas últimas suelen incluir memoria RAM y algún tipo de sensor como puede ser un acelerómetro o sensor de temperatura. TerasIC [TerasIC, 2018] ofrece una amplia variedad de este tipo de tarjetas.
- Plataformas de Desarrollo Híbridas comúnmente llamadas SoC y este tipo de tarjetas se componen por un microprocesador y lógica programable. Además, cuentan con elementos que se incluyen en los grupos anteriores. TerasIC también ofrece tarjetas que son catalogadas dentro de este grupo.

De los cuatro grupos, hemos considerado que tiene más potencial las plataformas basadas en FPGA, ya que este tipo de dispositivos lógicos programables, entre otras ventajas, son capaces de proporcionar soluciones hardware a medida, unos tiempos de ciclo de diseño menores y gran flexibilidad.

Antes de lanzar el desarrollo de una plataforma basada en FPGA, es recomendable hacer uso de kits de desarrollo para identificar si el dispositivo elegido es adecuado. Por esta

razón hemos explorado este tipo particular de plataformas de desarrollo basadas en FPGA en formato de kit de desarrollo.

1.5. Kits de desarrollo basados en FPGA

Estos kits son usados comúnmente por las universidades como herramientas que permiten, de forma eficaz, cubrir programas educativos de electrónica digital durante sesiones de laboratorio. De esta manera, los estudiantes pueden complementar los contenidos teóricos de los cursos de electrónica digital con casos prácticos.

Estos kits de desarrollo son normalmente muy generalistas. A parte de la FPGA, incluyen multitud de dispositivos hardware. Estos dispositivos son usados para manejar un monitor mediante la entrada VGA, interfaz serie para comunicar con un PC, conversores digitales analógicos, memorias y otros tantos tipos de dispositivos. Debido a esa gran cantidad de dispositivos, el coste del kit es normalmente muy elevado aparte de ser una buena herramienta de marketing usada por los fabricantes, los cuales proporcionan kits de desarrollo a precios muy competitivos.

Otra desventaja de los kits de desarrollo es el uso reducido que se hace de ellos. Aunque contienen una gran cantidad de dispositivos, en muchas ocasiones, no llegan a ser usados. Por ejemplo, las aplicaciones educativas no suelen requerir tantos componentes como los que incluyen estas plataformas. Que un kit esté formado por multitud de dispositivos, implica un coste asociado que no se aprovechará durante los experimentos que desarrollen los alumnos en las sesiones de laboratorio. Además, elaborar contenidos educativos con multitud de dispositivos supone un grado de complejidad que, en la mayoría de los casos, no se debería alcanzar en un proceso de aprendizaje de programas educativos de electrónica digital.

La necesidad de unas plataformas sencillas, y de menor coste, hoy en día, está siendo cubierta por la aparición de iniciativas de hardware abierto, donde se incluye la FPGA y un número muy reducido de componentes electrónicos suficiente para programar la FPGA y alimentar dicho dispositivo. Debido al reducido número de componentes y conexiones, se puede inferir un coste bastante asequible para cualquier persona que esté interesado en usar estas nuevas plataformas, tanto estudiantes, como para aficionados a la electrónica. La desventaja de este tipo de sencillas plataformas es la necesidad de añadir toda la electrónica relacionada con el proyecto que se desee elaborar.

Tras haber analizado este tipo de plataformas, debemos observar cómo queremos conectar los diferentes elementos que compondrán los escenarios educativos. Por tanto, hemos analizado los tipos de comunicaciones que es más conveniente usar.

1.6. Comunicaciones Wireless en la educación

La comunicación Wireless se puede definir como “la transferencia de información entre dos o más puntos que no están conectados mediante un conductor eléctrico” [Stair, 2011].

Las comunicaciones Wireless permiten a los estudiantes moverse a lo largo del área del aula. Esto se debe a la ausencia de cables que puedan provocar accidentes no deseados como puede ser tropezar con los cables de conexión entre dispositivos. Otra ventaja es la reducción de complejidad cuando un nuevo dispositivo tiene que ser añadido, o la limitación del número de dispositivos que pueden ser añadidos. Además, las redes Wireless promueven un área de trabajo limpio y ordenado para los estudiantes.

En la actualidad, las comunicaciones Wireless más populares están basadas en: WiFi, Radio Frecuencia y Bluetooth.

La comunicación Wireless es especialmente útil como parte de los laboratorios remotos educativos porque proporcionan una gran flexibilidad en el aula, como se muestra en [Carro, 2012a].

Con el fin de atraer el interés de los estudiantes en la ingeniería como campo de estudio, [Newberry, 2015] presenta una aplicación de sensores basada en Wireless como herramienta educativa para poder ser usada en una plataforma de automatización de vivienda, abarcando el área de conocimiento de la domótica.

[Cui, 2012] es una contribución en el campo de laboratorio online basado en la web, el cual proporciona una solución para estudiantes de instituto. WeFiLab facilita la inclusión de experimentos con redes Wireless posibilitando que estudiantes hagan experimentos sobre una página web vía Internet en cualquier momento y en cualquier lugar.

Al robot Boe-Bot de Parallax se le aplicaron modificaciones para incluir comunicaciones Bluetooth para poder ser usado en clase. Los resultados sobre programaciones simultáneas múltiples sobre Bluetooth se describen en [Ruberg, 2015].

1.7. Innovaciones que aporta este trabajo

Está claro que las tecnologías evolucionan vertiginosamente. Gracias a iniciativas basadas en compartir conocimiento y desarrollos de manera abierta se han logrado grandes avances en infinidad de campos, uno de ellos, el campo tecnológico. Hoy en día nos vemos envueltos por tecnologías tan complejas como el Internet de las Cosas o el Big Data. Los estudiantes actuales y futuros deben familiarizarse con el uso de estas tecnologías. En primer lugar, para poder acceder a una base de conocimientos que cada día que pasa tiene un volumen mayor. Por otro lado, para un desarrollo de su carrera profesional que les permita ejercer su profesión. Desde ya hace unos años nos hemos familiarizado en gran medida con el aprendizaje a través de Internet. También estamos muy habituados a manejar multitud de dispositivos Wireless, ya sea ese ratón que se comunica por radio con el PC, o ese altavoz Bluetooth que reproduce una lista de reproducción que tenemos en nuestro Smartphone, e incluso, esa Tablet que usamos mientras desayunamos para consultar las noticias a través de la conexión WiFi del rúter que tenemos en el salón.

Pues bien, todos esos dispositivos que existen, que se están desarrollando o que se desarrollarán en un futuro, están basados en distintos tipos de plataformas. Cuando se trata de aprender a hacer desarrollos basados en estas plataformas, disponemos de kits de desarrollo. Estos kits de desarrollo, o son muy caros, o tienen tantas cosas que nos da vértigo incluso aproximarnos para empezar a conocerlos y usarlos, o son como islas que no permiten desarrollos colaborativos. También se pueden encontrar soluciones cerradas que ni siquiera nos permiten utilizarlas o adaptarlas a nuestras necesidades.

El resultado de la investigación muestra una plataforma de hardware y software abierto que puede ser usada en aulas para elaborar programas educativos relacionados con robótica en un entorno de desarrollo colaborativo para potenciar la innovación y la motivación de los alumnos durante el proceso de aprendizaje. La plataforma se presenta como una herramienta educativa modular, reconfigurable, flexible, versátil y con un coste que puede ser fácilmente adaptado a las necesidades y al poder adquisitivo de la entidad interesada en incluir esta herramienta en sus programas educativos. De esta manera, el objetivo principal que se busca con esta tesis doctoral es desarrollar una plataforma de robótica educativa que cumpla con las siguientes premisas:

- Reconfiguración del sistema para permitir diferentes configuraciones de escenarios en función de las necesidades.

- Comunicaciones concurrentes que preservan los retrasos no deseados del sistema.
- Escalabilidad para adaptar los costes del sistema o su complejidad.
- Se requiere flexibilidad en una base modular con el objetivo de implementar cualquier ocurrencia sin trabajo pesado.
- Protección del sistema para mitigar los riesgos de pérdida, mal uso y mala administración.
- Protección menor en los periféricos, como el control de tensión y corriente y la corrección correspondiente.
- Facilidad de uso para el desarrollo de prototipos.
- Compatibilidad con PC y Smart Devices como Smartphones o Tablets que permite conexiones con diferentes dispositivos adicionales añadidos como parte del proceso de aprendizaje.

El objetivo final de la plataforma es ser una herramienta que ayude al formador a despertar la curiosidad de los alumnos mediante experimentos que potencien el trabajo en equipo y comunicación entre los alumnos.

El ámbito de aplicación es STEM, desarrollo de proyectos educativos científico-tecnológicos que ayuden en la enseñanza de materias relacionadas con la ciencia, la tecnología, la ingeniería y las matemáticas.

Los escenarios de utilidad incluidos en el capítulo 7 muestran las posibilidades de aplicación de la plataforma presentada. Dichos casos pueden ser llevados al aula, tanto presencial como de manera remota. Además, servirán de ejemplos de aplicación para poder ser usados, modificados o adaptados gracias al carácter de hardware y software abierto que tiene esta iniciativa. Para compartir todos los diseños elaborados se usa la plataforma Github [Github, 2018]. El proyecto creado para tal efecto se encuentra en [Repositorio, 2018] con el nombre de Educational-Robotic-Platform, donde se recogen los archivos del diseño hardware, diseño firmware, diseño software y la información relativa a los escenarios de utilidad de la plataforma.

1.8. Organización de este trabajo

En un primer lugar se exponen los agradecimientos a aquellas personas e instituciones que han contribuido a lo largo de la experiencia investigadora para lograr una herramienta útil para otras personas.

Posteriormente se incluye el índice general. También se incluye un listado de símbolos, abreviaturas y siglas que puede ayudar a comprender los diferentes conceptos expuestos a lo largo de este trabajo. Por otro lado, se incluye un listado de las figuras y tablas que facilitan la comprensión de conceptos o datos aportados.

Además, se incluye un resumen en castellano e inglés que incluye las motivaciones de esta investigación. Asimismo, se adjunta un prefacio donde se explica de dónde provienen las motivaciones y la visión personal de la experiencia adquirida a lo largo de esta investigación.

A lo largo de este capítulo 1, introductorio, se han expuesto, en líneas generales, las necesidades y la importancia de desarrollo de una herramienta de robótica educativa. Igualmente se han añadido las aportaciones innovadoras que ha supuesto la investigación desarrollada a lo largo de esta tesis doctoral. El estado del arte de esta tesis doctoral se expone a lo largo del capítulo 2. El capítulo 3 condensa los recursos de conocimiento adquiridos como fase inicial de la investigación. También se incluye en este capítulo una descripción detallada de las herramientas usadas como referencia durante la fase de desarrollo. Además, el capítulo 4 contiene los resultados fruto de experiencias educativas usando algunas de las herramientas mencionadas en el capítulo 3. Para ello, se ha trabajado con algunas herramientas de robótica educativa a lo largo de la investigación. Esto se hizo, en primer lugar, para tener un conocimiento en mayor profundidad de las capacidades que tienen estas herramientas. También se hizo para poder comparar resultados del desarrollo de la plataforma con otras alternativas existentes en la actualidad. Además, todas las experiencias realizadas han servido para la definición de los escenarios educativos descritos en el capítulo 9.

Como siguiente paso, el capítulo 5 contiene todos los detalles relacionados con la arquitectura de la plataforma de robótica educativa desarrollada. Desde la metodología usada, hasta la descripción de los diferentes elementos que se incluyen. El capítulo 6 agrupa los diferentes pasos que se han llevado a cabo a lo largo del desarrollo hardware de la plataforma. Pasos descritos en el capítulo 5 e implementados a lo largo de la fase de desarrollo de las placas y las conexiones eléctricas de los diferentes elementos que componen la plataforma de robótica educativa. A continuación, el capítulo 7 reúne toda la información relativa al desarrollo del firmware que se incluye en la FPGA. De la misma manera que con el desarrollo hardware, los pasos seguidos a lo largo del desarrollo firmware están descritos en el capítulo 5. En el capítulo 7 también se incluye información sobre las pruebas de integración hardware-firmware que se han llevado a cabo como parte de esta etapa. El capítulo 8 agrupa la última fase de desarrollo. En este momento

se describe el desarrollo del software ejecutado por las Arduino que forman parte de la plataforma de robótica educativa. También se siguen los pasos de descritos en la metodología definida en el capítulo 5. Asimismo, a lo largo del capítulo 8 se detallan las pruebas de integración del sistema completo, es decir, las pruebas de integración hardware-firmware-software.

En un siguiente paso, tal y como se describe en el capítulo 9, se describen los posibles escenarios educativos que pueden ser implementados con la plataforma y el escenario educativo usado durante las pruebas de sistema descritas en el capítulo 5. En este capítulo también se incluyen las pruebas con alumnos como última etapa de pruebas para validar la plataforma.

A lo largo de esta tesis se han implementado diferentes desarrollos. Estos desarrollos han supuesto una inversión de esfuerzo, tiempo y dinero. Por todo ello, el capítulo 10 condensa un informe económico de los costes relacionados con estos desarrollos.

El capítulo 11 incluye una relación de las lecciones aprendidas a lo largo de la investigación. Este capítulo incluye los errores cometidos, los riesgos a que la investigación se vio expuesta, las decisiones que mejor funcionaron y, los procesos y técnicas que más eficiencia y efectividad aportaron.

El capítulo 12 trata sobre las conclusiones y líneas de trabajo futuro que han surgido del resultado de la investigación llevada a cabo.

El capítulo 13 contiene el currículum del autor. El capítulo 14 incluye las referencias bibliográficas utilizadas a lo largo de la investigación que ha conducido a esta tesis doctoral.

Por último, el Anexo 1 describe el contenido del DVD que se incluye con la presente tesis doctoral y el Anexo 2 incluye algunos de los méritos y publicaciones destacables obtenidos durante la investigación.

CAPÍTULO 2

2. ESTADO DEL ARTE

La primera revolución industrial se remonta al período comprendido entre los siglos XVIII y XIX. La transición fue de la producción manual a las máquinas, lo que permitió la nueva fabricación química y el procesamiento del hierro. La industria textil era la industria dominante. Además, también surgió el uso de la energía de vapor con máquinas herramienta y sistemas de fábrica [Ashton, 1948].

Después de un siglo, una nueva evolución de la industria apareció entre los siglos XIX y XX. La segunda revolución industrial incluyó la mejora de la industria de la máquina-herramienta, el desarrollo de métodos para la fabricación de piezas intercambiables y la invención del Proceso Bessemer [Muntone, 2013].

La introducción de la maquinaria fue uno de los acontecimientos más importantes de la revolución industrial. La inclusión de la robótica en la industria se considera la tercera revolución industrial de principios del siglo XX [Carro, 2012]. La Tercera Revolución Industrial, además, se basa en la confluencia de tres grandes facilitadores tecnológicos (es decir, la analítica de Big Data, los servicios adaptativos y la fabricación digital) [Tien, 2012]. Hoy en día estamos familiarizados con el proceso de fabricación automatizado, máquinas que colaboran en el interior de la fábrica con el objetivo de producir diferentes tipos de productos [Rifkin, 2008].

La crisis de la deuda europea comenzó en la Unión Europea a finales de 2009. Por lo tanto, los líderes europeos marcaron en 2010 una nueva ruta para Europa. Este plan se basa en cinco objetivos mensurables para el año 2020:

- Empleo
- Investigación e innovación
- Cambio climático y sostenibilidad energética
- Educación y lucha contra la pobreza
- y la exclusión social.

Cada Estado miembro ha adoptado sus propios objetivos nacionales en cada uno de estos ámbitos. La estrategia se basa en acciones concretas a nivel europeo y nacional [EUROPE, 2018].

Así, Europa inició un viaje de utilización e implantación de las TIC (Tecnologías de la Información y la Comunicación) con el aval de la Comisión Europea en 2010. Esta iniciativa se materializó con la Agenda Digital cuyo principal objetivo era el desarrollo de un mercado digital único que condujera a Europa a un crecimiento inteligente, sostenible e integrador. Las acciones que debía llevar a cabo la Unión Europea eran:

- Introducir el objetivo de un mercado digital único
- Abrir un acceso legal en línea a los contenidos simplificando las autorizaciones
- Facilitar los pagos y la facturación por vía electrónica
- Unificar los servicios de telecomunicaciones
- Reforzar la interoperabilidad y las normas
- Consolidar la confianza y seguridad en línea
- Promover un acceso rápido a Internet
- Invertir en investigación e innovación
- Fomentar la cultura, las competencias y el sector de las TIC
- Aprovechar el uso ingenioso de la tecnología por parte de la sociedad

Varias empresas se están sumando al objetivo de Europa 2020 alineando sus visiones con la nueva era de la digitalización. Siemens se ha posicionado a lo largo de la electrificación, automatización y digitalización. La visión de Samsung es inspirar al mundo con tecnologías, productos y diseños innovadores que enriquezcan la vida de las personas y contribuyan a un futuro responsable y sostenible.

El enfoque de la transformación digital se acerca a la llamada Industria 4.0 o cuarta revolución industrial. Ésta se ha entendido como la aplicación a escala industrial de sistemas automatizados como los robots. Para ello, se centra en los procesos productivos, la llegada de sistemas ciber-físicos y la interconexión de las unidades productivas. Por todo esto, la transformación digital se orienta a la creación de redes de producción donde automatismos interactúan unos con otros dentro del marco productivo. Así, se puede lograr una fabricación digital más eficiente y ágil que la fabricación tradicional.

De ahí que se consiga crear redes de producción digital que aceleren la producción y utilicen los recursos de forma más eficiente que antes.

Según Boston Consulting Group, hay nueve tendencias tecnológicas que sientan las bases de esta evolución de la industria:

- Big Data, mediante el análisis de datos de diferentes fuentes (producción, gestión, equipos de consumidores) hará de la toma de decisiones en tiempo real un estándar.
- Robots autónomos, capaces de interactuar y ejecutar decisiones más complejas
- Simulación 3D de productos, materiales y procesos.
- Integración vertical y horizontal de procesos, entre proveedores y clientes, y entre todos los departamentos de la empresa.
- La IoT (Internet of Things), en la que se añadirán sensores en todos los procesos y productos no terminados para interactuar.
- Ciberseguridad, reforzando las conexiones y estandarizando los protocolos de comunicación.
- La nube (Cloud Computing), que permite el análisis en tiempo real de multitud de datos y la toma de decisiones inmediatas.
- Impresión 3D, para muestras de diseños complejos o prototipos.
- Realidad aumentada para tomar decisiones sobre la gestión de un almacén u órdenes de reparación de una máquina o un producto.

Se espera que la IIoT (Industrial Internet of Things) impulse el desarrollo industrial en un futuro previsible; los retos que deben superarse para realizar el potencial de la conectividad de la IIoT presentan interesantes oportunidades de investigación [Mumtaz, 2017].

Los ICPS (Industrial CyberPhysical Systems) son percibidos como un enfoque prometedora que extiende las actividades generales de los CPS (CyberPhysical Systems) al ámbito industrial y considera predominantemente sus requisitos tales como seguridad, migración, cumplimiento, agilidad, continuidad del negocio, rendimiento y colaboración [Colombo, 2017].

Estos dispositivos son capaces de funcionar sin la operación del ser humano. Claros ejemplos son los UGV (Unmanned Ground Vehicle) o los UAV (Unmanned Aerial Vehicle). En la literatura se encuentran ejemplos como el que [Menendez, 2017] describe. Este trabajo muestra cómo se llevan a cabo las tareas de inspección y mantenimiento de los sistemas de energía mediante robots y vehículos aéreos no tripulados. Las plataformas robóticas se están convirtiendo en actores importantes de la sociedad.

Poniendo el foco en la legislación educativa española, la inclusión de asignaturas técnicas en las escuelas secundarias se estableció en 1990 según la LOGSE [LOGSE, 1990]. Fue la primera vez que los planes de estudio introdujeron la asignatura de Tecnología e Informática, junto con asignaturas comunes como Matemáticas, Ciencias, Idiomas, etc.

Como siguiente paso, se presentó la tecnología como una forma de acercar a los estudiantes al mundo industrial. Esto se logró mediante la introducción de temas relacionados con:

- El proceso de resolución de problemas,
- el conocimiento de los materiales utilizados en la fabricación de productos como la madera, los metales y los plásticos,
- y, además, el aprendizaje de los circuitos eléctricos y electrónicos.

Era una época en la que se acababa de inventar la Web y la introducción de la informática en la escuela casi se hacía mediante juegos. Las primeras prácticas de aplicaciones de oficina como el procesamiento de textos WordPerfect, hojas de cálculo Lotus y la gestión de bases de datos.

Ya han transcurrido más de dos décadas desde entonces, y el panorama informático y técnico ha cambiado ampliamente, principalmente debido a la actualización de las tecnologías, a partir de las cuales se difunde globalmente el uso de Internet y de la Web como tecnologías habituales de información y comunicación en la sociedad.

La actualización más significativa de la legislación educativa española se ha producido en 2013, cuando se elaboró una nueva ley en España. La Ley de Educación conocida como LOMCE fue presentada como una ley en la cual los estudiantes deben estar ubicados en el centro del proceso de enseñanza-aprendizaje. En un enfoque más profundo, la ley literalmente establece que: "La educación es el motor que promueve el bienestar de un país. El nivel educativo de los ciudadanos determina su capacidad para competir con éxito en el ámbito internacional y para hacer frente a los retos que surjan en el futuro. Mejorar el nivel de los ciudadanos en la educación significa abrir las puertas a puestos de trabajo altamente cualificados, lo que representa un compromiso con el crecimiento económico y un futuro mejor" [LOMCE, 2013].

Además, la ley profundiza en la forma de lograr estas primeras afirmaciones, describiendo eso: "Es necesario promover condiciones que permitan el cambio metodológico apropiado, para que los estudiantes se conviertan en un elemento activo en el proceso de aprendizaje. Hoy los estudiantes han cambiado radicalmente en relación con los de hace una generación. La globalización y el impacto de las nuevas tecnologías hacen que la forma de aprender a comunicar, a centrar la atención o a abordar una tarea sea significativamente diferente" [LOMCE, 2013].

Se puede observar la importancia que se le da a las TIC y a las nuevas tecnologías como reconocimiento para que el área de Tecnología sea considerada como una asignatura

troncal en el ámbito de la educación como se menciona en la ley de la siguiente manera: "Las TIC serán la clave para producir el cambio metodológico hacia el logro del objetivo de mejorar la calidad de la educación. Además, el responsable y ordenado de estas nuevas tecnologías por parte de los alumnos debe estar presente en todo el sistema educativo. La tecnología de la información y las comunicaciones también será una herramienta clave en la formación del profesorado y el aprendizaje de los ciudadanos a lo largo de toda la vida, permitiéndoles combinar la formación con las obligaciones personales o laborales, así como en la gestión de los procesos" [LOMCE, 2013].

La tecnología no sólo ha afectado en gran medida a la educación. La evolución tecnológica tiene un impacto importante en el sector empresarial. La Secretaría de Estado de Investigación, Desarrollo e Innovación (SEIDI) destacó la responsabilidad de su área en todo lo relacionado con la gestión de los recursos para el desarrollo de la innovación, en este caso dirigida a la Industria 4.0. Entre los ocho retos que engloban sus objetivos se encuentra la sociedad y la economía digital, entre los que destacan el CDTI (Centro para el Desarrollo Tecnológico Industrial) y la SETSI (Secretaría de Estado para las Telecomunicaciones y para la Sociedad de la Información), trabajando conjuntamente en el marco de una coordinación permanente, especialmente en el ámbito de la bioeconomía, manteniendo una estrecha vinculación con el Ministerio de Agricultura y Medio Ambiente. Reconoce la existencia de duplicaciones en el uso de los recursos administrativos, en términos de subsidios, que están corrigiendo [SEIDI, 2018].

Cada vez más, las empresas están utilizando Big Data como herramienta de planificación estratégica. Telefónica ha ahorrado 2,8 millones de euros al modificar el proceso de seguimiento de las reclamaciones de sus clientes [Telefónica, 2018]. Usar Big Data no solo repercute en un ahorro de costes. Cuando los clientes de Telefónica aceptan compartir sus datos, se sienten más seguros y Telefónica resuelve problemas legales. No se trata de acumular datos, sino de procesar y sacar conclusiones de estos. Sin embargo, desde el punto de vista de un proveedor de comunicaciones, podría ser necesario optimizar aún más o incluso desarrollar tecnologías industriales específicas, especialmente cuando sea necesario cumplir con requisitos de aplicación rigurosos [Wollschlaeger, 2017]. El 20 de octubre de 2016, Telefónica lanzó LUCA (Last Universal Common Ancestor), su nueva unidad de servicios Big Data, que permite a sus clientes corporativos entender sus datos y fomentar un uso transparente y responsable de los mismos [Telefónica, 2018].

Un ejemplo análogo para ilustrar lo que la tecnología de Cloud Computing ha significado para el análisis de Big Data podría ser lo que la infraestructura de la red eléctrica significó para la electricidad. Antes de la creación de las redes de transporte y distribución, era

necesario disponer de un generador propio para poder utilizar esta fuente de energía. Hoy en día, la red eléctrica permite un acceso generalizado al suministro eléctrico. Lo mismo ocurre con el almacenamiento en la Nube y los servicios informáticos. Permite controlar a distancia - sin necesidad de estar físicamente en la propia fábrica - el funcionamiento de la maquinaria, las diferentes fases del proceso de producción, la gestión de inventarios, etc. Esto significa un ahorro significativo de tiempo en la resolución de problemas, gestión y costes de coordinación, junto con otros beneficios. [Manic, 2016] muestra una arquitectura BEMS (Building Energy Management System) basada en CI (Computational Intelligence) para proporcionar excelentes resultados en términos de ahorro de energía en diferentes escenarios.

Siemens se está posicionando a lo largo de la electrificación, automatización y digitalización. A lo largo de estas cadenas de valor, Siemens ha identificado varios campos de crecimiento en los que ve su mayor potencial a largo plazo. La compañía está orientando su asignación de recursos hacia estos campos de crecimiento y ha anunciado medidas concretas en esta dirección. Además, Siemens se orienta hacia su Visión 2020, que está alineada con la iniciativa Europa 2020. "Nuestra Visión 2020 aborda las perspectivas a largo plazo de nuestra empresa a lo largo de las modernas cadenas de valor de la electrificación y la automatización. Al ampliar la participación de los empleados basada en acciones en el éxito de nuestra empresa, estamos creando una cultura de propiedad sostenible en Siemens", dijo el presidente y Director General de Siemens [Siemens, 2018]. En España, la División de Movilidad de Siemens es responsable del transporte eficiente e integrado de personas y mercancías, por tren o carretera, y supervisa todos los productos, soluciones y servicios relacionados con la movilidad [Mobility, 2018].

Por otro lado, Indra está desarrollando la Ciudad 2020, un nuevo modelo de ciudad inteligente y sostenible. Se trata de un proyecto español de I+D+i de ciudades inteligentes, que ha recibido una financiación de 16,3 millones de euros en el marco del programa INNPRONTA del Centro para el Desarrollo Tecnológico Industrial (CDTI). Se está desarrollando a lo largo de 33 activos tecnológicos experimentales que han sido definidos, desarrollados y probados en ciudades como Málaga, Santander y Zaragoza, como parte de un nuevo modelo de ciudades inteligentes y sostenibles en el que el ciudadano es el protagonista [Indra, 2018]. La Ciudad 2020 es información en tiempo real para los gestores de la ciudad, así como soluciones y retos para el futuro.

Según las principales conclusiones del estudio de Accenture, el sector industrial prevé invertir 220.000 millones de euros, una cuarta parte de su presupuesto, en Inteligencia Artificial y máquinas durante los próximos cinco años para aumentar la productividad a

través de una plantilla conectada mediante la combinación de tecnologías de movilidad, seguridad y monitorización, mediante la analítica (sistema para realizar análisis de los datos recibidos), con el objetivo de mejorar la actividad de los trabajadores industriales. Esto supondría una inversión de 181.000 millones de euros por parte de las empresas de automoción y de 39.000 millones por parte del material industrial [Accenture, 2018].

Accenture predice que para 2020, una plantilla industrial conectada podría ayudar a un fabricante de automóviles - con unos ingresos anuales de 50.000 millones de euros - a obtener hasta 500 millones de beneficios adicionales (incluidos 50 millones de beneficios adicionales en I+D, 415 millones en fabricación y suministro y 30 millones después de la venta) [Accenture, 2018]. Al mismo tiempo, según PWC y Consulting Group, la digitalización de la industria podría generar unos ingresos adicionales de 110.000 millones de euros al año para la industria europea en los próximos cinco años [PWC, 2018]. Por todo lo mostrado, se prevé una necesidad de personal cualificado en áreas de conocimiento de ciencias, tecnología, ingeniería y matemáticas.

2.1. La educación STEM

La educación en ciencias, tecnología, ingeniería y matemáticas (STEM) integra conceptos que normalmente aparecen separados. Esto es, un curso de ciencias está formado por contenidos puramente científicos. El autor en [Tsai, 2018] muestra cómo es posible promover la capacidad de investigación de los estudiantes y el conocimiento científico de la microbiología y la intoxicación alimentaria. Un curso de tecnología se compone exclusivamente de conocimiento tecnológico. Valga como ejemplo un entorno de aprendizaje de realidad virtual que integra tecnología de realidad virtual y contenido de información digital de realidad virtual interactiva en 3D que se describe en [Liou, 2018]. Si se imparte un curso de ingeniería, los contenidos educativos se basan en conceptos ingenieriles. La referencia [Tabunshchuk, 2018] ilustra programas de estudios en implantes artificiales para la bioingeniería en Ucrania. No es diferente para la enseñanza de las matemáticas. Delice y Roper en su trabajo [Delice, 2006] detallan aspectos particulares de enseñanza de las matemáticas en el sistema educativo inglés. A pesar de ello, cuando se trabaja la educación en estas cuatro áreas de manera aislada se suele hacer hincapié en la aplicación del conocimiento a situaciones de la vida real.

Hoy en día estamos expuestos a problemas sociales y tecnológicos más complejos por lo que es necesario profesionales cualificados más polifacéticos. Se necesitan profesionales cualificados en STEM para resolver problemas sociales y tecnológicos complejos y hacer frente a las demandas contemporáneas como la energía sostenible y la atención

sanitaria eficiente [Knipprath, 2018]. La enseñanza de la ingeniería en el siglo XXI se enfrenta a múltiples obstáculos. Los autores en [August, 2015] proponen como solución parcial un Laboratorio Virtual de Aprendizaje de Ciencias de Ingeniería (VESLL) que proporciona objetos interactivos y actividades de aprendizaje, pantallas multimedia y procedimientos de retroalimentación instantánea. Con este ejemplo se logra combinar tres de las cuatro áreas de conocimiento: Ciencia, Tecnología e Ingeniería. Por su lado, en [Parkhomenko, 2018] se puede ver una combinación de Tecnología e Ingeniería mediante el uso de laboratorios remotos con fines formativos para ingenieros. También es posible encontrar ejemplos que combinan el método científico con la educación en ingeniería, tal y como se detalla en [Fan, 2018].

A continuación, se muestran algunos ejemplos de uso de STEM en universidades, escuela e industria:

- En [Stubbs, 2009] se muestra un taller STREAM que se llevó a cabo en iRobot para maestros de K-12 con una serie de módulos de robótica diseñados para capturar la imaginación de los estudiantes escolares, grandes y pequeños.
- Muchas universidades y escuelas han estado abordando esta falta de interés por parte de los alumnos en las áreas de conocimiento STEM. Lo han hecho utilizando una amplia variedad de programas de participación para alentar y retener a los estudiantes en las disciplinas de ciencias, tecnología, ingeniería y matemáticas. El ejemplo de LaserTag muestra cómo los estudiantes sueldan juntos y guardan los dispositivos electrónicos LaserTag, que pueden usar para dispararse paquetes de luz infrarroja unos a otros para tener sus propias actividades de escaramuza [Ross, 2017]. Como se muestra en este ejemplo, es positivo usar gamificación para atraer a los estudiantes y hacer que se involucren en un mayor grado durante el proceso de aprendizaje.
- Los autores en [Williams, 2017] muestran los resultados tras diez años de trabajos con socios de la industria para implementar programas sostenibles que ayudan a los maestros a dar a los estudiantes oportunidades de aprender las habilidades duras y blandas que necesitarán para ser competitivos en la fuerza laboral de ciencia, tecnología, ingeniería y matemáticas.

La educación STEM tiene un gran potencial y su popularidad crece día a día [Pickering, 2016]. La educación primaria se enfrenta al desarrollo del pensamiento STEM y de actitudes favorables a STEM en los alumnos de edades tempranas. [Kintsakis, 2017] describe el diseño, la implementación y la evaluación como una solución para abordar ambos

problemas. [Mavroudi, 2017] ofrece una visión y una orientación para el futuro sobre muchos aspectos interesantes: métodos y herramientas pedagógicas innovadoras (incluida la evaluación), aspectos comunitarios, aspectos del diseño del plan de estudios y aspectos del diseño de la instrucción que tienen en cuenta los costes de la UMI (computación Ubicua, computación Móvil e Internet de las Cosas) para mejorar la educación STEM. La tecnología educativa es una herramienta poderosa que está aumentando su presencia día a día. Algunos ejemplos son los proporcionados por [Delgado, 2017] como parte de la red eMadrid.

Pese a todo lo anterior, el reto de la educación STEM es desplegar un contenido educativo que conjugue las cuatro áreas de conocimiento. Se trata de lograr un hilo argumental que aproveche las sinergias de las cuatro áreas: ciencia, tecnología, ingeniería y matemáticas. Asimismo, las lecciones STEM deben asegurar que los estudiantes entiendan cómo los conceptos están conectados con el mundo real. Hoy en día, la educación STEM está en el foco de la comunidad científica [Plaza, 2016]. Además, hay una gran variedad de plataformas robóticas. La mayoría de ellas pueden ser fácilmente utilizadas como herramientas educativas en el contexto de la educación STEM. A pesar de todo ello, existen dos grandes barreras: el coste de la plataforma y la ausencia de contenidos educativos asociados a la herramienta educativa. Basándonos en el coste, se pueden encontrar dos tipos:

- Las plataformas con precios más elevados incluyen diferentes tipos de actuadores, sensores y piezas mecánicas. Asimismo, suelen proporcionar una completa colección documental sobre como desplegar contenido educativo haciendo uso de la plataforma.
- Plataformas más rentables que proporcionan un controlador principal, unos pocos actuadores - generalmente motores de corriente continua (CC) - y un tipo de sensor como el sensor de distancia por ultrasonido. En algunos casos, incluyen también algunas construcciones mecánicas simples.

La robótica educativa se muestra como un prometedor potenciador de la educación STEM ya que, permite transmitir a los alumnos de una manera transversal y divertida conocimientos, facilita el desarrollo del pensamiento lógico, al mismo tiempo que estimula la creatividad y permite familiarizarse con el funcionamiento de objetos programables.

2.2. Robótica educativa como potenciador de STEM

El creador del término "robótica" fue Isaac Asimov en una historia de 1941 publicada en la revista *Astounding Science Fiction*. Aunque, como el propio Asimov más tarde reconoció en el artículo "La palabra que inventé", pensó que estaba usando una palabra existente. Una de esas historias es el conocido libro "Yo, Robot". La imagen del robot que aparece en su obra es la de una máquina bien diseñada con una seguridad garantizada que actúa de acuerdo con las tres leyes de la robótica.

Desde 1941 hasta nuestros días, la robótica se ha convertido en un referente para la ciencia y la tecnología. También es una pieza clave para la modernización y mejora de la mayoría de los procesos [Plaza, 2017c]. Esto se debe a la facilidad con la que los robots pueden integrarse en los procesos industriales [Carro, 2012]. La autora de [Mazumder, 2016] presenta el desarrollo de una prótesis activa de tobillo y pie (AFO) para mejorar la capacidad de caminar. Otro ejemplo de implementación robótica es el robot con patas ligeras y elásticas que utiliza la tensión superficial para mantenerse en la superficie del agua descrita en [Zhou, 2017].

La robótica se está incluyendo como una herramienta con gran potencial para alentar a los estudiantes a acceder a materias relacionadas con la ciencia, la tecnología, la ingeniería y las matemáticas.

Sin embargo, la introducción a la robótica como proceso de aprendizaje no es una cuestión trivial. La robótica combina áreas de conocimiento como la mecánica, la electricidad, la electrónica y la informática. Los robots son máquinas capaces de tomar decisiones y adaptarse a diferentes situaciones. Normalmente se construyen con elementos como sensores, actuadores y unidades de proceso [Plaza, 2017c].

La educación robótica se está volviendo muy popular hoy en día. Los robots simples están siendo usados dentro de la Educación STEM como una herramienta poderosa que facilita la manera de enseñar el conocimiento STEM. Además, la robótica también proporciona una manera atractiva de transformar conceptos aburridos en un proceso de aprendizaje divertido. Los kits de robótica facilitan la conexión entre las disciplinas STEM [Susilo, 2016]. También se puede usar la robótica educativa para contrarrestar la cantidad de horas que emplean los jóvenes en el uso de Internet y los videojuegos desde la infancia.

Una de las causas por las que es posible el uso de la robótica en clase es que las nuevas generaciones se familiarizan y manejan dispositivos tecnológicos a una edad muy temprana. Este fenómeno puede ser tanto positivo como negativo. Por un lado, la tecnología facilita el proceso de comunicación con nuevos canales que acercan a las personas de forma digital con soluciones como Skype, WhatsApp y las redes sociales. Por otro lado, estas nuevas formas de comunicación crean una dependencia que reduce las actividades de comunicación físicas. Este escenario hiperconectado está afectando las relaciones entre los individuos y está introduciendo nuevas oportunidades y nuevos desafíos [Settembre, 2012].

Otro aspecto fundamental en la educación es la relación entre Internet y el aprendizaje. Esta relación conduce a la innovación en las prácticas educativas, como se ha señalado en estudios anteriores [Blázquez, 2013] y [Rodríguez, 2014]. Es posible articular nuevos medios de comunicación, crear multitud de materiales educativos, trabajar en una gran diversidad de espacios (presenciales o virtuales), adaptar los tiempos de aprendizaje a los alumnos y rediseñar los contenidos educativos. Para poder lograr todo esto, es necesario el uso de las tecnologías de la información y la comunicación. De esta manera, se logra que el conocimiento se adquiera con un mayor grado de experimentación a lo largo del proceso educativo. [Fernández, 2017] presenta una experiencia basada en el desarrollo de aplicaciones Android con la intención de aumentar la motivación, satisfacción y compromiso de los estudiantes en la programación.

Sin embargo, esta relación (entre Internet y el aprendizaje) lleva a cuestionar los roles y modelos de aprendizaje y el uso de las tecnologías tradicionales de información y comunicación. Cambiar el entorno educativo es una necesidad si se quiere tener éxito en la creación de comunidades virtuales que aprenden aprovechando las nuevas oportunidades de colaboración, interacción, exploración, experimentación y presentación que se tienen a nuestra disposición. El trabajo de [Perini, 2017] presenta los resultados de un estudio exploratorio destinado a comprender el impacto del aprendizaje basado en juegos digitales (DGBL) en la concienciación y el interés de los estudiantes de secundaria por la fabricación.

Entre los múltiples factores, los siguientes han sido probablemente las cinco causas más influyentes del auge de la robótica educativa:

- Prototipos fáciles de desarrollar. Cada vez es más fácil crear nuevos dispositivos tecnológicos, y las impresoras 3D hacen que la creación de prototipos sea rápida

y económica. Estas soluciones permiten a los estudiantes experimentar con dispositivos móviles. Android, Linux y Arduino también pueden combinarse en un pequeño ordenador para permitir la interacción con múltiples sensores y actuadores.

- Cambio de conectividad en las expectativas del cliente. El proceso común de compra de hardware consistía en "uno y hecho", es decir, los clientes compraban su propio hardware, y el proceso terminaba allí. Actualmente, necesitamos y queremos varios dispositivos que nos permitan hacer las mismas cosas, pero desde diferentes lugares y perspectivas. En otras palabras, necesitamos más hardware y esperamos mucho más de estos elementos. Además, esperamos que como estos dispositivos están conectados entre sí, sus posibilidades y ventajas aumenten. El hardware conectado define una nueva experiencia de hardware.
- Desafíos para la compatibilidad de software y hardware. En este punto, los desarrolladores deben ser capaces de combinar diferentes tipos de hardware con el mismo software para poder compartir la misma información con el usuario, y este escenario presenta un gran desafío. Si el hardware está diseñado para la excelencia, el software también debe estar diseñado para la excelencia. La financiación de multitudes ha cambiado la relación entre las marcas y los distribuidores. Los sitios web de financiación de multitudes como Kickstarter, Indiegogo o Crowdsupply han ayudado a muchos proyectos de hardware a avanzar, y estos sitios permiten a las marcas conectarse directamente con el cliente. Tradicionalmente, el distribuidor estaba situado en el centro y actuaba como intermediario entre el cliente y la marca. En la actualidad, las marcas pueden trabajar directamente con sus clientes, omitiendo así este paso intermedio y creando un tipo de relación de confianza entre la marca y el cliente.
- El hardware abierto aumenta la velocidad de la innovación. Cuando este movimiento se combina con el hardware, la innovación une a los diseñadores que pueden trabajar en el diseño original y en los accesorios. Con el hardware de código abierto, los desarrolladores y las empresas de nueva creación no tienen que buscar la aprobación del creador y pueden empezar a trabajar por su cuenta sin necesidad de obtener patentes o licencias.

Actualmente, existen varias opciones para introducir la robótica educativa. En [Plaza, 2017a], Crumble se presenta como una herramienta que puede ser utilizada para desplegar el conocimiento STEM en el hogar para adultos y niños. Además, [Plaza, 2017b] explica cómo usar Scratch para introducir a los estudiantes a la robótica, los fundamen-

tos de la programación y habilidades tales como el pensamiento sistémico, la mentalidad de programación, el aprendizaje activo, las matemáticas, la ciencia, el juicio y la toma de decisiones, la comunicación efectiva, el diseño tecnológico, la resolución de problemas complejos y la persistencia. Asimismo, los planes de estudio no relacionados con las ciencias de la computación pueden escalarse utilizando escenarios colaborativos para mejorar la enseñanza y el aprendizaje de las habilidades de programación [Echeverría, 2017].

La robótica se ha convertido en una herramienta importante para la sociedad. La robótica también se ha convertido en una herramienta útil para que los estudiantes se involucren en las áreas educativas STEM. Esto se debe a que se cuenta con kits de robótica educativa. Estos kits no suponen una gran inversión y son fácilmente integrables en el aula educativa. A continuación, se puede ver una serie de ejemplos de aplicación de robótica en STEM:

- [Aroca, 2013] muestra cómo un robot reconfigurable y extensible sirve como plataforma experimental en varios cursos de pregrado
- Los autores en [Bicchi, 2005] describen el trabajo realizado para hacer el laboratorio de robótica accesible a estudiantes y colegas permitiendo que los experimentos sean ejecutados y vistos en tiempo real en cualquier momento y desde cualquier lugar
- La robótica se utiliza como tecnología en actividades diseñadas para mejorar el compromiso y la enseñanza de los maestros con STEM a través de actitudes mejoradas hacia STEM [Kim, 2015]
- Además, [Tedder, 2004] describe una plataforma robótica móvil que consiste en varios módulos integrados que se utilizan para implementar el control de lógica difusa
- Para un equipo de robots, [Díaz-Báñez, 2017] ofrece un enfoque que puede utilizarse para múltiples tareas, como la vigilancia, la exploración de áreas y la búsqueda de objetivos en entornos peligrosos, entre otras
- Además, se pueden encontrar ejemplos en la literatura relacionada con el desarrollo de experimentos orientados al control de motores para un robot móvil remoto [Schilling, 2005]
- El autor en [Schreuder, 2011] demuestra cómo un simulador es una herramienta de entrenamiento útil para la cirugía robótica mediante cursos con diferentes herramientas educativas

- El autor en [Kaynov, 2008] propone una nueva arquitectura de software y hardware de control de movimiento para robots humanoides. Una importante aportación de esta arquitectura es que fue diseñada con un diseño abierto, que permite el desarrollo de plataformas robóticas humanoides de alta calidad tecnológicamente actualizables
- LEGO Mindstorm se utiliza ampliamente en la enseñanza de la robótica. En [Pinto, 2012] se describe un ejemplo de uso de LEGO NXT. En este caso, se describen experimentos en el marco de la enseñanza de conceptos de localización en robótica móvil. Además, el experimento se centra en explicar el concepto de filtro Kalman
- LEGO también se utiliza en concursos que fomentan el interés entre los estudiantes de edad escolar en los campos de la ingeniería y la ciencia [Karp, 2011]
- En la enseñanza de la ingeniería se están llevando a cabo prácticas innovadoras como las presentadas en [Gómez-de-Gabriel, 2015], donde los robots LEGO Mindstorms NXT se combinan con LabVIEW para utilizar laboratorios de enseñanza de DDF (Detección de Fallos y Diagnóstico) en sistemas mecatrónicos
- [Papadimitriou, 2007] analiza la posibilidad de utilizar elementos LEGO como material comercial de bajo coste con el objetivo de obtener una herramienta para el aprendizaje, la investigación y la experimentación en campos como la mecatrónica, la robótica y el control automático
- El uso de los robots LEGO Mindstorms NXT como plataforma de entrenamiento, redujo significativamente el esfuerzo de un grupo de profesores a la hora de preparar los ejercicios, e impulsó la reutilización de su trabajo entre varios temas y materias [Cruz-Martín, 2012]
- El autor en [Gatward, 2012] presenta un modelo pedagógico para un sistema de robots en línea conectado a Internet para su uso en la enseñanza superior. Se presenta una plataforma robótica móvil diseñada a medida y construida especialmente que, junto con su entorno de apoyo, proporciona una implementación del modelo pedagógico

Una vez analizado el impacto que tienen la robótica y la educación STEM en la sociedad, es el momento de profundizar más en las iniciativas que han hecho posible el movimiento actual relacionado con la robótica educativa.

2.3. Iniciativas y herramientas en torno a STEM

Siguiendo los pasos del movimiento del software libre, en los últimos años ha surgido lo que se ha dado en llamar hardware abierto, y su objetivo es crear diseños de dispositivos informáticos de forma abierta para que todas las personas puedan acceder, como mínimo, a los planos de construcción de dichos diseños de hardware.

Lejos de ser una novedad, esta tendencia se relaciona directamente con los años 70, cuando los primeros entusiastas de la informática construyeron sus propios equipos en garajes utilizando componentes adquiridos a diferentes fabricantes y crearon sus propias implementaciones. El movimiento de hardware abierto ha sido impulsado por iniciativas como Arduino [Arduino, 2018] y Raspberry Pi [Raspberry, 2018].

2.3.1. ARDUINO

Arduino fue desarrollado en 2005 por el entonces estudiante del IVRAE Massimo Banzi. Originalmente, Arduino surgió como una necesidad de aprendizaje para los estudiantes de informática y electrónica del mismo instituto porque, en ese momento, adquirir una placa basada en un microcontrolador era relativamente caro, y no se disponía del soporte adecuado.

Arduino también se extiende en la enseñanza de la robótica. [Jamieson, 2015] presenta una gama de placas para prototipado usadas con Arduino, Raspberry Pi y BeagleBone Black. Además, estos dispositivos están siendo utilizados en el currículo de los autores en una serie de cursos para proyectos. Los estudiantes se introducen en un marco basado en competencias con el objetivo de aprender los fundamentos de la mecatrónica y, en el proyecto, desarrollar un robot móvil utilizando una plataforma Arduino de bajo coste [Grover, 2014]. LabDuino, descrito en [Carvalho, 2016], es un laboratorio basado en Arduino que puede ser utilizado como repositorio de experimentos. Por lo tanto, usando LabDuino, los estudiantes pueden explorar temas y experimentar de acuerdo con las instrucciones correspondientes.

Arduino puede ser usado para una multitud de aplicaciones. Por ejemplo, para la implementación de un colorímetro de código abierto como plataforma para un mayor desarrollo y derivación de diseño para otros propósitos similares, como la nefelometría [Anzalone, 2013]. También puede utilizarse como parte de la construcción de un instrumento de medición de pH fiable y económico [Milanovic, 2018]. Arduino también se combina con FPGA (Field Programmable Gate Array) para el desarrollo de aplicaciones relacionadas con la detección de fallos en turbinas de vapor [Devi, 2018]. Los autores en

[Ghosh, 2017] proponen el diseño de dos controladores óptimos inteligentes para una trayectoria óptima utilizando un robot móvil autónomo basado en Arduino. También es posible desarrollar aplicaciones como un sistema de detección de perturbaciones en la red eléctrica en tiempo real [Natanael, 2017]. En el contexto de la Internet de las Cosas (IoT) se utiliza la plataforma Arduino, como se describe en la referencia [Al-Kaseem, 2017].

En el campo de la educación, Arduino también es ampliamente utilizado. Los estudiantes de ingeniería eléctrica pueden acceder a un laboratorio compacto para trabajar con él cuando y donde quieran [Graven, 2016]. También se utiliza el uso de Arduino en temas relacionados con la mecatrónica como se describe en las referencias [Phanomchoeng, 2016], [Sripakagorn, 2014] y [Chanchaoen, 2014]. La idoneidad de su uso en la educación se demuestra en obras como las que se muestran en las referencias [Garrigós, 2017], [Brox, 2016], [Carvalho, 2016], [Hertzog, 2016], [Petry, 2016] y [Assante, 2016]. En estos trabajos se investiga tanto en el campo de la educación universitaria como en la educación de pregrado.

Arduino ofrece un entorno de desarrollo intuitivo y múltiples recursos de hardware y software. Esto permite un rápido desarrollo del proyecto. Aun así, hay una desventaja, hay tanta información disponible que los estudiantes se vuelven incapaces de diseñar sus propios prototipos. En la referencia [Martínez-Santos, 2017] los autores proponen una metodología que introduce el estudio de microcontroladores usando Arduino. Con Arduino tienes todo tipo de información sobre tu diseño. Además, las plataformas basadas en Arduino son a menudo dispositivos virtualmente simples de usar. Sin embargo, a pesar de sus ventajas, aún no es confiable para su uso en infraestructura crítica [Yopez-Bonilla, 2017].

Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar [Arduino, 2018]. Es una plataforma que incorpora un simple microcontrolador y un entorno de desarrollo de interfaz para crear las aplicaciones que se descargan a la placa. El uso de los proyectos de Arduino cubre una amplia gama de aplicaciones, desde la robótica hasta los sistemas de control automático de riego.

El software de Arduino es gratuito, la solución hardware más económico tiene un precio de 20 €. Es muy fácil de usar, pero requiere programación textual. También incorpora hardware y componentes mecánicos que limitan la edad mínima de los usuarios. Es perfectamente válido a partir de los 12 años.

Arduino es una tarjeta capaz de manejar tanto señales analógicas como digitales. Integra una variedad de protocolos de comunicación tales como comunicación serial, SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit) y otros. Incluyendo escudos compatibles con Arduino, el controlador puede ser utilizado para cualquier tipo de aplicación, desde el control de motores hasta la implementación de un laboratorio remoto robotizado.

Aprender Arduino implica aprender muchas tecnologías dependiendo de cuán lejos quieras llegar, pero entre otras tecnologías se puede decir que Arduino incluye: electrónica digital y analógica, electricidad, programación, microcontroladores, procesamiento de señales, protocolos de comunicación, arquitectura informática, mecánica, motores, diseño de placas electrónicas, física, programación web, etc. Y Arduino nos permite empezar a trabajar con él sin tener que tener todo ese conocimiento, pero Arduino es el vehículo que nos ayudará a aprender todas esas tecnologías y ciencias de una manera práctica y a aprender haciendo cosas "aprendiendo haciendo", pero no hay duda de que hay que tener una cierta base para empezar.

Arduino ha socializado la tecnología, digamos que hemos tenido una buena idea desde hace tiempo y que no se puede hacer porque necesita más o menos conocimientos de electrónica, pero actualmente no los tenemos. Esa idea habrá pasado de proyecto a obstáculo. Arduino ayudará a hacer posible el proyecto gracias a su facilidad de uso, la gran información disponible en Internet y los desarrollos para Arduino publicados como código abierto. Leyendo manuales y practicando con el material proporcionado por Arduino y la comunidad, encontramos que en un tiempo relativamente corto somos capaces de ponernos a trabajar y atrevernos a escalar pequeños obstáculos y paredes hasta que encontramos nuestro propio límite o el del material mismo.

Prácticamente todas las variantes de Arduino tienen elementos en común como el puerto de comunicación/alimentación a través de un conector USB. También tienen puertos de entrada/salida, tanto analógicos como digitales, puertos que pueden ser usados para comunicaciones seriales, SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit). Algunas variantes de Arduino incluso incluyen control de motor y comunicaciones Bluetooth.

La Figura 1 muestra una tarjeta Arduino NANO con la identificación de cada uno de los puertos anteriores.

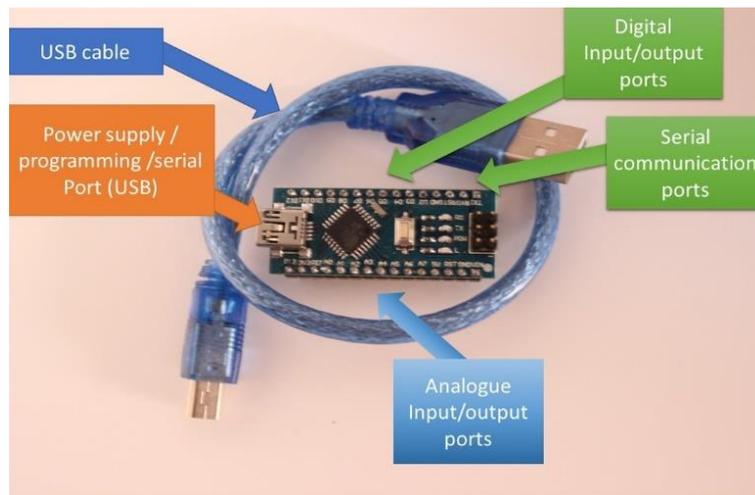


Figura 1. Puertos de la plataforma Arduino Nano.

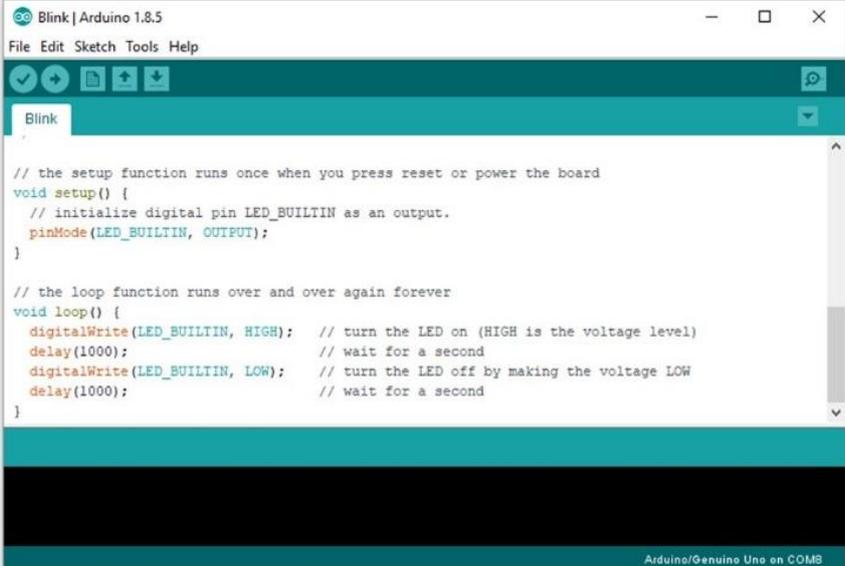
Arduino no sólo proporciona las placas (hardware), sino que también nos proporciona software que consiste en un entorno de desarrollo (IDE), un lenguaje de programación simplificado para el hardware y el cargador de arranque que se ejecuta en la placa. La característica principal del software (IDE) y del lenguaje de programación es su simplicidad y facilidad de uso.

La interfaz de desarrollo de Arduino incorpora ejemplos, parámetros de configuración y un editor de texto. Con el IDE de Arduino puedes seleccionar el puerto de comunicación que, para programar la tarjeta, el tipo de tarjeta que quieres programar, e incluso dispone de un terminal serie para comunicar el ordenador con la tarjeta.

Los programas de Arduino se componen de tres partes. En la parte superior se pueden incluir librerías, variables y constantes. Luego, en el área de configuración, puedes configurar los puertos de Arduino para que se comporten de la forma que desees. Por ejemplo, parámetros de configuración de puerto de salida, puerto de entrada y puerto de comunicación. Finalmente, dentro de la etiqueta de bucle, está la programación que quieres que la tarjeta Arduino ejecute.

La Figura 2 muestra un ejemplo que incluye la configuración del LED (Light Emitter Diode) que normalmente está integrado en todas las tarjetas Arduino. El código es muy simple y hace que el LED se comporte de la siguiente manera:

- Encienda el LED - ajuste el puerto que controla el LED a nivel alto
- Espere un segundo - la función de retardo se ajusta al tiempo en milisegundos
- Apague el LED
- Espere un segundo

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.8.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar shows icons for opening files, saving, and running. The main editor area displays the following code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

The status bar at the bottom indicates "Arduino/Genuino Uno en COM3".

Figura 2. Ejemplo de programa con el IDE de Arduino.

La placa Arduino es capaz de manejar una multitud de entradas y salidas, tanto analógicas como digitales. También puede configurar sus puertos para implementar diferentes tipos de comunicaciones. Los puertos de la placa Arduino no son capaces de alimentar dos motores DC, por lo que se deben utilizar componentes electrónicos separados.

Hay un tipo de placa llamada Arduino Shields que tiene un factor de forma compatible con los conectores de entrada/salida. Hoy en día existen placas Arduino para implementar casi cualquier funcionalidad. Ejemplos de escudos Arduino son el control de motores, la pantalla táctil TFT, las comunicaciones Ethernet, las comunicaciones Bluetooth, las comunicaciones WiFi y mucho más. Debido a que Arduino es hardware y software de código abierto, muchos fabricantes han desarrollado variantes que incluyen algunas de las características anteriores.

La Figura 3 muestra una construcción básica de un robot con ruedas usando Arduino. Este robot incluye un controlador Arduino en particular, este controlador integra el control de dos motores DC, sensor y servomotores y botones. Este kit robótico también incluye una batería y cuatro motores de corriente continua.

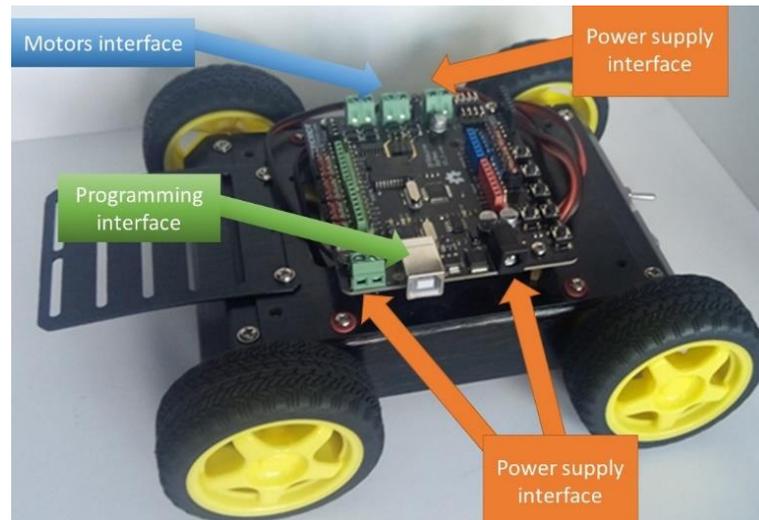


Figura 3. Ejemplo de robot con Arduino.

2.3.2. RASPBERRY PI

Más tarde, apareció la iniciativa Raspberry Pi. En 2006, los primeros diseños de Raspberry Pi se basaron en el microcontrolador Atmel ATmega644. El proyecto Raspberry Pi fue desarrollado por un grupo de profesores, académicos y entusiastas de la informática para crear una computadora con el fin de animar a los niños a aprender habilidades informáticas. Los esquemas y el diseño de la tarjeta de circuito impreso están disponibles para su descarga pública.

La Fundación Raspberry Pi desarrolló una computadora de tarjeta única (SBC) con el objetivo de promover la enseñanza de la informática básica en escuelas y países en desarrollo. Este SBC se llama Raspberry Pi y ha sido desarrollado en el Reino Unido. El modelo original se hizo mucho más popular de lo esperado, así que al principio fue bastante difícil conseguirlo. Además del uso para el que fue diseñado, la gente compró la Raspberry Pi para uso doméstico e incluso para usos relacionados con la robótica [Raspberry, 2018].

Aunque no se indica expresamente si se trata de hardware libre o con derechos de marca, la página web oficial explica que tienen contratos de distribución y venta con dos empresas, pero al mismo tiempo cualquier persona puede convertirse en revendedor o distribuidor de tarjetas Raspberry Pi, por lo que significa que se trata de un producto con propiedad registrada, manteniendo el control de la plataforma, pero permitiendo su libre uso tanto a nivel educativo como privado.

Sin embargo, el software es de código abierto, y su sistema operativo oficial es una versión adaptada de Debian, llamada Raspbian, aunque permite otros sistemas operativos, incluyendo una versión de Windows 10. Todas las versiones incluyen un procesador

Broadcom, memoria RAM, GPU, puertos USB, HDMI, Ethernet (el primer modelo no lo tenía), 40 pines GPIO y un conector para cámara. Ninguna de sus ediciones incluye memoria, siendo ésta en su primera versión una tarjeta SD y en ediciones posteriores una tarjeta microSD.

La Raspberry Pi ha evolucionado a la versión 3. Además, hay un modelo llamado Raspberry Pi Zero con un tamaño aún más compacto. La Figura 4 muestra una tarjeta de Raspberry Pi 3. Esta tarjeta SBC incluye: una CPU ARMv8 de 64 bits y cuatro núcleos de 1,2 GHz, LAN inalámbrica 802.11n, Bluetooth 4.1, Bluetooth Low Energy (BLE), 4 puertos USB, 40 pines GPIO (General Purpose Input Output), puerto HDMI completo, puerto Ethernet, conector de audio combinado de 3,5 mm y vídeo compuesto, interfaz de cámara (CSI), interfaz de pantalla (DSI), ranura para tarjeta microSD y núcleo de gráficos 3D VideoCore IV.



Figura 4. Raspberry Pi 3 Modelo B.

La Raspberry Pi es ampliamente usada en educación. En la literatura hay multitud de trabajos que detallan las experiencias educativas llevadas a cabo con este SBC:

- En [Hajdarevic, 2014] los autores detallan experimentos con software VOIP de código abierto (svxlink) usando una Raspberry Pi, una de las distribuciones de Linux disponibles, frecuencias de radioaficionados, transceptores y antenas de radioaficionados
- Por otro lado, los autores en [Jamieson, 2015] describen el uso, entre otros dispositivos, de la Raspberry desde un punto de vista educativo, y algunas mejores prácticas/técnicas que han aprendido y adoptado para incluir estos dispositivos en sus cursos

- También se encuentran ejemplos de implementación de una arquitectura alternativa con el propósito de proporcionar experimentos reales de acceso remoto y expandir las prácticas de enseñanza enfocadas en la experimentación hacia la educación en física [Daros, 2015]
- Los autores en [Trowbridge, 2017] describen cómo utilizan la Raspberry Pi como una infraestructura inalámbrica en la que los estudiantes podían acceder libremente a su Frambuesa Pi a través de la red de malla
- Los estudiantes universitarios pueden construir una cadena completa de hardware y software para una aplicación de procesamiento de imágenes digitales usando una Raspberry Pi como ordenador económico para ejecutar los programas de procesamiento de imágenes [Marot, 2017]
- Los autores en [Yamanoor, 2017] proporcionan una serie de ejemplos de proyectos y de código que pueden adaptarse rápidamente a diversas situaciones de aprendizaje

2.3.3. SCRATCH

Scratch fue desarrollado por la división de Educación del MIT (Massachusetts Institute of Technology). La misión del Instituto Tecnológico de Massachusetts es avanzar el conocimiento y educar a los estudiantes en ciencia, tecnología y otras áreas de estudios que mejor servirán a la nación y al mundo en el siglo XXI. También se ven impulsados a aportar conocimientos sobre los grandes desafíos del mundo [MIT Education, 2018a]. Con Scratch se pueden programar historias interactivas, juegos y animaciones. Además, permite compartir las creaciones con otras personas de la comunidad online. Scratch ayuda a los jóvenes a aprender a pensar de manera creativa, a razonar sistemáticamente y a trabajar en colaboración, habilidades esenciales para la vida en el siglo XXI. El proyecto Scratch ha recibido apoyo financiero de la National Science Foundation, Scratch Foundation, Google, LEGO Foundation, Intel, Cartoon Network, Lemann Foundation y la MacArthur Foundation [MIT Education, 2018b]. A lo largo de esta sección se detalla cómo se puede obtener Scratch. Además, también se describe el entorno de programación de Scratch. Finalmente, se incluye el tipo de usuarios y comunidades.

Scratch es un entorno de programación fácil de usar. Su interfaz de programación utiliza un lenguaje de programación en bloque que facilita su uso a los niños menores de 16 años. La robótica educativa puede implementarse fácilmente gracias a la sencilla estructura de la interfaz Scratch.

El sitio web de MIT Education Scratch ofrece diferentes maneras de usar Scratch:

- Uso en línea en [MIT Education, 2018b].
- Uso fuera de línea para diferentes sistemas operativos:
 - Windows
 - Debian / Ubuntu
 - Mac OS X

Hay dos versiones en línea: la versión 1.4 que se puede descargar en [Scratch, 2018a] y la versión 2.0 que se puede descargar en [Scratch, 2018b].

Los siguientes requisitos mínimos del sistema son necesarios para ejecutar Scratch correctamente:

- Sistema operativo Windows, Mac o Linux (32 bits)
- Adobe Air versión 2.6+ (incluido en la descarga)
- Más de 23 megabytes de espacio libre en el disco duro

MIT Education también proporciona dos tipos de recursos de referencia para los proyectos de programación de Scratch. El primero se encuentra en el sitio web de Scratch [Scratch, 2018c]. El segundo se puede encontrar como parte del entorno de programación. La Figura 5 muestra la ventana para importar proyectos con el fin de utilizarlos como referencia o para modificarlos.



Figura 5. Importar proyectos en Scratch.

El entorno de programación Scratch está dividido en tres secciones principales: el panel de comandos, el panel de programación y el panel de visualización. La Figura 6 muestra el entorno Scratch. La Figura 6 también identifica cada panel: círculo naranja para el panel de comandos, círculo azul para el panel de programación y círculo gris para el

panel de visualización. El panel de comandos se encuentra a la izquierda. Contiene 8 categorías de comandos: movimiento, control, apariencia, detección, sonido, operadores, lápiz y variables. Cada categoría tiene un conjunto de comandos que están relacionados entre sí. El panel de programación está situado en el centro. Contiene controles sobre el Sprite, un espacio para construir el guion que ejecutará el Sprite, los trajes que pueden usar el Sprite y los sonidos que pueden lanzar el Sprite. Los personajes son los actores de las historias o los personajes de los juegos que se crean con Scratch. La pantalla se encuentra a la derecha. Se divide en dos zonas. El área superior permite la vista de los personajes mientras ejecutan sus scripts, y los fondos que han sido programados en el escenario. La zona inferior permite seleccionar tanto el fondo como los personajes que se han creado en el proyecto abierto.



Figura 6. Entorno y paneles de Scratch.

En la parte superior del panel de comandos, se pueden ver las categorías de comandos. En la parte inferior de este panel se muestran los comandos agrupados en la categoría seleccionada.

Por otro lado, en la parte superior del panel de programación, se puede ver el Sprite. A la izquierda del Sprite hay tres botones, de arriba hacia abajo tienen las siguientes funciones:

- Permite que el carácter gire 360 grados
- Permite que el carácter gire sobre su eje vertical
- No permite que el carácter gire

En esta zona se muestra el nombre del Sprite. Además, la ubicación del Sprite se define como las coordenadas x e y donde está. También se incluye la dirección en la que apunta el Sprite. En este panel hay tres pestañas:

- Programas: lugar donde se añaden los comandos para construir un script
- Trajes: trajes para el Sprite, pueden ser importados o creados
- Sonidos: sonidos para el Sprite, pueden ser importados o creados

Finalmente, en la parte superior del panel de visualización se puede ver el fondo y los personajes mientras ejecutan sus programas. En la parte inferior del panel de visualización se puede seleccionar el fondo y los diferentes personajes.

Aunque Scratch fue diseñado principalmente para niños de 8 a 16 años, también es utilizado por personas de todas las edades, incluyendo niños pequeños con sus padres. El equipo de MIT Scratch trabaja con la comunidad para mantener un ambiente amistoso y respetuoso para personas de todas las edades, razas, etnias, religiones, orientaciones sexuales e identidades de género [Scratch, 2018d]. También trabaja todos los días para administrar la actividad en el sitio y responder a los informes, con la ayuda de herramientas como el filtro de blasfemia CleanSpeak [CleanSpeak, 2018]. Además, Scratch es ampliamente utilizado en la educación [Scratch, 2018e]. Los educadores están utilizando Scratch en una amplia variedad de:

- Entorno: escuelas [de Matos, 2018], museos [Ibarra, 2010], bibliotecas y centros cívicos
- Cursos: Educación Primaria [Hijón-Neira, 2017], Educación Secundaria Obligatoria y Bachillerato [Papavlasopoulou, 2018]
- Asignaturas: lengua, ciencias, ciencias sociales, matemáticas [Förster, 2018], informática, lenguas extranjeras y artes

Hay una comunidad para educadores, ScratchEd [ScratchEd, 2018a] es una comunidad en línea en la que los educadores de Scratch comparten historias, intercambian recursos, hacen preguntas y conocen gente. ScratchEd es desarrollado y apoyado por Harvard Graduate School of Education. Además, hay reuniones de educadores de Scratch que quieren aprender unos de otros, compartir sus ideas y estrategias para apoyar la creatividad computacional en todas sus formas [ScratchEd, 2018b].

2.3.4. CRUMBLE

Crumble es una tarjeta electrónica a la que se pueden conectar hasta dos motores, luces, interruptores y sensores para controlar su funcionamiento. Se conecta al PC a través de USB y puede programarse mediante el software gratuito. Redfern Electronics se encarga de fabricarlos y se pueden obtener algunos accesorios que son compatibles con la tarjeta Crumble [Redfern, 2018].

La controladora de Crumble es una tarjeta basada en un microcontrolador PIC de Microchip. Tal y como se describe en [Plaza, 2017a], esta tarjeta combina facilidad de uso con un gran número de características que la hacen apta para su uso en un entorno de educación STEM. Cabe destacar que incluye dos puertos de alimentación de entrada conectados internamente que pueden alimentarse con un nivel de tensión de 4,5 a 5,5 V DC (corriente continua). Estos puertos están marcados como POWER en la PCB (placa de circuito impreso) con su polaridad (+ y -). Por otro lado, la tarjeta dispone de un puerto USB tipo C utilizado para programar. También hay dos puertos de salida que pueden controlar motores DC de hasta 1 A DC. Los motores pueden ser gestionados en ambas direcciones. Además, las salidas del motor pueden actuar como salidas de alta potencia para muchos tipos de dispositivos. Estos puertos están marcados como MOTOR DRIVERS 1 y 2 en el PCB (placa de circuito impreso) con su polaridad (+ y -). Además, se pueden usar hasta cuatro puertos de entrada o salida que pueden ser utilizados como puertos analógicos o digitales. Estos puertos están marcados como A, B, C y D en la PCB (placa de circuito impreso).

La Figura 7 muestra la tarjeta Crumble y los diferentes puertos.

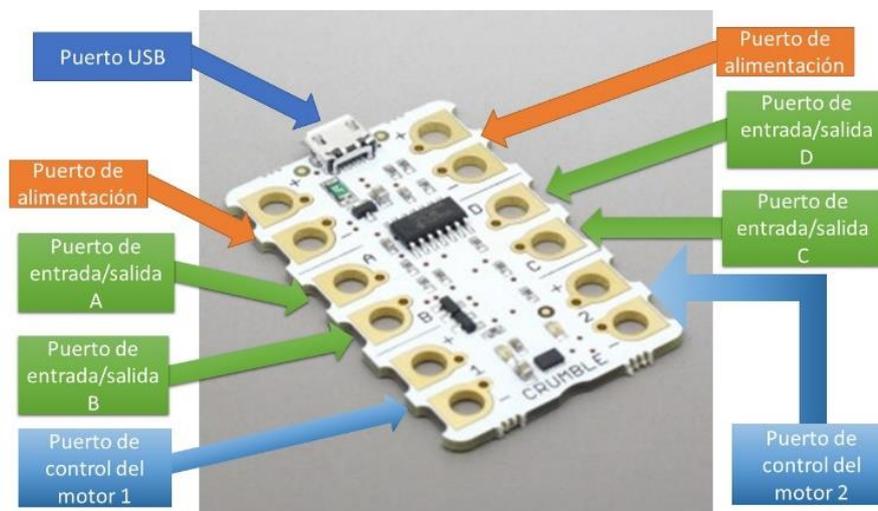


Figura 7. Puertos de la tarjeta Crumble.

No se requiere experiencia en programación, un sistema gráfico de arrastrar y soltar inspirado en el software MIT Scratch se proporciona libremente.

La interfaz de programación es un software de programación de bloques que se divide en tres zonas: pulsadores de inicio de programa / parada de programa, panel de comandos y panel de programación. La Figura 8 muestra una instantánea del software Crumble. También se incluye un programa simple.

El programa representado está destinado a:

1. Inicio del programa
2. Parar motores 1 y 2, y fijar el primer Sparkle al color rojo
3. Esperar 1 segundo
4. Ejecutar un bucle infinito con los siguientes comandos:
 - a. Fijar el sentido de los motores 1 y 2 en avance a una potencia del 75 % y fijar el primer Sparkle al color verde
 - b. Esperar 1 segundo
 - c. Fijar el primer Sparkle al color naranja y fijar el sentido del motor 2 en retroceso a una potencia del 25 %
 - d. Esperar 1 segundo

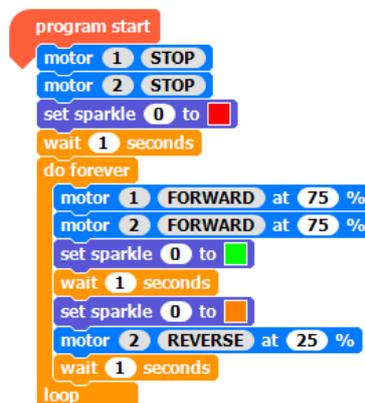


Figura 8. Ejemplo de programa con Crumble.

Hay seis categorías de comandos: Básico, Entrada/Salida, Destellos, Control, Variables y Operadores. Cada grupo contiene comandos que tienen algún tipo de relación.

La placa de Crumble es capaz de manejar dos motores de corriente continua de manera independiente. Esto se debe a que tiene dos puertos dedicados al control de motores.

Por otro lado, el Puerto D es capaz de manejar hasta 32 Sparkles independientemente cuando estos Sparkles se conectan en serie. Los Sparkles son placas de LED RGB (Rojo, Verde y Azul).

Además, los servomotores pueden ser gestionados por la controladora Crumble. La señal de control del servomotor tiene que estar conectada a los puertos A, B, C o D. Se pueden controlar hasta cuatro servomotores a la vez.

Crumble también se puede utilizar para realizar lecturas de un sensor de ultrasonidos. Un sensor de ultrasonidos es un dispositivo para medir distancias. Su funcionamiento se basa en el envío de un pulso de alta frecuencia, no audible por el ser humano. Este pulso

rebota en los objetos cercanos y es reflejado hacia el sensor, que dispone de un micrófono adecuado para esa frecuencia. El sensor se basa simplemente en medir el tiempo entre el envío y la recepción de un pulso sonoro. Se sabe que la velocidad del sonido es 343 m/s en condiciones de temperatura 20 °C, 50% de humedad, presión atmosférica a nivel del mar. Cuando se usa este sensor con Crumble, sólo hay que conectar un puerto de la controladora al disparador y otro puerto al eco del sensor. Mediante un sencillo comando de Crumble donde se le indica qué puerto está conectado al disparador y qué puerto está conectado al eco, el comando devuelve el valor de distancia a objeto en centímetros.

Finalmente, se puede conectar un módulo de seguimiento de línea a dos de los puertos A, B, C y D en la Crumble. Los sensores infrarrojos son unos componentes electrónicos compuestos normalmente de un LED infrarrojo y un fototransistor colocados uno al lado del otro, de forma que el LED actúa como emisor y el fototransistor como receptor. El LED infrarrojo emite luz infrarroja, es decir, de mayor longitud de onda (o menor frecuencia) que la que pueden ver los humanos, así que para nosotros es invisible. Si esta luz choca con una superficie blanca se reflejará y llegará al fototransistor. Si por el contrario golpea en una superficie negra, el material absorberá la mayoría de la luz y no llegará al fotorreceptor. Este sensor tiene 3 pines de conexión, + y - para la alimentación y OUT para enviar la señal a la Crumble que nos indicará si está llegando o no el reflejo del LED al fototransistor. Funciona de esta forma:

- Si está llegando la luz al fotorreceptor enviará una señal LOW al exterior.
- Si no está llegando enviará una señal HIGH.

La Figura 9 muestra una construcción básica de robot de ruedas usando Crumble. Este robot incluye una controladora Crumble, un portapilas como sistema de alimentación, dos motores y un Sparkle.

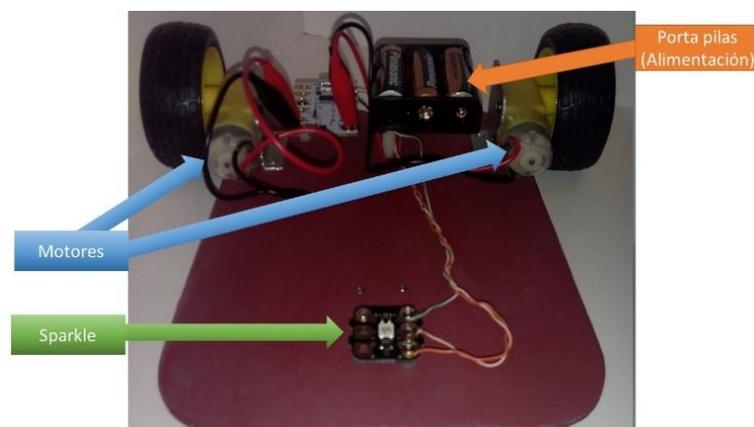


Figura 9. Ejemplo de robot con Crumble.

2.3.5. OTRAS HERRAMIENTAS EN TORNO A STEM

Para impartir educación científica y tecnológica, las instituciones de educación formal están utilizando diferentes herramientas robóticas educativas como LEGO Education (WeDo y Mindstorms), Fischertechnik y PICAXE. También se utilizan otras herramientas de desarrollo de software como App Inventor. Además, se incluyen herramientas de software tipo sketchbook como Processing, SketchUp y QCAD. Asimismo, el lenguaje de programación Python está incluido como herramienta tecnológica. Por último, otros robots como Sphero y Printbot Evolution BQ también se utilizan para la educación tecnológica. Estas herramientas requieren una computadora que está presente en la mayoría de las instituciones de educación formal.

LEGO Education lleva 35 años trabajando con profesores y especialistas en educación para ofrecer experiencias de aprendizaje lúdico. [Klassner, 2003] demuestra que LEGO Mindstorms se ha convertido en una plataforma adecuada para que los estudiantes universitarios investiguen una amplia gama de temas dentro del ACM/IEEE Computing Curriculum 2001. Actualmente, hay dos herramientas comunes proporcionadas por LEGO Education: LEGO Education WeDo y LEGO Mindstorms Education EV3. LEGO Education WeDo proporciona una manera fácil de desplegar ciencia, tecnología, ingeniería, matemáticas y conocimientos de codificación a los niños. Usando LEGO Mindstorms Education EV3, los estudiantes pueden desarrollar el pensamiento crítico y las habilidades para resolver problemas en la escuela secundaria. El kit de robótica LEGO Mindstorms Education EV3 se utilizó para construir una impresora braille DIY (Do It Yourself) [Beasley, 2014]. LEGO Mindstorms puede demostrar fácilmente el concepto de movilidad mediante un ejercicio en el que los estudiantes construyen un sigue líneas, tal y como se describe en [Jung, 2012]. El uso de los robots LEGO Mindstorms NXT como plataforma de entrenamiento, redujo significativamente el esfuerzo de un grupo de profesores a la hora de preparar los ejercicios, y estimuló la reutilización de su trabajo entre varios temas y materias [Cruz-Martín, 2012]. Los kits LEGO NXT Mindstorms y LabVIEW estándar se combinaron para trabajos de laboratorio y competiciones de estudiantes [Gomez-de-Gabriel, 2011]. [McKenna, 1998] describe un módulo didáctico que enfatiza el diseño integrador usando seis máquinas simples: la palanca, la rueda y el eje, la polea, el plano inclinado, el tornillo y el engranaje. Este sencillo módulo de máquina hace uso de un original entorno de aprendizaje multimedia basado en la web, así como de actividades de construcción offline y prácticas con el kit LEGO Technic I. La combinación de robótica

LEGO EV3 y diseño de juegos para desarrollar las estrategias de pensamiento computacional de los estudiantes de secundaria también es muy útil, [Leonard, 2016] lo demuestra.

La educación con Fischertechnik proporciona a los estudiantes y educadores las soluciones de aula necesarias para inspirar un plan de estudios STEM que sea atractivo y eficaz. Los sets de Fischertechnik STEM PREP y STEM Engineering están diseñados para ofrecer a las escuelas una integración ideal entre el aprendizaje basado en proyectos y los planes de estudio basados en estándares. Cada juego fomenta la investigación, el diseño y la resolución de problemas gracias al sistema de construcción único y versátil de Fischertechnik. [Xie, 2010] muestra cómo Fischertechnik ofrece a los estudiantes experimentos más completos y desafiantes. Se ha desarrollado una arquitectura novedosa, un sistema de operación y monitorización remoto de robots industriales con tres grados de libertad basado en Fischertechnik [Wang, 2012]. Las actividades cubiertas que se dirigen a la formación STEM con Fischertechnik son:

- Esbozo y documentación,
- Electricidad básica,
- Programación,
- Sistemas de detección y visión,
- Actuadores robóticos móviles,
- Fundamentos de automatización,
- y robótica.

Por otro lado, un microcontrolador PICAXE está diseñado para ser el cerebro de un proyecto electrónico. Originalmente fue diseñado como un sistema educativo para las escuelas. Hoy en día, el sistema PICAXE también está siendo ampliamente adoptado por los aficionados debido a su facilidad de uso. Los chips PICAXE son populares porque son de muy bajo coste y fáciles de programar usando software gratuito y fácil de aprender. El chip PICAXE puede reaccionar a los sensores de entrada. También puede activar y desactivar las salidas. Existen varios tamaños de chips PICAXE (8, 14, 18, 20, 28 y 40 pines). Por lo tanto, es un sistema flexible y escalable, se puede seleccionar el tamaño del chip según lo requiera el proyecto a desarrollar. En los circuitos integrados PICAXE casi todos los pines pueden ser configurados por el usuario para ser una salida digital, una entrada digital (encendido / apagado) o una entrada analógica (para la lectura de los valores de un sensor táctil, por ejemplo). Las características avanzadas tales como PWM, I2C, SPI, RS232 y la interfaz de 1-Wire también se puede lograr fácilmente si es necesario.

Durante el año 2010 surgió App Inventor. El equipo de App Inventor estuvo liderado por Hal Abelson y Mark Friedman del MIT. Se trata de un entorno de programación visual intuitivo que permite a todo el mundo - incluso a los niños - crear aplicaciones totalmente funcionales para Smartphones y tabletas. Aquellos que se inician en el uso de MIT App Inventor pueden lograr el desarrollo de una primera aplicación sencilla en menos de 30 minutos. App Inventor es una sencilla herramienta de programación para crear aplicaciones Android. La creación de una aplicación de App Inventor comienza en el navegador web, donde se diseña el aspecto de la aplicación. Luego, al igual que encajar las piezas del rompecabezas, se establece el comportamiento de la aplicación. Mientras tanto, a través de una conexión directa entre el ordenador y el teléfono, la aplicación se puede cargar en el teléfono. Con App Inventor se pueden crear muchos tipos diferentes de aplicaciones. A menudo la gente comienza por construir juegos como MoleMash o juegos que les permiten dibujar divertidos dibujos. Los sensores del teléfono también pueden ser gestionados. Por ejemplo, para mover una bola a través de un laberinto basado en inclinar el teléfono. Pero la creación de aplicaciones no se limita a juegos simples. También se pueden crear aplicaciones que informen y eduquen. Se puede hacer una aplicación para ayudar a los compañeros de clase a estudiar para un examen. Con las funciones de conversión de texto a voz de Android, el teléfono puede hacer las preguntas en voz alta. App Inventor, no está destinado a ser utilizado por desarrolladores profesionales. Esto se debe a que, en lugar de escribir código, el aspecto de la aplicación, y el comportamiento de la aplicación se especifica mediante el uso de bloques. En un caso de estudio sobre un curso de creación de juegos para estudiantes de secundaria en Beijing, App Inventor fue utilizado para el conocimiento cognitivo de física involucrado en la creación del juego [Chen, 2017]. [Chtourou, 2017] presenta una solución para facilitar la tarea de programación de gadgets electrónicos utilizando programación gráfica basada en App Inventor.

SketchUp fue desarrollado por la empresa @Last Software de Boulder, Colorado, cofundada en 1999 por Brad Schell y Joe Esch. SketchUp debutó en agosto de 2000 como una herramienta de creación de contenido 3D de uso general. Es un programa informático de modelado en 3D para una amplia gama de aplicaciones de dibujo como arquitectura, diseño de interiores, paisajismo, ingeniería civil y mecánica, diseño de películas y videojuegos. Está disponible como versión gratuita, SketchUp Make, y como versión de pago con funciones adicionales, SketchUp Pro. La frase "software CAD fácil" puede parecer una denominación errónea, pero en el caso de SketchUp Pro, es la verdad. El verdadero punto fuerte de SketchUp Pro es su facilidad de uso. Los iconos de gran tamaño y las barras de herramientas personalizables facilitan la tarea incluso a los principiantes en

CAD (Computer-Aided Design) sin quitarles las potentes herramientas que los profesionales necesitan para crear obras de arte. [Tang, 2010] describe cómo integrar orgánicamente el software en la enseñanza de arquitectura para guiar a los estudiantes a centrarse en el diseño de formas y espacios en el proceso de diseño utilizando SketchUp como herramienta de enseñanza. Los autores en [Chen, 2015] muestran que Google SketchUp (con todas sus funciones) es una herramienta de aprendizaje eficaz para ayudar a los estudiantes a desarrollar habilidades como la visión espacial.

QCAD es desarrollado por RibbonSoft. El desarrollo del QCAD comenzó en octubre de 1999, a partir del código de CAM Expert. CAM Expert, es un sistema CAM (Computer-Aided Manufacturing) para grabado y corte por láser. CAM Expert era capaz de disponer de suficientes funciones CAD como para que una derivación para un sistema CAD independiente tuviera sentido. QCAD es una aplicación de software gratuita de diseño asistido por ordenador para diseño y dibujo en 2D. Está disponible para Linux, Apple OS X, Unix y Microsoft Windows. El GUI (Graphical User Interface) de QCAD está basado en el framework Qt.

Python fue concebido a finales de la década de 1980 y su implementación comenzó en diciembre de 1989 por Guido van Rossum en Centrum Wiskunde & Informatica (CWI) en los Países Bajos como sucesor del lenguaje ABC (a su vez inspirado en SETL) capaz de manejar excepciones e interconectarse con el sistema operativo Amoeba. Hoy en día, Python es un lenguaje de programación de alto nivel ampliamente utilizado para la programación de propósito general. Python cuenta con un sistema de tipo dinámico y gestión automática de memoria y soporta múltiples paradigmas de programación, incluyendo estilos de programación orientados a objetos, imperativos, funcionales y procedimentales. Dispone de una amplia y completa biblioteca estándar. [Fraanje, 2016] presenta ejemplos del uso de Python en la educación de robótica y mecatrónica, incluyendo visión por computador (OpenCV), dinámica 3D (ODE/VPython), control (python-control), cinemática 3D y robótica (python-robotics) y aprendizaje automático (scikit-learn). Python también se utiliza para enseñar física computacional [Backer, 2007]. Además, [Jiménez, 2016] muestra los proyectos de los estudiantes resultantes de la aplicación de conceptos clave de procesamiento de imágenes y análisis de patrones soportados por la metodología PBL (Project Based Learning). Además, BlockPy, un entorno de programación Python basado en la web y de acceso abierto, da soporte a los programadores durante su iniciación en un contexto de ciencia de datos a través de una vista de programación dual de bloque/texto [Bart, 2017].

Printbot Evolution es suministrado por BQ como kit y plataforma de programación. BQ es una marca española dedicada al diseño, venta y distribución de lectores electrónicos, tabletas, smartphones, impresoras 3D y kits de robótica. Usando Printbot Evolution, un estudiante puede programar robots con facilidad. Esta gratificante actividad es la oportunidad perfecta para pasar tiempo y compartir el entusiasmo con los jóvenes. Construir un PrintBot Evolution ayuda a desarrollar la creatividad, el trabajo en equipo, el pensamiento crítico, así como las habilidades motoras y de diseño en 3D. Es una experiencia que va más allá de la diversión, Evolution es el primer paso para aprender y entender la tecnología.

2.4. Laboratorios remotos aplicados a STEM

En los últimos años se han lanzado algunas plataformas rentables como los laboratorios remotos [Carro, 2015]. Esta solución facilita la inclusión de plataformas robóticas para el desarrollo de actividades educativas donde el aprendizaje proactivo se potencia a través de experimentos en el mundo real. Laboratorios remotos como las experiencias expuestas en [Carro, 2014b] demuestran cómo se puede ayudar a los estudiantes a comprender con mayor facilidad conceptos abstractos en las asignaturas de ingeniería y ciencias, asociando estos conceptos abstractos a la vida cotidiana. De acuerdo con el tiempo de formación necesario para el profesorado, para obtener un conocimiento mínimo, de 10 a 16 horas de formación pueden ser suficientes. Sin embargo, cuando se pretende adquirir un conocimiento más profundo, se requiere una formación de 40 a 100 horas. Las competencias TIC (Tecnologías de la Información y la Comunicación) son un elemento clave para la formación de los profesores en TIC [Almerich, 2016].

El Laboratorio como Servicio y el Lenguaje de Descripción de Laboratorio se propone a la comunidad internacional como un prototipo potencial de estándar universal para la especificación y conexión de laboratorios remotos y virtuales y sus componentes asociados. La consideración de este proyecto de idea por parte de ingenieros, tecnólogos y científicos contribuirá al éxito a largo plazo del intercambio de laboratorios y fomentará la participación [Seiler, 2012]. El autor en [Hernández, 2012] muestra una nueva metodología de control de laboratorios remotos que integra y combina distintas tecnologías y aplicaciones en el campo de la experimentación remota.

Gracias a arquitecturas o middleware como el presentado por el autor en [San Cristóbal, 2010] se permite a los estudiantes utilizar una única solución para adquirir conocimiento

teórico y práctico. También es posible reutilizar los servicios ofrecidos por las plataformas educativas. Asimismo, se permite el uso de los estándares e-learning soportados por las plataformas educativas. Por ello es posible reutilizar laboratorios ya existentes.

El autor en [Orduña, 2014] explora y evalúa un nuevo modelo para la ampliación de esta compartición -permitiendo un mayor número de usuarios para acceder a equipos distribuidos en múltiples instituciones, y la gestión de escenarios más complejos que los descritos en la literatura compartir o subcontratar los accesos a los laboratorios o distribuirlos a través de dispositivos de bajo coste. Tanto el modelo de federación y la viabilidad técnica han sido implementadas en el WebLab-Sistema de gestión remota del laboratorio de Deusto, también descrito en este documento.

M2Learn se presenta como un marco de trabajo por el autor en [Martin, 2010] como una API abierta y fácil de usar que ayuda a los desarrolladores a crear nuevas aplicaciones de aprendizaje móviles y ubicuas y permite a los usuarios no expertos crear sistemas por sí mismos.

El autor en [Carro, 2014a] propone una metodología para simplificar la instalación y puesta en marcha de los laboratorios remotos de robótica. Se usa software y el hardware libre siempre que ha sido posible, o bien, desarrollar herramientas específicas utilizando siempre la filosofía de software y hardware libre.

Existen multitud de ejemplos de laboratorios remotos. El autor en [Pastor, 2006] muestra el uso de una pasteurizadora a escala de laboratorio de la casa ArmField, donde las variables a controlar son del tipo nivel, flujo y temperatura.

2.5. Conclusiones

La cuarta revolución industrial será para el aprendizaje altamente cualificado y altamente adaptable, flexible y continuo, con perfiles más multidisciplinares, con un currículo de competencias en Ciencia, Tecnología, Ingeniería y Matemáticas, ya sea en la enseñanza universitaria o en un perfil más técnico de la formación profesional, de hecho, está surgiendo una mayor demanda laboral. El trabajador del futuro debe tener competencias en el trabajo colaborativo y en equipos; Gestión del tiempo; Resolución de problemas; Razonamiento analítico; Capacidad para buscar, filtrar y priorizar la información; Y, sobre todo, para compartir decisiones. La baja cualificación, de los servicios para la sociedad, también tendrá un espacio recurrente.

La aparición de las nuevas tecnologías es tan rápida que resulta difícil gestionar su impacto en la sociedad. Los cambios tecnológicos afectan a todos los aspectos de la vida, desde la naturaleza del trabajo hasta lo que significa ser humano, y pueden ser abrumadores si se colabora para comprenderlos y gestionarlos.

Los puestos de trabajo del futuro requieren una formación que no se imparte ahora, mientras que los jóvenes necesitarán una sólida base teórica, una buena formación práctica y estar en contacto con el mundo del hardware y el software. La industria y los servicios deberían poder influir en los planes de estudios universitarios y de formación profesional para indicar los intereses de formación. En la actualidad, cada vez es más difícil encontrar perfiles de tipo junior con formación en estos dos idiomas. Por ello, la división Siemens Automation está trabajando en SCE (Siemens Automation Cooperates with Education). Siemens ofrece planes de estudio y formación en automatización basados en el Massive Open Online Course (MOOC) y en conceptos de aprendizaje mixto para transmitir conocimientos sobre Industry 4.0. Las instituciones educativas se benefician de condiciones especiales, apoyo y asociaciones [SCE, 2018].

La revolución tecnológica y digital en la que se encuentra inmersa la sociedad potencia el talento, como fórmula para que aprovechen todas las oportunidades que ofrece. Y es un talento que no tiene género, lo que facilita la entrada de las mujeres en la nueva industria 4.0, dándole un protagonismo que no ha disfrutado en décadas anteriores.

Los revolucionarios avances en inteligencia artificial, robótica, Internet de los objetos, vehículos autónomos, impresión en 3D, nanotecnología, biotecnología, ciencia de los materiales, almacenamiento de energía y computación cuántica están redefiniendo industrias enteras y creando otras nuevas. En el Foro Económico Mundial, esta ola de innovación fue llamada la "Cuarta Revolución Industrial" porque es un cambio fundamental en la forma en que vivimos, trabajamos y nos relacionamos.

El mundo está experimentando una transformación digital y tecnológica muy fuerte, y las empresas necesitan el mejor talento científico y tecnológico para aprovechar cada oportunidad. Y es un talento "sin género", ya que se demuestra que los equipos mixtos son más poderosos, generan más valor gracias a la diversidad y riqueza que aportan sus miembros.

A corto plazo, existen grandes oportunidades para el desarrollo industrial en los siguientes campos, que también tendrán un efecto de arrastre transversal de otras industrias (como la automoción, la construcción, la máquina-herramienta y el equipamiento industrial):

- Infraestructuras inteligentes de transporte (se pueden ver ejemplos en [Sjoberg, 2017] y [Mare, 2016]), Ciudades inteligentes (se pueden ver ejemplos en [Islam, 2017] y [Liu, 2017]), Nearshoring (se pueden ver ejemplos en [Zieris, 2013] y [Prechelt, 2013]), E-Salud (se pueden ver ejemplos en [Ma, 2015] y [Shen, 2014]), Industria 4.0 (se pueden ver ejemplos en [Wan, 2016] y [Wollschlaeger, 2017]), Big Data (se pueden ver ejemplos en [Yan, 2016] y [Basanta, 2016]), Cloud Computing (se pueden ver ejemplos en [Liu, 2017] y [Xie, 2017]), Construction and Smart and sustainable rehabilitation of homes and Digital Home (se pueden ver ejemplos en [Alam, 2012] y [Rodriguez-Diaz, 2016]), Cybersecurity (se pueden ver ejemplos en [Cheng, 2017] y [Hashem, 2017]), etc.
- Aplicaciones específicas, basadas en la electrónica de automoción (se pueden ver ejemplos en [Enisz, 2014] y [Dusmez, 2014]), aeroespacial (se pueden ver ejemplos en [Guo, 2017] y [Borg Bartolo, 2017]), redes eléctricas inteligentes (se pueden ver ejemplos en [Navarro, 2013] y [Kumar, 2017]), equipos portuarios y aeropuertos (se pueden ver ejemplos en [Matsumoto, 2016] y [Hirabayashi, 2015]), equipos médicos (se pueden ver ejemplos en [Flynn, 2015] y [Saleh, 2015]), Internet industrial (se pueden ver ejemplos en [Lade, 2017] y [Mumtaz, 2017]).
- Centros nacionales de referencia en Big Data, Cloud, tecnologías de diseño y arte digital, Smart Cities e impresión 3D.

Las herramientas educativas robóticas son oportunidades prometedoras para ser usadas en la educación STEM, pero una herramienta educativa STEM más compensada es todavía necesaria para ser incluida en las Instituciones de Educación Formal. La solución adecuada debería reducir los esfuerzos económicos, de formación y de comprensión. Además, una herramienta educativa STEM debe promover las siguientes competencias de los estudiantes: Trabajo en equipo, resolución de problemas, resiliencia y comunicación. Además, la creatividad también puede reforzarse con prototipos de ciencia ficción dentro de un marco educativo. Por lo tanto, se pueden diseñar y crear construcciones educativas robóticas disruptivas con el objetivo de mejorar las sesiones de aprendizaje. La educación robótica agrega campos de conocimiento STEM (Ciencia, Tecnología, Electrónica y Matemáticas) a las sesiones de aprendizaje de una manera fácil. El proceso de aprendizaje se empodera gracias a la inclusión de herramientas motivacionales mecánicas y robóticas [Carro, 2015]. Se pueden aplicar conceptos sencillos para llevar la tecnología a las escuelas y a los jóvenes que mañana se convertirán en ingenieros, científicos y ciudadanos curiosos [Carro, 2014]. El autor en [Lind, 2014] proporciona un marco de trabajo de software, denominado PyMoCo. Está totalmente implementado en

el lenguaje de programación interpretado de alto nivel Python, y permite un control de movimiento en tiempo real rápido y altamente flexible y la integración de aplicaciones de robots industriales.

Los gobiernos también deberían participar activamente en la introducción de innovaciones en la sociedad. Las autoridades deben trabajar en estrecha colaboración con los tecnólogos y empresarios que impulsan la revolución, para no quedarse atrás. Y los individuos deben estar informados, a fin de comprender y responder a los nuevos problemas que surgen de la compleja interacción entre la tecnología y la sociedad.

La Cuarta Revolución Industrial traerá consigo cambios sistémicos que exigirán un enfoque colaborativo y nos obligarán a pensar en nuevas formas de trabajar juntos en las esferas pública y privada. El ritmo del cambio no se detendrá, sino que se acelerará, de modo que la transparencia debe mantenerse en beneficio de todas las partes interesadas para que puedan sopesar los riesgos y los beneficios de cada nuevo desarrollo.

CAPÍTULO 3

3. PLATAFORMAS DE DESARROLLO Y SU USO

Actualmente, las tecnologías están evolucionando rápidamente. Por tanto, es fácil encontrar un gran número de estudiantes familiarizados con el manejo y la programación de dicha tecnología. Es bastante común que prácticamente cualquier estudiante del entorno de las áreas de conocimiento STEM (Science, Technology, Engineering and Mathematics) esté familiarizado con el uso y la programación de una o varias plataformas de desarrollo.

Con la gran variedad de plataformas de hardware abierto en la actualidad, existen infinidad de aplicaciones. Aunque, actualmente se pueden encontrar un gran número de ellas, esta sección se centrará en aquellas que son más conocidas y que han sido más relevantes para el estudio de esta tesis.

Actualmente, existe una amplia gama de plataformas de desarrollo. Estas pueden ser clasificadas de varias maneras diferentes. Según [Plaza, 2016b], las plataformas de desarrollo pueden clasificarse en cuatro grupos:

- Basado en Microcontroladores,
- Basado en Microprocesadores,
- Basado en FPGA (Field Programmable Gate Array), y
- Plataformas de Desarrollo Híbrido.

El movimiento de la IoT (Internet of Things) ha tenido un impacto en estas plataformas de desarrollo. Actualmente hay muchas aplicaciones basadas en IoT que se están desarrollando en muchos campos diferentes. Una de las aplicaciones encontradas a lo largo de la investigación es el uso de Raspberry Pi como plataforma para red de sensores acústicos. Se monitoriza la medida de molestia de ruidos producidos por el tráfico, tal y como se detalla en [Saraidevi, 2015]. También se ha encontrado la implementación de dos tipos de sistemas de reconocimiento basados en el uso de la voz. El diseño y programación se realiza de tal forma que puedan ejecutarse en un sistema embebido de bajo coste utilizando una Raspberry Pi para este fin. En [Haro, 2014] se describe este trabajo. Por otro lado, se usa Arduino Yún como base para la monitorización de la calidad del aire mediante el uso de sensores de temperatura, humedad, partículas orgánicas volátiles y partículas de polvo para ser consultado a través de Internet mediante PC o Smart Devices en [Joardar, 2015]. Laboratorios Virtuales y Remotos para Ingeniería de Control

han sido migrados de un PC a una plataforma Beaglebone Black por su bajo coste, soporte de comunidad de desarrolladores y aficionados, como puede verse en [Sáenz, 2015]. Los autores, en el trabajo expuesto en [Alexander, 2015] muestran una implementación de Laboratorios Remotos basado en Beaglebone Black. Se trata de un sistema de monitorización remoto con gran disponibilidad y fiabilidad. Una aproximación modular y extensible para monitorización de Sistemas de Distribución basada en Beaglebone Black como alternativa hardware de bajo coste para los Operadores de Sistemas de Distribución (DSOs) es expuesto en [Ferdowsi, 2015]. Por último, en [Ferdowsi, 2015] se muestra una red de sensores visuales compuesto por nodos basados en Beaglebone Black para reducir el consumo energético siendo esta una plataforma de consumo eficiente. Se incluye un estudio que aporta el impacto de la cobertura espaciotemporal de los nodos empleados.

Otros ejemplos son [Xu, 2014] en servicios médicos de emergencias, [Nastic, 2013] en computación en la nube, [Pruet, 2015] en loET (Internet of Educational Things) en zonas rurales desfavorecidas y [Carro, 2015a] en laboratorios remotos educativos. Según el autor en [Stankovic, 2014], surgen nuevos problemas de investigación debido a la gran escala de dispositivos, la conexión de los mundos físico y cibernético, la apertura de los sistemas y los continuos problemas de privacidad y seguridad.

A lo largo de este capítulo se analizarán en profundidad las siguientes plataformas de desarrollo de IoT:

- El Shield WiFi de Arduino proporciona a la placa Arduino ([Arduino, 2018a]) una conexión inalámbrica a Internet [Arduino, 2018b]. No puede funcionar en modo autónomo. Por lo tanto, esta placa requiere un microcontrolador para interactuar con otros elementos.
- El Shield Yún trae fácilmente las características de Yún a las placas Arduino y Genuino. Es una buena opción para proyectos de IoT que utilizan conexión inalámbrica para acceder a Internet [Arduino, 2018c].
- Arduino Yún Linino OS es la placa perfecta para usar cuando se diseñan dispositivos conectados en el contexto de proyectos de Internet of Things. Esta placa combina la potencia de uso de Linux con la facilidad de uso de Arduino [Arduino, 2018d].
- Genuino MKR1000 es una potente placa que combina la funcionalidad del Arduino Zero y el Shield WiFi de Arduino. Es la solución ideal para los creadores que deseen diseñar proyectos de IoT con una experiencia previa mínima en la creación de redes [Arduino, 2018e].

- NodeMcu ESP8266 es un kit de desarrollo de hardware y firmware de código abierto que ayuda a la creación de prototipos de productos de IoT dentro de unas pocas líneas de script Lua [NodeMcu, 2018a].
- Onion OMEGA es una plataforma de desarrollo de hardware con WiFi incorporado y un completo Sistema Operativo Linux [Onion, 2018a].
- El dispositivo Particle Photon es un pequeño kit de desarrollo WiFi para crear proyectos y productos conectados para la Internet de los objetos. Es fácil de usar, es poderoso y está conectado a la nube [Particle, 2018a].

También se ha incluido un análisis detallado de algunas plataformas de robótica educativa Wireless, que bien podrían ser usadas en combinación con la plataforma de robótica educativa que se describe a lo largo de esta tesis. La herramienta propuesta está basada en plataformas de desarrollo.

3.1. Plataformas para aplicaciones IoT basadas en Arduino

Arduino es un proyecto de software y hardware de código abierto. Arduino está soportado por una comunidad internacional, cada día más grande, de usuarios que diseñan y fabrican dispositivos y objetos interactivos [Arduino, 2018f]. En el momento de escribir esta tesis, existen cuatro placas de desarrollo proporcionadas por Arduino con propósitos de IoT: Shield WiFi de Arduino, Shield Arduino Yún, Arduino Yún Linino OS y Genuino MKR1000. Un ejemplo de uso relacionado con IoT de Arduino puede ser la implementación de un banco de pruebas para 6LoWPAN, tal y como se describe en [Al-Kaseem, 2017]. Los autores en [Satija, 2017] demuestran la viabilidad de la implementación en tiempo real de una propuesta en el contexto de IoT SQA (Signal Quality-Aware) utilizando Arduino.

Las placas Arduino están destinadas a Estados Unidos, mientras que Genuino es la marca hermana de los productos que se venden fuera de Estados Unidos.

3.1.1. ARDUINO WIFI SHIELD

El Shield WiFi de Arduino proporciona una conexión inalámbrica a las placas Arduino. La conexión se establece siguiendo unas sencillas instrucciones para conectar sensores y actuadores a través de Internet [Arduino, 2018b]. Esta placa presenta las siguientes características:

- La tensión de funcionamiento es de 5 Vdc que debe ser suministrado por la placa Arduino.

- Es compatible con Arduino Due.
- Microcontrolador: Atmel AT32UC3.
- SRAM: interno 64 KB.
- Flash: interno 512 KB.
- Conexión vía: Redes 802.11b/g.
- Tipos de cifrado: WEP y WPA2 Personal.
- Conexión con Arduino en el puerto SPI (Serial Peripheral Interface).
- Ranura microSD integrada.
- Cabeceras ICSP (In-Circuit Serial Programming).
- Conexión FTDI para depuración en serie del Shield WiFi.
- Mini-USB (Universal Serial Bus) para actualizar el firmware del WiFi.
- Está basado en la LAN inalámbrica HDG204 (red de área local) 802.11b/g System in-Package. Hay una ranura para tarjetas microSD, que se puede utilizar para almacenar archivos y permitir que estén disponibles en la red.

Los datos mecánicos de la placa son:

- Longitud: 63,2 mm.
- Ancho: 53.5 mm.

El coste de la placa Shield WiFi de Arduino, en el momento de escribir esta tesis, es de 69,00 € [Arduino, 2018g]. La Figura 10 (DFRobot, 2018) muestra el aspecto que tiene un Shield WiFi Arduino.



Figura 10. Shield WiFi de Arduino.

Los autores en [Bocci, 2014] detallan un ejemplo de aplicación con el Shield WiFi de Arduino. Este ejemplo, el ArduSiPM es un módulo compacto, rentable y fácilmente replicable para la lectura de detectores SiPM.

3.1.2. GENUINO YÚN SHIELD

Al igual que Shield WiFi de Arduino, Shield Yún de Arduino extiende la placa Arduino con la potencia de un sistema basado en Linux que permite conexiones de red y aplicaciones avanzadas [Arduino, 2018c].

A través del panel Web de Yún y el sketch de "YúnFirstConfig" es posible conectarse fácilmente a través de WiFi o red cableada (Ethernet) a la plataforma de desarrollo Genuino Yún Shield. Las preferencias del Shield y la carga de bocetos se pueden realizar directamente desde la placa Arduino/Genuino adjunta. Esta placa incluye las siguientes características:

- Procesador: Atheros AR9331.
- Arquitectura: MIPS (Microprocessor without Interlocked Pipeline Stages) a una frecuencia de reloj de 400MHz.
- Tensión de funcionamiento: 3,3 Vdc.
- Ethernet: IEEE 802.3 10/100Mbit/s.
- WiFi: IEEE 802.11b/g/n.
- USB Tipo A: Host 2.0.
- RAM: 64 MB DDR2.
- Memoria Flash: 16 MB.
- Ranura microSD integrada.
- Interfaz de la placa: SPI y HW Serial.

Los datos mecánicos de la placa son:

- Longitud: 68,6 mm.
- Ancho: 53,3 mm.

El precio de la plataforma Genuino Yún Shield, en el momento de escribir esta tesis, es de 39,90 € [Genuino, 2018a]. La Figura 11 (CNXSOFT, 2018) muestra el aspecto que tiene una plataforma Genuino Yún Shield.



Figura 11. Genuino Yún Shield.

En el artículo [Bruneo, 2017] los autores se centran en la virtualización de placas inteligentes, implementando una solución flexible para placas Arduino basada en Stack4Things, que permite crear, multiplexar, migrar e implementar placas virtuales en contextos de IoT-Cloud.

3.1.3. ARDUINO YÚN LININO OS

Arduino Yún Linino OS es una placa microcontroladora basada en el microcontrolador ATmega32u4 y el procesador Atheros AR9331. Esta placa incluye una distribución Linux basada en OpenWrt llamada Linino OS. Las comunicaciones Ethernet y WiFi son soportadas por hardware. También se incluyen el puerto USB-A y la ranura para tarjetas microSD [Arduino, 2018d]. Las características del microcontrolador AVR Arduino son las siguientes:

- Microcontrolador: Atmel ATmega32U4.
- Tensión de funcionamiento: 5 Vdc.
- Tensión de entrada: 5 Vdc.
- Pines de E/S digitales: 20.
- Salida PWM (modulación por ancho de pulso): 7.
- Pines de E/S analógicas: 12.
- Corriente continua por pin de E/S: 40 mA en los pines de E/S; 50 mA en los pines de 3,3 Vdc.
- Memoria Flash: 32 KB (de los cuales 4 KB son utilizados por bootloader).
- SRAM: 2,5 KB.
- EEPROM: 1 KB.
- Velocidad de reloj: 16 MHz.

Las características del procesador Arduino son las siguientes:

- Procesador: Atheros AR9331.
- Arquitectura: MIPS.
- Tensión de operación: 3.3 Vcc.
- Ethernet: 802.3 10/100Mbit/s.
- WiFi: 802.11b/g/n 2.4 GHz.
- Tipo USB: 2.0 Host.
- Lector de tarjetas: microSD.
- RAM: 64 MB DDR2.
- Memoria Flash: 16 MB.

- SRAM: 2,5 KB.
- EEPROM: 1 KB.
- Velocidad de reloj: 400 MHz.

Los datos mecánicos de la tarjeta son:

- Longitud: 68,6 mm.
- Ancho: 53,3 mm.

El coste de la placa Arduino Yún Linino OS, en el momento de escribir esta tesis, es de 52.00 € [Arduino, 2018h]. La Figura 12 (buildelectroniccircuits, 2018) muestra el aspecto que tiene una plataforma Arduino Yún Linino OS.



Figura 12. Arduino Yún Linino OS.

El trabajo [Saraladevi, 2015] describe un ejemplo de uso con la plataforma Arduino Yún Linino OS. El trabajo propuesto por los autores contiene un robot móvil de cuatro ruedas autónomo controlado por la plataforma Arduino Yún.

3.1.4. GENUINO MKR1000

Genuino MKR1000 ha sido diseñado para ofrecer una solución práctica y rentable para proyectos que requieren conectividad WiFi. Está basado en el Atmel ATSAMW25 SoC (System on Chip). Este procesador forma parte de la familia SmartConnect de dispositivos Atmel Wireless [Arduino, 2018e]. La familia SmartConnect está diseñada específicamente para proyectos de IoT.

El ATSAMW25 se compone de tres bloques principales:

- SAMD21 Cortex-M0+ 32bit ARM MCU de baja potencia.
- WINC1500 WiFi de bajo consumo a 2,4 GHz IEEE® 802.11 b/g/n.
- ECC508 CryptoAuthentication.

El ATSAMW25 incluye también una única antena PCB (Printed Circuit Board) de 1x1.

La tarjeta incluye un circuito de carga de Li-Po que permite el uso de una batería de Li-Po como fuente de alimentación externa. Además, se permite una fuente de alimentación externa de 5 Vdc. Internamente, el MKR1000 conmuta automáticamente desde ambas fuentes de alimentación. Esta placa presenta las siguientes características:

- Microcontrolador: SAMD21 Cortex-M0+ 32bit baja potencia ARM MCU.
- Fuente de alimentación de la placa (USB/VIN): 5 Vdc.
- Batería soportada: Una célula de Li-Po, 3.7 Vdc, 700mAh mínimo.
- Tensión de Operación del Circuito: 3.3 Vdc.
- Pines de E/S digitales: 8.
- Pines PWM:12.
- Interfaces serie: UART (Receptor/Transmisor Asíncrono Universal), SPI e I2C (Inter-Integrated Circuit).
- Pines de entrada analógica: 7 (ADC 8/10/12 bit).
- Pines de salida analógica: 1 (DAC 10 bit).
- Interrupciones externas: 8.
- Corriente DC por pin de E/S: 7 mA.
- Memoria Flash: 256 KB.
- SRAM: 32 KB.
- EEPROM: no.
- Velocidad de reloj: 32.768 kHz (RTC), 48 MHz.
- LED_BUILTIN: 6.
- Dispositivo USB de alta velocidad y Host integrado.

Los datos mecánicos de la placa son:

- Longitud: 65,0 mm.
- Ancho: 25,0 mm.

El coste de la placa Genuino MKR1000, en el momento de escribir esta tesis, es de 31,99 € [Genuino, 2018b]. La Figura 13 (Arduino, 2018) muestra el aspecto que tiene una plataforma Arduino Yún Linino OS.



Figura 13. Genuino MKR1000.

[Ogu, 2017] muestra un ejemplo de uso de la plataforma Genuino MKR1000. Esta plataforma se utiliza como elemento bifuncional; para coordinar las actividades del sistema y conectar el sistema a una red WiFi y posteriormente a Internet.

3.2. NodeMCU ESP8266

NodeMCU es una plataforma de IoT de código abierto. El término NodeMCU se refiere al firmware. El lenguaje de programación Lua se utiliza para programar la placa. Se basa en el proyecto eLua. El kit de desarrollo está basado en ESP8266, integra GPIO (General Purpose Input Output), PWM, IIC, 1-Wire y ADC en la misma placa [NodeMcu, 2018a].

Las características del NodeMCU ESP8266 son las siguientes:

- Fuente de alimentación de la placa: 5 Vdc desde el puerto USB o 3,3 Vdc desde el puerto VIN.
- Convertidor serial USB-TTL.
- 10 GPIO, cada GPIO puede ser PWM, I2C, 1-Wire.
- Antena PCB.

Las características del microprocesador ESP8266 son las siguientes:

- CPU RISC de 32 bits: Tensilica Xtensa LX106 funcionando a 80 MHz.
- RAM: 64 KB de instrucción RAM, 96 KB de datos RAM.
- Flash externo QSPI (Quad Serial Peripheral Interface) - 512 KB a 4 MB.
- WiFi: IEEE 802.11 b/g/n.
- Interruptor TR integrado, balun, LNA, amplificador de potencia y red correspondiente.
- Autenticación WEP o WPA/WPA2, o redes abiertas.
- 16 pines GPIO.
- SPI, I2C.
- Interfaces I2S (Inter-IC Sound) con DMA (compartiendo pines con GPIO).
- En la GPIO2 se puede habilitar UART en pines dedicados, además de un UART sólo para transmisión.
- ADC: 1 10-bit.

Los datos mecánicos del tablero son:

- Longitud: 38,0 mm.
- Ancho: 25,0 mm.

El coste de la placa NodeMCU ESP8266, en el momento de escribir esta tesis, es de 7,95 € [NodeMCU, 2018b]. La Figura 14 (tiendateo, 2018) muestra el aspecto que tiene una plataforma NodeMCU ESP8266.



Figura 14. NodeMCU ESP8266.

En el trabajo presentado en [Barai, 2017] se utilizaron dos NodeMCU ESP8266. Cada uno de ellos actuaba como nodo sensor, como punto de acceso o como estación durante las pruebas que detallan los autores.

3.3. Onion OMEGA

Esta placa incluye un WiFi incorporado, es compatible con Arduino y en su interior se ejecuta Linux. Permite la creación de prototipos de hardware utilizando herramientas familiares como Git, pip, npm. Se pueden utilizar lenguajes de programación de alto nivel como Python, Javascript, PHP. El Onion OMEGA está totalmente integrado con el Onion Cloud con el objetivo de crear aplicaciones de Internet of Things. Es Código Abierto. El procesador es el Qualcomm Atheros AR9331 SoC [Onion, 2018a].

Con el SoC y los componentes que incluye, la Onion OMEGA tiene las siguientes características:

- Fuente de alimentación de la placa: 5 Vdc desde el puerto USB o 3,3 Vdc desde el puerto VIN.
- Procesador MIPS 24Kc MIPS a 400 MHz.
- 64 MB de RAM DDR2 a 400 MHz.
- 16 MB de memoria flash integrada.
- Soporte para USB 2.0.
- Soporte para Ethernet a 100 Mbps.
- 802.11b/g/ WiFi a 150 Mbps.
- 18 GPIOs digitales.
- Un único UART serie.
- Soporte para SPI.

- Soporte para I2S.

Los datos mecánicos de la placa son:

- Longitud: 42,7 mm.
- Ancho: 26.4 mm.

El coste de la placa Onion OMEGA, en el momento de escribir esta tesis, es de 19.99 \$ [Onion, 2018b]. Si se utiliza [Conversión, 2018] para la conversión de moneda de dólares de los Estados Unidos a euros, el coste de la plataforma es de 17,94 €. La Figura 15 (Onion, 2018) muestra el aspecto que tiene una plataforma Onion OMEGA.



Figura 15. Onion OMEGA.

3.4. Particle PHOTON

Esta placa se basa en la arquitectura WICED (Wireless Internet Connectivity for Embedded Devices) de Cypress. El Particle Photon combina un microcontrolador STM32 ARM Cortex M3 y un chip Cypress WiFi. Se puede programar o gestionar mediante el uso de: Particle Local IDE, Particle Web IDE, Particle Cloud o Particle Device Management Console. El lenguaje de programación es C++ y está soportado por una amplia gama de librerías que reducen el tiempo de desarrollo [Particle, 2018a].

Esta placa presenta las siguientes características:

- Fuente de alimentación de la placa: 5 Vdc desde el puerto USB o 3,6-5,5 Vdc desde el puerto VIN.
- Módulo WiFi Particle PØ.
- Chip WiFi Broadcom BCM43362.
- WiFi 802.11b/g/n.
- STM32F205RGY6 Corteza ARM M3 de 120 MHz.
- Flash de 1 MB, 128 KB de RAM.
- LED de estado RGB integrado (unidad externa incluida).

- 18 GPIO de señal mixta y periféricos avanzados.
- Pines de E/S digitales: 18.
- Pines de entrada analógica: 8 (ADC).
- Pines de salida analógica: 2 (DAC).
- Pines PWM: 7.
- Interfaces serie: SPI, I2C, I2S y UART.
- Diseño de código abierto.
- Sistema operativo en tiempo real (FreeRTOS).
- Configuración de Soft AP.
- Certificados por FCC, CE e IC.

Los datos mecánicos de la placa son:

- Longitud: 36,6 mm.
- Ancho: 20.3 mm.

El coste de la plataforma Particle Photon, en el momento de escribir esta tesis, es de 19.99 \$ [Particle, 2018b]. Si se utiliza [Conversión, 2018] para la conversión de moneda de dólares de los Estados Unidos a euros, el coste de la plataforma es de 18,87 €. La Figura 16 (Onion, 2018) muestra el aspecto que tiene una plataforma Particle Photon.



Figura 16. Particle Photon.

3.5. Comparativa de acuerdo con el hardware de procesamiento

Dentro de la categoría de plataformas basadas en Arduino/Genuino para aplicaciones IoT hay dos tipos: las plataformas tipo Shields y plataformas que no requieren de otros dispositivos externos para funcionar. De esta manera, las plataformas tipo Shield requieren un microcontrolador adicional con el objetivo de interactuar con otros elementos como sensores o actuadores que son ampliamente utilizados en la educación robótica. Por otro lado, Arduino Yún Linino OS, Genuino MKR1000, NodeMCU, Onion OMEGA y

Particle Photon pueden utilizarse sin necesidad de ser conectadas a un dispositivo de procesamiento interactuar con otros elementos como sensores o actuadores. La Tabla 1 resume el microcontrolador y el procesador para cada tarjeta.

Tabla 1. Dispositivo de procesamiento de tarjeta de desarrollo IoT.

Placa de desarrollo IoT	Microcontrolador	Procesador
Arduino WiFi Shield	Atmel AT32UC3	No
Genuino Yún Shield	No	Atheros AR9331
Arduino Yún Linino OS	Atmel ATmega32U4	Atheros AR9331
Genuino MKR1000	SAMD21 Cortex-M0+	No
NodeMCU ESP8266	No	Tensilica Xtensa LX106
Onion OMEGA	No	Big-Endian
Particle Photon	ARM Cortex M3	No

Los microcontroladores son comúnmente empleados para tareas sencillas. Sin embargo, un microprocesador es capaz de:

- Realizar múltiples tareas, con capacidad para ejecutar aplicaciones grandes y genéricas, a diferencia del microcontrolador.
- Pueden combinarse con periféricos externos, RAM y almacenamiento hasta los límites deseados.
- Se suelen usar para aplicaciones donde las tareas son inespecíficas, tales como el desarrollo de software, juegos, sitios web, edición de fotos, creación de documentos y demás.
- Asimismo, tienen una velocidad de reloj del orden de los GHz, mientras que los microcontroladores tienen una velocidad de reloj alrededor de los 30-50 MHz.

Cuando un microcontrolador y un microprocesador están integrados en la misma tarjeta, las ventajas de ambos se combinan. Por lo tanto, la plataforma es capaz de realizar operaciones en tiempos muy reducidos. Además, es posible conectar cualquier tipo de dispositivo directamente, ya sean sensores, o bien actuadores. Por otro lado, un microcontrolador y un microprocesador integrados en la misma plataforma implican un desarrollo complejo desde el punto de vista de la sincronización entre ambos dispositivos y en relación con el intercambio de información.

La Figura 17 muestra un ejemplo de Shield Yún de Arduino apilado en un Arduino MEGA y un NodeMCU incluyendo una interfaz Shield para el control de motores y servos.

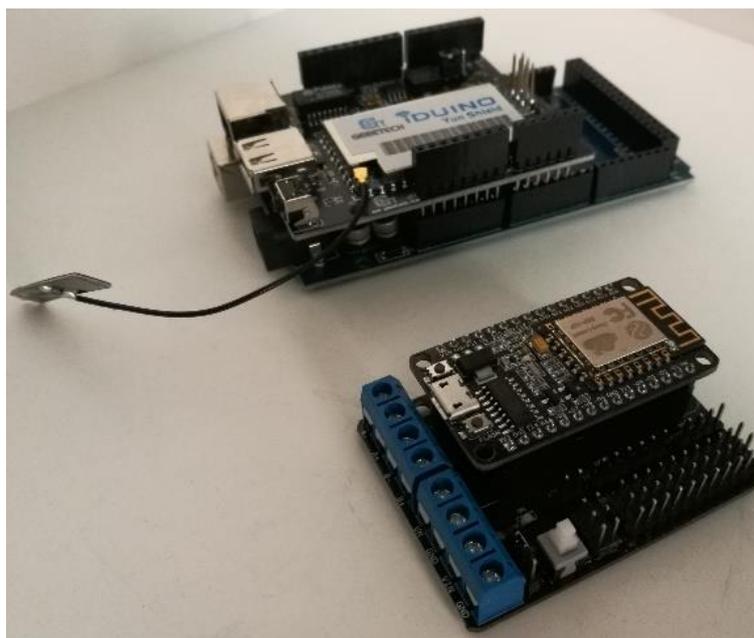


Figura 17. Plataformas Shield Yún de Arduino y NodeMCU.

3.6. Comparativa de acuerdo con el sistema de alimentación

Con el fin de alimentar las tarjetas, es importante saber cuál es el nivel de tensión de cada una para adaptar estos niveles de la tarjeta a otros dispositivos conectados. En la Tabla 2 se enumeran las tarjetas de desarrollo IoT y la tensión para la fuente de alimentación y las interfaces de los puertos de entrada y salida.

Tabla 2. Alimentación y puertos de interfaz de tarjeta de desarrollo IoT.

Placa de desarrollo IoT	Alimentación	Tensión de Entrada/Salida
Arduino WiFi Shield	5 Vdc externamente.	No hay interfaz de Entrada/Salida
Genuino Yún Shield	3.3 Vdc	No hay interfaz de Entrada/Salida
Arduino Yún Linino OS	5 Vdc	5 Vdc
Genuino MKR1000	5 Vdc o Única celda de Li-Po, 3.7 Vdc, 700 mAh mínima	3.3 Vdc
NodeMCU ESP8266	5 Vdc desde USB o 3.3 Vdc desde VIN	3.3 Vdc
Onion OMEGA	5 Vdc desde USB o 3.3 Vdc desde VIN	3.3 Vdc
Particle Photon	5 Vdc desde USB o 3.6-5.5 Vdc desde VIN	3.3 Vdc

Una vez identificado el valor de tensión de alimentación de la plataforma, hay que decidir el tipo de fuente de alimentación que le proporcionará la energía.

Actualmente, hay una gran variedad de mecanismos para alimentar una placa. Desde las simples baterías formadas por pilas tradicionales de 1,5 Vdc hasta los PowerBank que han surgido en los últimos años como medio de carga para los Smartphones cuando no se tiene disponible una toma de corriente eléctrica. Estos dispositivos proporcionan una fuente de alimentación estable de 5 Vdc.

Otras fuentes de alimentación extendidas con diferentes valores de tensión de salida dependiendo de cómo estén conectadas son las baterías LiPo (polímero de litio). Estas baterías pueden proporcionar 3.7 Vdc o un múltiplo de 3.7 Vdc. Este tipo de baterías tienen aproximadamente cuatro veces la densidad de energía de las baterías de níquel cadmio o níquel metal hidruro. Sin embargo, hay varios problemas de seguridad y desventajas con respecto a las baterías LiPo. Se ha descubierto que la sobrecarga y el sobrecalentamiento de las baterías pueden provocar un incendio.

Por todo esto, se debe elegir el sistema de alimentación que mejor se adecúe al tipo de aplicación para el que va a ser destinada la plataforma de IoT.

Normalmente los periféricos conectados a las tarjetas de desarrollo funcionan a 5 Vdc. Los motores DC, servos y sensores trabajan con un nivel de alimentación de 5 Vdc. No obstante, otro tipo de periféricos requieren un nivel de tensión de 3,3 Vdc para ser alimentados o interconectados. El módulo Bluetooth y las pantallas táctiles TFT son algunos ejemplos.

3.7. Comparativa de acuerdo con la memoria disponible

Otra característica importante para el desarrollo es la memoria disponible. Las tarjetas presentadas incluyen diferentes tipos de recursos de memoria, la Tabla 3 compila qué tipo de memoria - volátil y no volátil - está disponible y cuánta memoria se puede utilizar.

Tabla 3. Recursos de memoria volátil y memoria no volátil de tarjeta de desarrollo IoT.

Placa de desarrollo IoT	Memoria volátil	Memoria no volátil
Arduino WiFi Shield	SRAM interna: 64 KB	Internal Flash: 512 KB On-board micro SD slot
Genuino Yún Shield	RAM: 64 MB DDR2	Flash: 16 MB On-board micro SD slot

Placa de desarrollo IoT	Memoria volátil	Memoria no volátil
Arduino Yún Linino OS	Internal SRAM: 2.5 KB RAM: 64 MB DDR2	Internal EEPROM (2x): 1 KB Internal Flash (2x): 32 KB Flash: 16 MB On-board micro SD slot
Genuino MKR1000	SRAM interna: 32 KB	Flash interna: 256 KB
NodeMCU ESP8266	Internal RAM: 64 KB for in- structions Internal RAM: 96 KB for data	QSPI Flash: 512 KB to 4 MB
Onion OMEGA	RAM: 64 MB DDR2	Flash: 16 MB
Particle Photon	RAM: 128 KB	Flash: 1 MB

Hay dos tipos de memoria: memoria volátil y memoria no volátil. La primera, normalmente, es usada por los microprocesadores, mientras que la otra es usada comúnmente por los microcontroladores.

La memoria volátil, a diferencia de la memoria no volátil, es una memoria que requiere energía para mantener la información almacenada. Por lo tanto, retiene su contenido mientras está encendido, pero cuando se interrumpe la alimentación, los datos almacenados se pierden muy rápida o inmediatamente.

Hay dos tipos de memoria RAM (Random-Access Memory): estática y dinámica. Ambos tipos presentan ventajas y desventajas en cuanto a velocidad, costes y tamaño físico de los módulos RAM. Por un lado, la memoria RAM estática les aporta a los microcontroladores mayor velocidad que la RAM dinámica. Por el contrario, la memoria RAM estática necesita un mayor espacio en la placa que una memoria RAM dinámica de la misma capacidad. Esto se debe a que la RAM estática necesita de cuatro a seis transistores para almacenar un poco de información, mientras que la RAM dinámica sólo necesita uno para para almacenar la misma cantidad de información.

Por otro lado, la memoria no volátil se utiliza normalmente para la tarea de almacenamiento secundario o almacenamiento persistente a largo plazo. Las transacciones de información para memorias volátiles son más rápidas que las transacciones de información de memorias no volátiles.

3.8. Comparativa de acuerdo con las comunicaciones

En aplicaciones comunes, las tarjetas de IoT deben interactuar con otros dispositivos utilizando una conexión por cable. Estas comunicaciones se realizan normalmente mediante una interfaz serie. Además, los puertos digitales y analógicos se utilizan para leer

los valores de los sensores o interactuar con algún tipo de actuadores. La Tabla 4 resume las interfaces serie y de puerto que se pueden realizar para cada tarjeta de IoT.

Tabla 4. Puertos serie y puertos discretos de tarjeta de desarrollo IoT.

Placa de desarrollo IoT	Interfaces serie	Puertos discretos
Arduino WiFi Shield	SPI, USB, ICSP, UART y FTDI	No
Genuino Yún Shield	SPI, USB, ICSP y UART	No
Arduino Yún Linino OS	SPI, I2C, USB, ICSP Ethernet y UART	Pines E/S digitales: 20 Salida PWM: 7 Pines E/S analógicos: 12
Genuino MKR1000	SPI, I2C, y UART	Pines E/S digitales: 8 Salida PWM: 12 Pines de entrada analógicos: 7 (ADC 8/10/12 bit) Pines de salida analógicos: 1 (DAC 10 bit)
NodeMCU ESP8266	SPI, I2C, I2S y UART	Pines E/S digitales: 10 (PWM, I2C, 1-wire) Pines de entrada analógicos: 1 (ADC 10 bit)
Onion OMEGA	SPI, I2S y UART	Pines E/S digitales: 18
Particle Photon	SPI, I2C, I2S, USB y UART	E/S de propósito general: 18 Pines E/S digitales: 18 Salida PWM: 9 Pines de entrada analógicos: 8 Pines de salida analógicos: 2

Con el fin de comunicarse con otros dispositivos, los microprocesadores y microcontroladores son capaces de implementar diferentes protocolos. De la simplicidad a la complejidad, las comunicaciones se pueden clasificar de la siguiente manera:

- Comunicaciones básicas utilizando GPIO: esta comunicación no está estandarizada. No hay una velocidad de transmisión y recepción definida y no hay un flujo de datos estándar. Este tipo de comunicación se utiliza normalmente para el control del motor o para la adquisición de datos simples del sensor, como la distancia medida por un sensor de infrarrojos. Adicionalmente, PWM es un uso específico de la comunicación básica comúnmente usada con el objetivo de manejar servos o adquirir distancia medida de un sensor de distancia de ultrasonido. Este tipo de comunicación es la más sencilla de implementar y de usar.
- Comunicaciones serie: hay muchas comunicaciones serie estandarizadas. Todas ellas fueron liberados para reducir el número de puertos de E/S necesarios. Las

comunicaciones serie están estandarizadas con un conjunto de reglas sobre las funciones de los puertos y cómo se formatean e intercambian los datos. Ejemplos de comunicaciones seriales son: UART, USB, SPI, I2C, I2S y Ethernet. Suele ser un tipo de comunicación más complejo que las comunicaciones anteriores.

- Programación de comunicaciones: es una forma particular de comunicación destinada a transmitir el programa a la memoria de configuración del microprocesador. ICSP es una comunicación de programación que incluye la Tabla 4. Este tipo de comunicaciones rara vez son gestionadas por los usuarios de las plataformas, ya que cada fabricante particulariza este tipo de comunicación para su plataforma. Sólo usuarios expertos de plataformas suelen trabajar con este tipo de comunicaciones.

Los microprocesadores y microcontroladores incluyen interfaces serie estándar para facilitar la comunicación con otros dispositivos. Por otro lado, los periféricos como los sensores o las memorias son compatibles con algunas comunicaciones serie.

Los tipos de interfaz de comunicación serie más utilizadas son las siguientes:

- UART: El Receptor/Transmisor Asíncrono Universal no es un protocolo de comunicación como SPI e I2C, pero es un circuito físico en un microcontrolador basado en el protocolo RS-232. Su objetivo principal es transmitir y recibir datos en serie.
- I2C: El bus Inter-Integrated Circuit es una interfaz patentada desarrollada por Philips Semiconductors. Es un bus semidúplex, síncrono, multi-master que requiere sólo dos cables de señal: datos (SDA) y reloj (SCL).
- SPI: El Serial Peripheral Interface (SPI) es un bus serie síncrono desarrollado por Motorola y presente en muchos de sus microcontroladores. El bus SPI consta de cuatro señales: MOSI (Master Out Slave In), MISO (Multiple-Input Single-Output), reloj serie (SCK) y selección activo nivel bajo del esclavo (/SS).
- USB: El Universal Serial Bus es ampliamente extendido desde simples periféricos de computadora como un teclado hasta dispositivos complejos como teléfonos inteligentes. Se desarrolló como un nuevo medio para conectar un gran número de dispositivos al PC (ordenador personal) y, finalmente, para sustituir los puertos "heredados". El USB es una comunicación asíncrona que utiliza dos puertos para la alimentación y dos puertos más para el intercambio de datos.
- I2S: El estándar de sonido Inter-IC fue introducido en 1986 por Philips. Es un estándar de interfaz utilizado para conectar dispositivos de audio digital entre sí. El bus I2S consta de al menos tres señales: Línea de reloj de bits (SCK o BCLK), línea de reloj de Word (WS) y una línea de datos multiplexada (SD). También puede

incluir dos líneas más: Reloj maestro (LRCLK) y una línea de datos multiplexada para la carga

- Ethernet: Fue desarrollado en Xerox PARC entre 1973 y 1974. En febrero de 1980, el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) inició el proyecto 802 con el objetivo de estandarizar la LAN (redes de área local). Ethernet utiliza cables trenzados de dos pares (TX+/TX- y RX+/RX-) con conectores RJ-45.

3.9. Comparativa de acuerdo con el método de programación

Una herramienta de programación es un programa de software o utilidad que ayuda a los desarrolladores de software o programadores a crear, editar, depurar y mantener una tarea de desarrollo específica. Una herramienta de programación es también conocida como una herramienta de desarrollo de software. En desarrollos de productos tecnológicos, el software es un factor clave para su éxito en el mercado. Elementos como la programación de la plataforma y el lenguaje de desarrollo de aplicaciones son factores igual de importantes que el hardware utilizado. La Tabla 5 indica el software de programación de la plataforma de desarrollo IoT y el lenguaje utilizado para su programación.

Tabla 5. Herramientas y lenguajes de programación de tarjeta de desarrollo IoT.

Placa de desarrollo IoT	Herramienta de programación	Lenguaje de programación
Arduino WiFi Shield	Arduino IDE	C++
Genuino Yún Shield	Arduino IDE	C++
Arduino Yún Linino OS	Arduino IDE	C++
Genuino MKR1000	Arduino IDE	C++
NodeMCU ESP8266	Lua over Espressif SDK, Non-OS SDK, RTOS SDK, Non-OS SDK SSL y Arduino IDE	Lua y C++
Onion OMEGA	Onion console y Command line	C y Python
Particle Photon	Online IDE, Particle Dev y Particle CLI	C++

Arduino proporciona una herramienta de programación de código abierto y fácil de usar, para escribir código y subirlo a la placa Arduino. Arduino IDE también incluye un monitor serie para interactuar con la placa programada.

Lua sobre Espressif SDK (Software Development Kit) emplea una arquitectura basada en eventos y orientada a tareas para la programación a nivel de aplicaciones. El SDK sin

OS permite incrustar funciones o eventos a través de temporizadores y funciones de devolución de llamada. RTOS (Real Time Operating System) SDK facilita la operación multitarea. Por lo tanto, este sistema permite a los usuarios utilizar la interfaz estándar para la gestión de recursos, el funcionamiento cíclico, el tiempo de retraso dentro de una tarea y la entrega y sincronización de información entre tareas. SSL de Non-OS SDK es como el SDK de Non-OS, aunque la conexión se basa en SSL (Secure Sockets Layer).

La Consola Onion es un escritorio virtual Web que da acceso a varias aplicaciones que permiten la interacción con la plataforma Onion OMEGA de una forma más visual que la línea de comandos. La línea de comandos de Onion es una poderosa herramienta para desarrollar y construir proyectos basados en Onion OMEGA. Pese a ello, siempre es más amigable usar una interfaz visual durante el desarrollo que la línea de comandos.

Para trabajar con la plataforma Particle PHOTON se dispone del Particle Web (IDE en línea) y el Desktop IDE (Particle Dev). Estas herramientas comparten los mismos iconos y características conectadas a la nube. Por lo tanto, es fácil hacer la transición entre ellos. Hay muchas librerías y ejemplos de firmware que se pueden usar en línea en Build que son compatibles con el IDE Local. Además, los dispositivos Particle pueden ser administrados, el firmware puede ser compilado y las actualizaciones del firmware pueden ser desplegadas de manera remota. El IDE de escritorio proporciona la confianza de almacenar todos los archivos localmente y la comodidad de Particle Cloud. La CLI (Command-Line Interface) proporciona una alternativa a la interfaz con la placa Particle cuando se conecta a un ordenador a través de un cable USB.

3.10. Comparativa de acuerdo con las dimensiones

El tamaño de una plataforma de desarrollo suele tener un impacto mínimo en el robot, hasta que se destina a controlar pequeños robots o vehículos aéreos no tripulados. La robótica educativa se basa comúnmente en pequeños robots de dos o cuatro ruedas. Además, se utilizan otros robots más complejos, como los brazos robóticos, con fines educativos.

Por lo tanto, el tamaño del conjunto es una característica importante que debe tenerse en cuenta. Por eso las dimensiones de la plataforma son una especificación importante para cualquier aplicación relacionada con robótica educativa. Ninguna de las placas analizadas es de gran tamaño, la más grande tiene una longitud de 68.6 mm y un ancho de 53.3 mm - Genuino Yún Shield y Arduino Yún Linino OS - y la más pequeña tiene un tamaño de 36.6 mm y un ancho de 20.3 mm - Particle Photon. La Tabla 6 incluye una comparación de las dimensiones de cada una de las plataformas de desarrollo para IoT.

Tabla 6. Dimensiones de tarjeta de desarrollo IoT.

Placa de desarrollo IoT	Longitud	Anchura
Shield WiFi Arduino	63,2 mm	53,5 mm
Genuino Yún Shield	68,6 mm	53,3 mm
Arduino Yún Linino OS	68,6 mm	53,3 mm
Genuino MKR1000	65,0 mm	25,0 mm
NodeMCU ESP8266	38,0 mm	25,0 mm
Onion OMEGA	42,7 mm	26,4 mm
Particle Photon	36,6 mm	20,3 mm

La Figura 18 incluye un ejemplo de un robot de 4 ruedas controlado por un Shield Yún de Arduino apilado sobre una placa Arduino Leonardo.

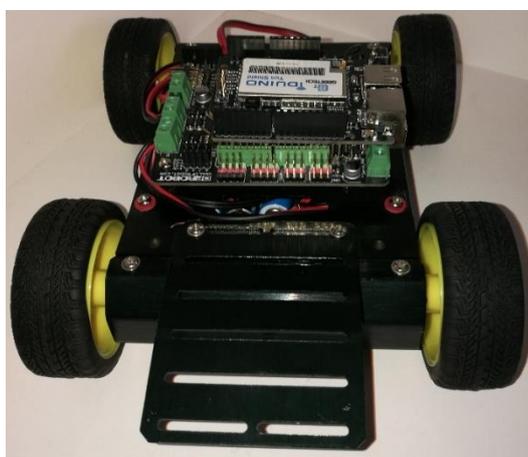


Figura 18. El Shield de Arduino Yún controla un robot de 4 ruedas.

La Figura 19 incluye un ejemplo de un robot de 6 grados de libertad controlado por una placa NodeMCU. El brazo del robot tiene integrados 6 servos que son accionados por un escudo.

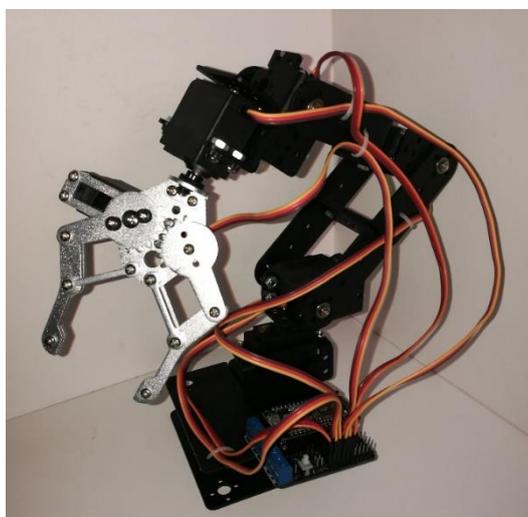


Figura 19. NodeMCU controlando un brazo robot de 6 grados de libertad.

3.11. Comparativa de acuerdo con el coste

Asimismo, se incluye en el análisis el coste asociado a las plataformas. Debido a la financiación del proyecto, el coste es un aspecto importante que debe tenerse en cuenta. Como se puede ver, ninguna de ellas es especialmente cara, la placa más cara cuesta 69,00 € - Shield WiFi Arduino - y la más barata 7,95 € - NodeMCU ESP8266. Todos ellos son asequibles para la mayoría de los proyectos. La Tabla 7 especifica el coste de las plataformas de desarrollo de aplicaciones IoT.

Tabla 7. Coste de tarjeta de desarrollo IoT.

Placa de desarrollo IoT	Coste
Arduino WiFi Shield	69,00 €
Genuino Yún Shield	39,90 €
Arduino Yún Linino OS	52,00 €
Genuino MKR1000	31,99 €
NodeMCU ESP8266	7,95 €
Onion OMEGA	17,94 €
Particle Photon	18,87 €

El coste es un factor importante para la selección de la plataforma de desarrollo de IoT. Cuando la plataforma está destinada a ser desplegada con fines de aprendizaje en las escuelas. Las actividades de aprendizaje suelen realizarse en aulas formadas por al menos 20 alumnos. Si la plataforma es usada de forma individual por los alumnos, el coste de las plataformas necesarias por aula se multiplica por 20 para las actividades individuales. Por lo tanto, el coste del material relacionado con las plataformas de desarrollo es un factor importante. Si se utiliza el NodeMCU ESP8266, se requiere un presupuesto de 159 €, por ejemplo. Cuando se utiliza la plataforma Arduino Yún Linino, esto implica un presupuesto de 1040 €.

3.12. Comparativa de acuerdo con el soporte de las comunidades

Por último, pero no por ello menos importante, otro aspecto importante es disponer de soporte durante las fases de desarrollo. Tradicionalmente, los fabricantes proporcionaban algún tipo de canal de comunicación como puede ser el teléfono o el correo electrónico con este fin. Hoy en día, la mayoría de los fabricantes incluyen un foro con el objetivo de dar soporte a sus clientes. Estos foros se mantienen por personal experto de la empresa y los clientes. Todos ellos forman una comunidad para el producto correspondiente. Las comunidades han demostrado y sigue demostrando ser de gran utilidad.

A continuación, se mostrará que las placas o dispositivos IoT descritos anteriormente tienen una gran comunidad que les da soporte:

- Arduino WiFi Shield, Arduino Yún Shield, Arduino Yún Linino OS y Arduino MKR1000 son soportados por la comunidad Arduino [Arduino, 2018f].
- El NodeMCU ESP8266 es soportado por dos comunidades: Comunidad NodeMCU [NodeMCU, 2018c] y comunidad ESP8266 [ESP8266, 2018].
- Onion OMEGA es apoyada por la comunidad Onion [Onion, 2018c].
- Particle Photon es apoyado por la comunidad Particle [Particle, 2018c].

3.13. Uso de las plataformas por la comunidad científica

Cuando se inicia un desarrollo es importante obtener referencias sobre las cosas que otras personas han hecho y la forma en que están utilizando las tarjetas de desarrollo. Los proyectos de Arduino se extienden ampliamente a la comunidad científica, profesionales y aficionados:

- [Orha, 2014] presenta una red de área corporal, para la adquisición de datos relacionados con la posición del cuerpo y algunos movimientos simples basados en un WiFi Shield apilado en un Arduino ChipKIT MAX32.
- Por otro lado, un robot móvil autónomo fue diseñado usando la placa Arduino Yún, a la que se le permite viajar en un entorno dinámico junto con la planificación del camino y la capacidad de evitar obstáculos, tal y como se describe en [Saraladevi, 2015].
- [Thaker, 2016] describe una red de sensores WiFi de bajo coste basada en ESP8266.
- La monitorización de los datos del sensor de pulso cardíaco en la nube con el ESP8266 se detalla en [Škraba, 2016].
- Utilizando una de las plataformas de desarrollo de IoT descritas, los robots móviles pueden utilizarse como laboratorios remotos para enseñar informática de una manera similar a la descrita en [Lopes, 2016].
- Puede utilizarse una guía con el objetivo de emplear IoT como herramienta TEL (Technology Enhanced Learning). Además, se pueden elaborar laboratorios sencillos usando el modelo detallado en [Charlton, 2016].
- Asimismo, hay trabajos como el que describen los autores en [de la Guía, 2016] donde se utilizan tecnologías del tipo wearable para la introducción al IoT.

- Otro ejemplo es el uso de la IoT para apoyar la visión de Smart City [Zanella, 2014].
- Existe un potencial de las tecnologías de sensores en red a gran escala para las ciudades inteligentes en la era de los grandes datos [Ang, 2017].

3.14. Plataformas de robótica educativa Wireless

Actualmente, Internet es utilizado para conectar personas o con el objetivo de acceder a una amplia gama de información y compartir sus ideas. Tal y como se describe en el trabajo [Lin, 2012], en el ámbito de IoT (Internet of Things), los objetos pueden identificarse y conectarse. Después de eso, intercambiarán información y tomarán decisiones por sí mismos.

Existen multitud de aplicaciones basadas en IoT. Estas aplicaciones se están desarrollando en una gran diversidad de campos. Un ejemplo es [Boyi, 2014] en el campo de los servicios médicos, donde se demuestra cómo recopilar, integrar e interoperar datos de IoT de manera flexible para prestar apoyo a los servicios médicos de emergencia. Además, [Nastic, 2013] propone un novedoso modelo de programación que facilita una forma intuitiva y eficiente de desarrollar aplicaciones de IoT basadas en una plataforma en nube.

También se pueden encontrar aplicaciones educativas basadas en la IoT. Como se afirma en [Pruet, 2015] y [Carro, 2015a]. En la primera referencia, mediante el uso de dispositivos de bajo coste, los estudiantes ubicados en zonas rurales desfavorecidas pueden acceder a un aprendizaje interactivo, en cualquier momento y en cualquier lugar. En el segundo se observa que los laboratorios sencillos pueden conectarse fácilmente como elementos de la IoT que aplican herramientas educativas.

Todos los ejemplos mostrados presentan un común denominador, basan las comunicaciones entre dispositivos en la comunicación inalámbrica. Este tipo de comunicación puede definirse como "la transferencia de información entre dos o más puntos que no están conectados por un conductor eléctrico". Las comunicaciones inalámbricas permiten a los estudiantes moverse libremente por el área de la clase. Esto se debe a la ausencia de cables que dificulten la movilidad de los estudiantes en el aula. Otra ventaja es la reducción de la complejidad cuando se pretende incluir un nuevo dispositivo. También contribuye a incrementar el número de dispositivos que se pueden añadir en los diversos escenarios educativos planteados. Además, las redes inalámbricas promueven un área de trabajo ordenada, limpia y ordenada para los estudiantes.

Actualmente, las comunicaciones inalámbricas más populares se basan en: WiFi, Radio Frecuencia y Bluetooth. Ac continuación se enumeran ejemplos de uso de este tipo de comunicaciones inalámbricas:

- La comunicación inalámbrica es especialmente útil como parte de los laboratorios robóticos educativos porque proporciona flexibilidad en el aula, como se muestra en [Carro, 2012a].
- Con el objetivo de convertirse en estudiantes interesados en la ingeniería como campo de estudio, [Newberry, 2015] presenta un sensor inalámbrico como herramienta educativa para el uso de una plataforma domótica.
- Lin, 2012] es una contribución en el campo de los laboratorios en línea basados en la web que proporciona una solución para los estudiantes de pregrado. We-FiLab facilita la inclusión de experimentos de redes inalámbricas que permiten a los estudiantes hacer experimentos en la página web a través de Internet en cualquier momento y en cualquier lugar.
- Las modificaciones del robot del Boe-Bot Parallax fueron realizadas para incluir comunicaciones Bluetooth para ser usadas en clase y los resultados de la programación inalámbrica simultánea múltiple sobre Bluetooth son descritos en [Rueberg, 2015].
- Las escuelas y los cursos de ingeniería pueden promover el aprendizaje colaborativo con la solución descrita en [Longo, 2015] utilizando su entorno de aprendizaje propuesto. El mencionado trabajo presenta el denominado EDM (Electromagnetic Device-Mediated Monitoring) como una plataforma para la detección oportuna de multitudes móviles de niveles de campo electromagnético.
- Además, [Dobrilovic, 2015] es un ejemplo de plataforma de hardware de código abierto que utiliza la red RFID/ZigBee y que muestra diferentes formas para los estudiantes durante los ejercicios de laboratorio.

Los dispositivos móviles han extendido su presencia en los últimos años de una manera vertiginosa. La mayoría de los estudiantes llevan sus propios dispositivos al aula. Las comunicaciones inalámbricas significan libertad para moverse por la clase o el laboratorio. Un Smartphone suele incluir WiFi y Bluetooth. Por lo tanto, los estudiantes podrían utilizar sus propios dispositivos durante la clase para aprovechar su uso como parte del proceso de aprendizaje.

En el marco de las comunicaciones inalámbricas, el uso del Bluetooth es muy común, ya que una hay una amplia gama de dispositivos que se puede conectar a través de este

tipo de comunicación. Además, desarrollar aplicaciones basadas en Bluetooth no es muy complicado.

Se pueden encontrar multitud de plataformas que incluyen al menos un canal de comunicaciones Bluetooth. También hay un gran número de proyectos basados en Bluetooth como medio de comunicación inalámbrica. Varios fabricantes promueven el movimiento Maker proporcionando soluciones que están destinadas a ser utilizadas por aficionados, instructores y empresarios. Los fabricantes se definen en [Axup, 2014] como importante grupo de miembros en el mundo de la fabricación.

Un ejemplo de este tipo de fabricantes es DFRobot, un proveedor de robótica y hardware de código abierto, que atiende a ingenieros de hardware, entusiastas del bricolaje y diseñadores interactivos. El DFRobotShop Rover v2 Bluetooth Kit es un robot versátil y programable. Se puede controlar fácilmente usando un Smartphone [DFRobotShop, 2018]. La Figura 20 (DFRobot, 2015) muestra una imagen de DFRobotShop Rover v2 Bluetooth Kit. En la parte superior del robot se puede ver el módulo Bluetooth.

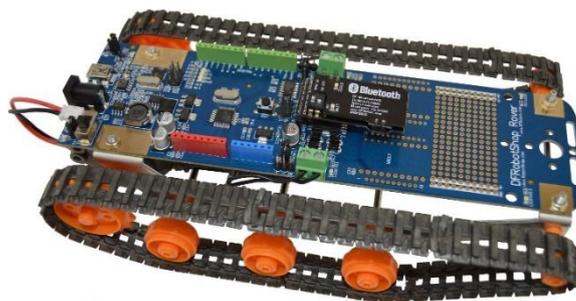


Figura 20. Kit Bluetooth DFRobotShop Rover v2.

Otro ejemplo de plataforma robótica basada en Bluetooth es el Starter Robot Kit que proporciona Makeblock. Makeblock es una plataforma de construcción de código abierto para convertir ideas en éxito. Combina una gran cantidad de nueva tecnología. El Makeblock Starter Robot Kit es una plataforma robótica que se puede adquirir con una configuración Bluetooth. Este Kit es otro ejemplo que facilita el aprendizaje de robótica, electrónica y programación de Arduino [Makeblock, 2018]. La Figura 21 (Makeblock, 2015) muestra el Makeblock Starter Robot Kit.

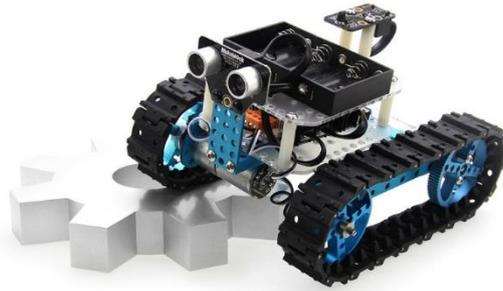


Figura 21. Kit de iniciación a la robótica Makeblock.

Makeblock también proporciona mBot - STEM Educational Robot Kit for Kids como una solución todo en uno para que los niños disfruten de la experiencia práctica sobre programación, electrónica y robótica [mBot, 2018]. La Figura 22 (Makeblock, 2015) es un ejemplo de la forma de proporcionar mBot - STEM Educational Robot Kit. En este caso, se trata de un robot de 2 ruedas.



Figura 22. Makeblock mBot – Kit robótico STEM para niños.

El robot móvil autónomo con red Wireless con cámara i90, zoom de alta resolución y movimiento horizontal es una plataforma robótica para investigadores que desarrollan aplicaciones robóticas avanzadas tales como monitorización remota, telepresencia y navegación/patrulla autónoma [i90, 2018]. La Figura 23 (Dr. Robot, 2015) muestra el robot móvil autónomo en red inalámbrico con cámara Pan-Tilt-Zoom de alta resolución i90.



Figura 23. Robot móvil autónomo con red Wireless.

La referencia [Florez, 2014] muestra un IDE (Entorno de Desarrollo Integrado) de Android que permite a los estudiantes utilizar el controlador y la planta en el mismo dispositivo. Existe otro modo de uso basado en la tecnología inalámbrica Bluetooth como puente para llevar la información de la planta al controlador. En este modo, el estudiante puede obtener información de la planta para verificar si el controlador implementado es efectivo en el mundo real.

Smartphone y Raspberry Pi están conectados a través de WiFi para gestionar el brazo robot desde Internet [Premkumar, 2015]. En este caso, el Smartphone puede interactuar con el robot a través de Android App.

Los ejemplos mencionados son implementaciones basadas en una placa externa que se incluye externamente en la placa principal de una manera sencilla. [DFRobotShop, 2018] incluye la tarjeta DFRobot Bluetooth V3. Makeblock, 2018] y [mBot, 2018] se basan en el mismo módulo Bluetooth.

Otra alternativa es un Arduino Bluetooth Shield. Esta solución integra un módulo Serial Bluetooth en forma de Arduino Shield para compatibilidad con Arduino y fácil montaje.

LEGO Education se esfuerza por animar a los estudiantes a pensar de forma creativa, razonar sistemáticamente y liberar su potencial para dar forma a sus propias ideas futuras MINDSTORMS LEGO también es capaz de conectarse a través de una conexión Bluetooth tanto en NXT como en EV3 Bricks [LEGO, 2018]. Esto proporciona una conexión inalámbrica entre los robots LEGO y los estudiantes.

La Figura 24 (LEGO MINDSTORMS, 2015) ilustra el vínculo entre un ordenador portátil con un dongle Bluetooth y un procesador LEGO MINDSTORMS EV3.



Figura 24. Conexión Bluetooth LEGO MIDSTORMS.

National Instruments (NI) es también un proveedor de dispositivos para la educación. NI myRIO [myRIO, 2018] es un dispositivo para la enseñanza de controles, robótica, mecatrónica y conceptos integrados. Utiliza la última tecnología Zynq de Xilinx con un FPGA (Field Programmable Gate Array) integrado con un procesador que ejecuta un sistema

operativo en tiempo real. También incluye una capacidad WiFi incorporada. Por lo tanto, los estudiantes pueden transferir datos e implementar código de forma inalámbrica.

Una aplicación de robot de 4 ruedas puede verse en la Figura 25 (Mazvydas Mark Narvidas, 2015). Donde se encuentra la plataforma de desarrollo de NI myRIO en la parte superior del robot.

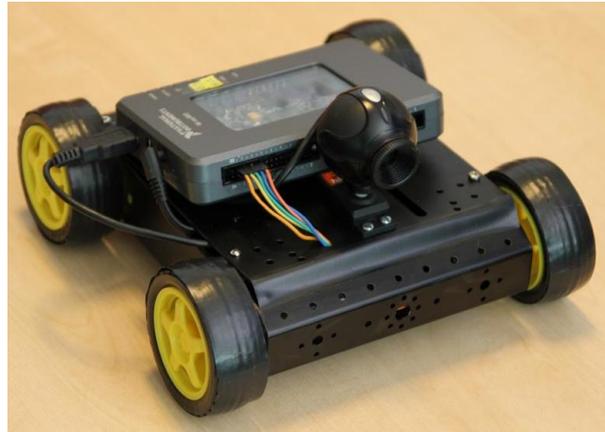


Figura 25. Robot de cuatro ruedas basado en NI myRIO.

Las herramientas mencionadas anteriormente y las otras que no se mencionan son comúnmente usadas por una comunidad creciente de Creadores. Estos creadores trabajan generalmente fuera de los límites de las actividades curriculares establecidas de educación en ingeniería [Jordania, 2014]. Por lo tanto, las plataformas como las mencionadas en esta sección pueden permitir a los estudiantes aplicar los conocimientos aprendidos en el aula y hacer.

3.15. Conclusiones

Tras comparar los diversos aspectos de las siete plataformas de desarrollo de IoT y algunas de las aplicaciones que pueden construirse con ellos. Se puede concluir que cualquiera de las alternativas presentadas es una opción adecuada para que los estudiantes, el profesorado, los profesionales y los aficionados realicen su proceso de aprendizaje, enseñen contenidos y trabajen a cualquier velocidad, coste y tiempo (ver secciones 3.5, 3.6, 3.7, 3.8, 3.9, 3.10 y 3.11). Además, cualquiera de estas plataformas cuenta con el apoyo de las comunidades del foro, tal y como se ve en la sección 3.12. Asimismo, se han incluido una serie de ejemplos de plataformas de robótica educativa Wireless.

Cualquiera de estas placas de desarrollo de IoT puede utilizarse en las aulas o de forma remota con el objetivo de facilitar la inclusión de la robótica en un contexto STEM. Además, permiten realizar aplicaciones caseras enmarcadas en un contexto de bricolaje DIY (Do It Yourself). Estos dispositivos pueden integrarse fácilmente como PODVs (Power

Oriented Device) con el objetivo de ser integrados para comunicaciones H2H (human-to-human) e IoTs (IoT Devices) [Elhattab, 2017]. Asimismo, se pueden construir redes de Big Data para datos recopilados de sensores en un contexto de IoT [Ang, 2017].

CAPÍTULO 4

4. ESCENARIOS EDUCATIVOS Y PRUEBAS CON ALUMNOS

Con el objetivo de definir los escenarios educativos más adecuados para la plataforma, se desarrollaron y se celebraron un conjunto de actividades formativas con diferentes herramientas de robótica educativa. Estas actividades también se utilizaron para poder realizar una comparativas entre la plataforma desarrollada a lo largo de esta tesis doctoral y las herramientas ya existentes.

En este capítulo se detallarán las actividades, cursos y talleres, realizados para buscar los escenarios educativos más adecuados para la plataforma desarrollada en esta tesis.

4.1. Actividades realizadas en 2016/2017

A lo largo del curso académico 2016/2017 se llevaron a cabo una serie de actividades formativas cuyo objetivo fue introducir la robótica educativa:

1. Un taller en el que los asistentes utilizaron la herramienta Crumble
2. Un curso anual en el que los asistentes utilizaron las herramientas Scratch, Crumble y Arduino. Este curso anual se dividió en tres módulos, donde cada módulo estaba orientado al uso de cada una de las herramientas mencionadas.

4.1.1. TALLER DE CRUMBLE

Hoy en día, hay muchas soluciones en el campo de robótica educativa. Pero muchas de ellas suelen ser costosas o no tienen suficiente documentación para ser utilizadas por padres e hijos que participan en la misma actividad. Sin embargo, pensamos que la herramienta Crumble puede proporcionar una forma fácil y barata de aprender, disfrutar y comunicarse durante las sesiones de robótica educativa casera.

Crumble es un controlador fácil de programable y de usar. Su interfaz de programación utiliza un lenguaje de programación basada en bloques que facilita su uso a los niños menores de 14 años [Crumble, 2018]. La Figura 26 muestra un ejemplo de un kit de robot basado en Crumble. Esta plataforma robótica incluye el controlador Crumble, dos motores DC con ruedas, un Sparkle (LED RGB), un portapilas con interruptor integrado, diferentes elementos mecánicos y cables para conectar eléctricamente los componentes.

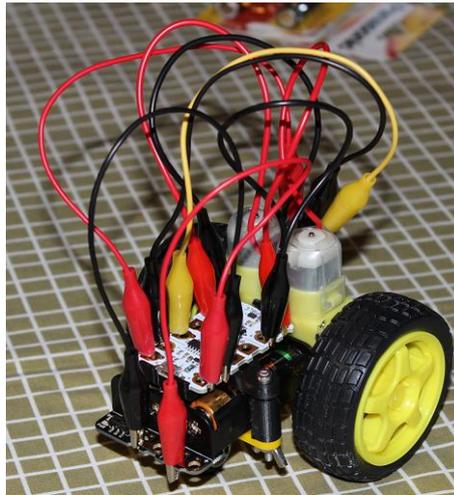


Figura 26. Kit robótico basado en Crumble.

Para comprobar que Crumble podía ser adecuado en el campo de la robótica educativa se realizó un pequeño taller robótico-educativo. Dicho taller se celebró en la tienda El Baúl de Alexandra [BAUL, 2016] y duró cuatro horas en las que se pusieron en práctica conceptos básicos sobre robótica. Los participantes del taller se dividieron en cuatro equipos. Donde cada equipo estaba compuesto por un adulto, de entre 35 y 45 años, y un niño, de entre 8 y 13 años que disponían de un kit robótico y ordenador portátil que les permitía realizar la tarea de programación.

Durante la sesión, cada equipo trabajó en las tareas de:

1. Montaje del robot
2. Control de motores
3. Sensor de seguimiento de línea
4. Sensor de detección de objetos
5. Y actividades relacionadas con la iluminación usando el Sparkle.

Al final del taller todos los participantes quedaron satisfechos con la sesión, el contenido y las cosas que eran capaces de hacer. Toda la información sobre los detalles del contenido del taller y los resultados obtenidos están descritos en [Plaza, 2017a].

4.1.2. CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2016/2017

Este curso tenía como objetivo principal introducir la robótica a niños de entre 7 a 15 años. Para ello, se valoró el uso de Scratch, de Crumble y de Arduino. Estas herramientas proporcionan a los estudiantes habilidades como el pensamiento sistémico, la mentalidad de programación, el aprendizaje activo, las matemáticas, la ciencia, el juicio y la toma de decisiones [Pickering, 2016], [Yoshihara, 2016], [Cápay, 2015].

El curso se realizó en el centro de La Estera [La Estera, 2018] que está ubicado en la localidad de Camarma de Esteruelas, zona este de la Comunidad de Madrid. La duración del curso fue de ocho meses y los contenidos se impartieron durante dos horas semanales. Cada herramienta seleccionada para el curso se usó en módulos de manera independiente. Así, Scratch fue utilizada durante las primeras 8 sesiones. Crumble se utilizó en las siguientes 8 sesiones. Por último, se usó Arduino durante las últimas 16 sesiones.

El grupo de estudiantes estaba formado por ocho estudiantes de 7 a 15 años. La Tabla 8 resume las calificaciones de los estudiantes se incluye la correspondencia entre el nivel educativo español y el nivel ISCED (Clasificación Internacional Normalizada de la Educación).

Tabla 8. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	7
Educación Secundaria Obligatoria	2 y 3	1

Los contenidos de las sesiones fueron diseñados con el objetivo de combinar teoría y práctica. La teoría se desplegó durante los primeros 20 minutos de la sesión y el resto del tiempo se dedicó exclusivamente a la resolución de problemas de forma práctica para aplicar los conceptos teóricos explicados anteriormente.

Con la idea de fomentar el trabajo colaborativo los alumnos trabajaron en parejas y las mesas se distribuyeron en forma de isla. Cada pareja disponía de un PC. Estos PC ejecutan Ubuntu como sistema operativo. Este sistema operativo fue elegido por su compatibilidad con Scratch, Crumble y Arduino, y porque tanto Ubuntu como Scratch no requieren muchos recursos como la velocidad del microprocesador o la memoria RAM (Random-Access Memory). Además, Ubuntu se puede utilizar de forma gratuita. Ubuntu es un sistema operativo basado en GNU/Linux que se distribuye bajo una licencia de software libre y se basa en un sistema gráfico basado en Unity. En el que se pueden instalar programas de software libre para fines educativos.

Módulo Scratch

El primer módulo en el que los alumnos utilizaron Scratch tuvo una duración de 8 sesiones (16 horas) y se dividió en dos partes:

1. La primera se centró en el conocimiento de la herramienta Scratch
2. Y la segunda se alineó con la construcción de aplicaciones mediante actividades de resolución de problemas.

A lo largo de este módulo se detectó un aumento Los estudiantes aumentaron su de motivación en las tareas de para programación y creación de proyectos sencillos. Al principio del módulo, todos los estudiantes solamente tenían conocimientos sobre nuevas tecnologías como ordenadores, tabletas y teléfonos inteligentes, pero estos conocimientos que se limitaban a consultas por Internet o juegos. Durante el módulo, se ampliaron sus habilidades sobre el pensamiento sistémico, la mentalidad de programación, el aprendizaje activo, las matemáticas, la ciencia, el juicio y la toma de decisiones, la buena comunicación, el diseño tecnológico, la resolución de problemas complejos y la persistencia.

También se detectó que El resultado de este trabajo muestra que es mejor hacerla estructura del módulo basado en plantear retos sencillos y en una combinación de teoría y práctica hizo que los alumnos disfrutaran del aprendizaje.

En conclusión, se ha demostrado que Scratch es una buena opción a la hora de introducir la robótica de una manera económica, sencilla y conveniente para que los profesores desarrollen conceptos escalables – ver más detalles en [Plaza, 2017d]. Además, los estudiantes encuentran la programación con Scratch una actividad agradable mientras adquieren conceptos STEM.

Módulo Crumble

El segundo módulo tenía como objetivo familiarizar a los niños y niñas con los elementos de hardware comúnmente utilizados en el campo de la robótica, tales como sistemas de energía, placas de control, luces, sensores y actuadores. A lo largo del segundo módulo, además de los ordenadores, anteriormente mencionados, los estudiantes tuvieron kits robóticos basados en Crumble, Figura 27. El precio de estos kits no es muy alto y tiene todos los elementos necesarios para mostrar el contenido del curso.

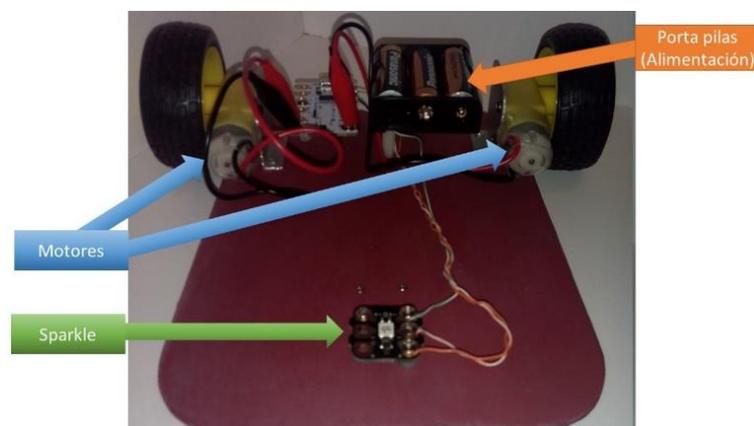


Figura 27. Ejemplo de robot con Crumble.

Al igual que en el módulo anterior, los estudiantes no tenían experiencia previa en el uso de la herramienta (Crumble), pero si tenían experiencia previa en programación gracias al módulo de Scratch.

El módulo tuvo una duración de dos meses y el contenido se impartió en 8 sesiones de 2 horas cada una, donde:

1. Las primeras cuatro sesiones se centraron en el aprendizaje de la herramienta Crumble
2. Y las cuatro últimas se centraron en la construcción de aplicaciones a través de actividades basadas en resolución de problemas.

Cada sesión se dividió en una pequeña orientación de los contenidos teóricos de la sesión y del reto propuesto. A lo largo de las sesiones se notó un cambio en las preguntas hechas por algunos de los estudiantes, tanto al conectar las experiencias de la actividad con ejemplos del mundo real como al tratar de entender los conceptos subyacentes.

Los estudiantes aumentaron su motivación para programar y crear proyectos simples. Además, la experiencia con Crumble facilitó a los estudiantes que se familiarizaran con componentes electrónicos sencillos. Los estudiantes también pudieron experimentar con componentes comúnmente usados en robótica industrial, como son los motores de corriente continua y diversos tipos de sensores.

En conclusión, también se ha demostrado que Crumble es una buena opción a la hora de introducir la robótica de una manera económica, sencilla y conveniente para que los profesores desarrollen conceptos escalables – ver más detalles en [Plaza, 2018f]. Además, los estudiantes encuentran el uso de Crumble una actividad divertida mientras aplican conceptos STEM.

Módulo Arduino

El tercer módulo tenía como objetivos del módulo centrado en el uso de Arduino era mejorar las habilidades adquiridas en mencionadas en los módulos anteriores y continuar profundizando en el conocimiento inherente a la robótica. Se utiliza como herramienta Arduino ya que permite gestionar muchas más entradas y salidas que Crumble. Además, Arduino permite implementar una infinidad de aplicaciones que no es posible realizar con Crumble debido a sus limitaciones.

En este módulo los estudiantes tuvieron kits robóticos como el que se muestra en la Figura 28. El precio de estos kits no es muy alto y tiene todos los elementos necesarios para mostrar el contenido del curso.

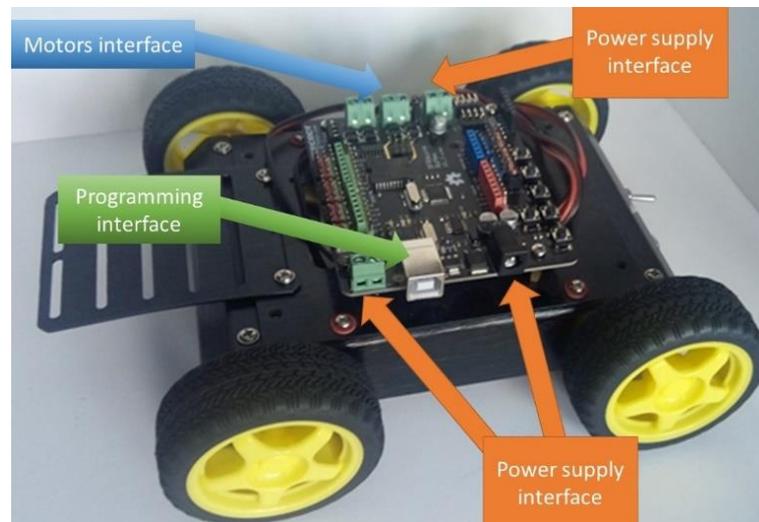


Figura 28. Ejemplo de robot con Arduino.

El módulo tuvo una duración de cuatro meses y el contenido se impartió durante 16 sesiones de dos horas cada una, donde:

- La primera parte se centró en el aprendizaje de la herramienta Arduino a través de un total de 8 sesiones
- y la segunda parte se alineó con la construcción de aplicaciones a través de actividades de resolución de problemas a lo largo de 8 sesiones.

Los contenidos de la sesión fueron diseñados con el objetivo de combinar teoría y práctica. La teoría se aplicó durante los primeros 40 minutos de la sesión y el resto del tiempo se dedicó exclusivamente a la resolución de problemas de forma práctica.

Al principio del módulo, los estudiantes no habían usado Arduino en ningún otro contexto, por lo que esta herramienta era completamente desconocida para ellos. A pesar de ello, todos los alumnos habían participado en cursos en los que se trabajó la introducción a la robótica con Scratch y Crumble.

Arduino es también una alternativa a considerar, ya que el software de Arduino es libre y compatible con la mayoría de los sistemas operativos. No requiere una gran cantidad de recursos informáticos. El controlador tiene un precio de 20 €. Es muy fácil de usar, pero requiere programación textual. Es fácil integrar hardware y componentes mecánicos para el desarrollo de aplicaciones. Es perfectamente válido a partir de los 12 años. Para las edades más jóvenes se ha hecho evidente que requiere una mayor

atención del instructor a los estudiantes, aunque esto no es una barrera insuperable – ver más detalles en [Plaza, 2018k].

Toda la información sobre los detalles del contenido del curso y los resultados obtenidos están descritos en [Plaza, 2017d], [Plaza, 2018d], [Plaza, 2018f] y [Plaza, 2018k].

4.2. Actividades realizadas en 2017/2018

A lo largo del curso académico 2017/2018 se llevaron a cabo una serie de actividades formativas cuyo objetivo fue introducir la robótica educativa. En este curso académico se usaron los conocimientos adquiridos en las actividades realizadas dentro del curso 2016/2017 para preparar nuevos contenidos. Se desplegaron tres talleres:

1. Un taller en el que los asistentes utilizaron la herramienta Scratch,
2. Otro con Crumble
3. Y un tercer taller con Scratch, Crumble y Arduino

También se desarrolló un curso anual similar al del 2016/17 pero con la diferencia del diseño de diferentes niveles de dificultad para el uso de Scratch, Crumble y Arduino.

4.2.1. TALLER SCRATCH DAY 2017

Este taller está dentro del movimiento mundial que ha promovido Scratch y donde diferentes países, regiones y localidades realizan talleres de un día basados en dicha herramienta. Y cuyo objetivo es dar a conocer el uso de esta herramienta en diferentes ámbitos o campos como la robótica.

El taller tuvo lugar en el centro comercial El Círculo [El Circulo, 2018], situado en la ciudad de Torrejón de Ardoz, zona este de la Comunidad de Madrid. Dicho taller fue apoyado, tanto a nivel logístico, como didáctico por:

- Incomaz (Tienda de Informática y Sonido) que dio apoyo publicitario y logístico a este taller [Incomaz, 2018].
- Plaza Robótica [Plaza Robotica, 2018]
- La Rama de Estudiantes de la UNED IEEE
- La Escuela Industrial de Ingeniería de la UNED,
- El Departamento de Ingeniería Eléctrica e Informática (DIEEC)
- Y la Escuela de Doctorado de la UNED

El grupo de estudiantes estaba formado por veinte estudiantes. Tres estudiantes eran menores de 10 años. Casi la mitad de los estudiantes tenían entre 10 y 15 años. Un estudiante tenía entre 16 y 18 años. Dos estudiantes tenían entre 19 y 25 años. Cuatro estudiantes tenían entre 26 y 45 años. Finalmente, tres estudiantes de más de 45 años asistieron al taller. El grupo estaba formado por seis mujeres/niñas y catorce hombres/niños. La Tabla 9 resume las calificaciones que tienen los estudiantes. Se incluye la correspondencia entre el nivel educativo español y el nivel ISCED.

Tabla 9. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	7
Educación Secundaria Obligatoria	2 y 3	7
Bachillerato	4	1
Formación Profesional	5	2
Ingeniería Industrial	6	1
Máster	7	2

Con la idea de que los alumnos, además de iniciarse a la robótica, pudieran trabajar de forma colaborativa los alumnos trabajaban en mesas con forma de U. En un lado de la U el material de aprendizaje visual fue mostrado usando una televisión. La Figura 29 incluye una imagen de la disposición de las mesas y sillas en la parte superior; y la ubicación del televisor.

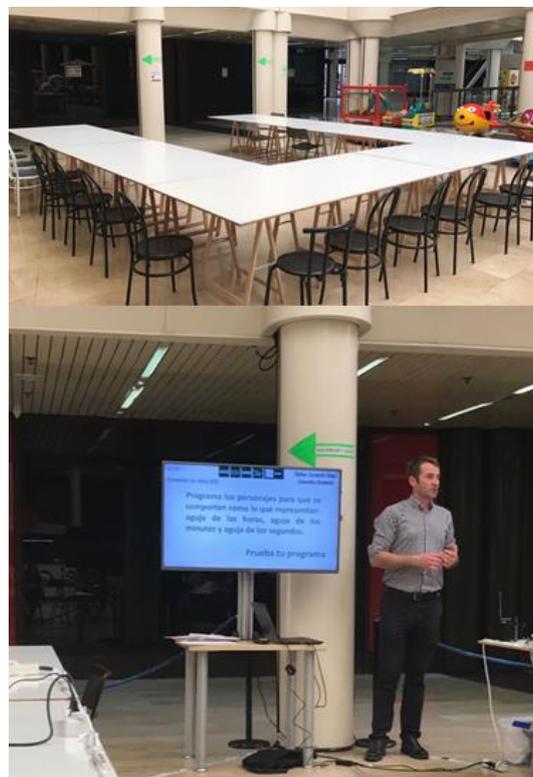


Figura 29. Configuración del aula del taller.

Los estudiantes tenían ordenadores portátiles con diferentes sistemas operativos. Algunos ordenadores ejecutan Windows 10, otros ejecutan Ubuntu 16.04 como sistema operativo. Un estudiante también usó MAC OS. Scratch es compatible con todos los sistemas operativos mencionados son compatibles con Scratch. Además, Scratch no requiere muchos recursos tales como velocidad del microprocesador o cantidad de memoria RAM.

La duración del taller fue de 4 horas y se dividió en tres partes:

1. La primera parte se utilizó para introducir la robótica, presentar los objetivos del taller y proporcionar recursos de Scratch. Como parte de la introducción del taller, el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) fue presentado. Tras la presentación del IEEE, se también se presentó la rama de estudiantes de la IEEE UNED
2. La segunda se centró en el conocimiento de la herramienta Scratch
3. La última parte se alineó con la construcción de aplicaciones mediante actividades de resolución de problemas. El taller constó de un total de 4 horas. Fue un enfoque pedagógico muy práctico. Las actividades fueron individuales y de colaboración.

Se observó cómo los estudiantes, durante el taller, mejoraron habilidades como el pensamiento sistémico, la mentalidad de programación, el aprendizaje activo, las matemáticas, las ciencias, el juicio y la toma de decisiones, la buena comunicación, el diseño tecnológico, la resolución de problemas complejos y la persistencia.

Toda la información sobre los detalles del contenido del taller y los resultados obtenidos están descritos en [Plaza, 2018b].

4.2.2. TALLER CRUMBLE DAY 2017

Este taller estuvo dirigido a niños y adultos que querían descubrir lo que la herramienta Crumble permite hacer en el campo de la robótica [Plaza, 2017a].

El grupo de estudiantes estaba formado por 5 estudiantes. Tres estudiantes tenían entre 10 y 15 años. Un estudiante tenía entre 26 y 45 años. Finalmente, un estudiante de más de 45 años asistió al taller. La Tabla 10 resume las calificaciones que tienen los estudiantes. Se incluye la correspondencia entre el nivel educativo español y el nivel ISCED.

Tabla 10. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Secundaria Obligatoria	2 y 3	3
Formación Profesional	5	1
Máster	7	1

La metodología utilizada fue una combinación de teoría y práctica. Al principio del taller, el equilibrio entre los dos era principalmente teórico y no práctico. Para tratar de mantener la atención de los asistentes, se les preguntó sobre los contenidos teóricos. De esta manera, los asistentes se involucraron más y su atención fue mayor. A partir de la mitad del taller, apenas se proporcionó contenido teórico y fueron los participantes los que tuvieron que poner en práctica todos los conocimientos que habían recibido.

La duración del taller fue de 4 horas y se dividió en tres partes:

1. La primera parte se utilizó para introducir la robótica, presentar los objetivos del taller y proporcionar información sobre los recursos de Crumble como el entorno de desarrollo, el controlador de Crumble y algunos elementos de hardware utilizados a lo largo del taller
2. La segunda parte se centró en adquirir conocimientos básicos sobre el Crumble. Esto se hizo a través de actividades sencillas destinadas a familiarizar a los asistentes con la herramienta y los componentes que iban a utilizar
3. Finalmente, la tercera parte se orientó a la construcción de aplicaciones basadas en la resolución de problemas. De esta manera, los participantes pudieron utilizar los conocimientos adquiridos en las fases para resolver los problemas planteados durante la última parte del taller. Todo el contenido del taller se impartió durante cuatro horas, aunque hubo una pausa en medio del taller

La Figura 30 muestra un robot basado en Crumble como los que se usaron durante el taller. Este robot consta de un controlador de desmoronamiento. También incluye un portapilas. También tiene dos motores y ruedas. En la parte inferior se puede ver un Sparkle, y debajo un sensor ultrasónico. Aunque no se muestra en la figura, la parte inferior del robot está equipada con un sensor de seguimiento de línea, que consiste en un par de sensores infrarrojos de encendido/apagado.



Figura 30. Ejemplo de kit robótico con Crumble.

A pesar de no tener un gran número de asistentes, la experiencia proporcionó resultados que pueden ser útiles para que otros profesores promuevan un taller con contenido similar o igual para obtener más resultados.

Toda la información sobre los detalles del contenido del taller y los resultados obtenidos están descritos en [Plaza, 2018h].

4.2.3. TALLER DE ROBÓTICA EDUCATIVA MULTI-PLATAFORMA 2017

El evento de la Semana Europea de Robótica (ERW) fue concebido en el Foro Europeo de Robótica (ERF) 2011 basándose en el deseo de la comunidad robótica europea de acercar la investigación y el desarrollo de la robótica al público y ayudar a construir una futura sociedad basada en la robótica. Cada año desde 2011, ERW ha ofrecido al público de toda Europa una semana de diversas actividades relacionadas con la robótica para destacar la creciente importancia de la robótica en una amplia variedad de áreas de aplicación y la creciente importancia de las habilidades en ciencia, tecnología, ingeniería y matemáticas (STEM).

El taller se realizó en el centro de La Estera [La Estera, 2018]. Este taller fue apoyado, tanto a nivel logístico, como didáctico por:

- Incomaz (Tienda de Informática y Sonido) que dio apoyo publicitario y logístico a este taller [Incomaz, 2018].
- Plaza Robótica [Plaza Robotica, 2018]
- La Rama de Estudiantes de la UNED IEEE
- La Escuela Industrial de Ingeniería de la UNED,
- El Departamento de Ingeniería Eléctrica e Informática (DIEEC)
- Y la Escuela de Doctorado de la UNED

Este taller utiliza un enfoque altamente práctico utilizando una metodología pedagógica basada en proyectos. Se realizan experimentos en los que el ensayo y el error son un componente del aprendizaje y el autodescubrimiento. Los asistentes aprenden a tener más autonomía y responsabilidad. Los conocimientos se adquieren en diversas disciplinas para desarrollar las siguientes:

- Destreza motriz (mayor movilidad en las manos).
- Habilidad de grupo, permitiendo a la gente socializar.
- Habilidades creativas.
- Aprender de una manera divertida.

La duración del taller fue de 4 horas y se dividió en cuatro partes:

1. La primera sección se utilizó para introducir la robótica y presentar los objetivos y la agenda del taller
2. La segunda sección se centró en la obtención de conocimientos de Scratch y en el trabajo sobre los contenidos multiplataforma asociados
3. La tercera sección se centró en la obtención de conocimientos sobre Crumble y el trabajo sobre los contenidos multiplataforma asociados
4. La sección final o cuarta fue diseñada para desarrollar el conocimiento de Arduino y trabajar sobre el contenido multiplataforma asociado. El taller tuvo una duración total de 4 horas y utilizó un enfoque pedagógico práctico. Las actividades fueron individuales y de colaboración.

El grupo de asistentes estaba formado por ocho personas. Las edades de los asistentes estaban comprendidas entre 18 y 65 años. El grupo estaba formado por una mujer y siete hombres. La Tabla 11 resume las calificaciones de los asistentes. Se incluye la correspondencia entre el grado educativo español y el nivel ISCED.

Tabla 11. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Secundaria Obligatoria	2 y 3	1
Formación Profesional	5	4
Ingeniería / Diplomatura	6	3

No se necesitan grandes inversiones para replicar este taller. Además, el contenido basado en la luz puede modificarse añadiendo motores, servos y sensores para hacer que el contenido sea más complejo. Por ejemplo, en lugar de abordar el tema del semáforo, el instructor puede proponer una actividad que implemente una barrera de acceso al

garaje como siguiente paso. Para ampliar la complejidad, el instructor podría proponer una actividad basada en un robot de seguimiento de línea que evita obstáculos.

Toda la información sobre los detalles del contenido del taller y los resultados obtenidos están descritos en [Plaza, 2018c].

4.2.4. TALLERES DE ROBÓTICA EDUCATIVA MULTI-PLATAFORMA 2018

El contenido del taller de robótica educativa multi-plataforma 2017 se usó para desplegar una serie de talleres en FlexiEspacio [FlexiEspacio, 2018] con el objetivo de obtener más resultados y acumularlos a los ya obtenidos. El contenido educativo usado es el mismo descrito en la sección 8.2.3. Por ello, se han agrupado los resultados obtenidos en todos los talleres realizados.

El grupo de asistentes estaba formado por diez personas. Uno de los asistentes era menor de 8 años, cuatro de los asistentes tenían entre 10 y 12 años, dos de los asistentes tenían entre 26 años y 36 años, y el resto de los asistentes tenían entre 37 y 55 años. El grupo estaba formado por una mujer, cinco hombres y cinco niños. La Tabla 12 resume las calificaciones de los asistentes. Se incluye la correspondencia entre el grado educativo español y el nivel ISCED.

Tabla 12. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	1
Educación Secundaria Obligatoria	2 y 3	4
Formación Profesional	5	1
Ingeniería / Diplomatura	6	5

Antes de que comenzara cada taller, se realizó una prueba previa para recoger las preocupaciones relacionadas con la asistencia al taller. También se preguntó a los asistentes sobre sus conocimientos previos de programación, robótica y uso de las herramientas Scratch, Crumble y Arduino.

La Tabla 13 recopila las respuestas a otras preguntas relacionadas con los conocimientos previos de programación, robótica y uso de las herramientas Scratch, Crumble y Arduino. La mayoría de los asistentes no tenían experiencia en programación ni conocimientos de programación. Además, la mayoría de los asistentes no tenían conocimientos relacionados con la robótica, Scratch, Crumble o Arduino. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 13. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	1	1	4	3	1
Conocimientos de programación	1	3	3	2	1
Conocimientos de robótica	2	6	2	0	0
Experiencia con Scratch	4	2	1	1	2
Conocimiento de Scratch	4	2	2	2	0
Experiencia con Crumble	7	1	1	1	0
Conocimiento de Crumble	7	1	1	1	0
Experiencia con Arduino	6	2	1	1	0
Conocimiento de Arduino	6	2	1	1	0

La Tabla 14 incluye información sobre las actividades realizadas durante la segunda sección. Además, la tabla incluye cuántos asistentes completaron cada una de estas actividades sin ayuda o con algo de ayuda.

Tabla 14. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda
Paneles de Scratch	10 (100 %)	0 (0 %)
Agrupación de comandos	10 (100 %)	0 (0 %)
Panel de programación	10 (100 %)	0 (0 %)
Panel de visualización	10 (100 %)	0 (0 %)
Creando un semáforo	8 (80 %)	2 (20 %)
Editando el fondo	5 (50 %)	5 (50 %)
Creando los personajes	6 (60 %)	4 (40 %)
Programando los personajes	9 (90 %)	1 (10 %)
Sincronizando el semáforo	10 (100 %)	0 (0 %)

La Tabla 15 incluye información sobre las actividades de la tercera sección. Esta tabla también incluye cuántos asistentes completaron cada una de estas actividades sin ayuda o con algo de ayuda para completar la actividad correspondiente.

Tabla 15. Resultados de las actividades de las sesiones de la tercera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda
Controlador Crumble	10 (100 %)	0 (0 %)
Software de Crumble	10 (100 %)	0 (0 %)
Sistemas de alimentación	10 (100 %)	0 (0 %)
Motores eléctricos	10 (100 %)	0 (0 %)
Luces LED	10 (100 %)	0 (0 %)
Conexiones eléctricas de Crumble	2 (20 %)	8 (80 %)
Primer programa con Crumble	1 (10 %)	9 (90 %)

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda
Usando Sparkles	9 (90 %)	1 (10 %)
Programando el semáforo	10 (100 %)	0 (0 %)

La Tabla 16 se presenta información sobre las actividades realizadas durante la cuarta sección.

Tabla 16. Resultados de las actividades de las sesiones de la cuarta parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda
Controlador Arduino	10 (100 %)	0 (0 %)
Software de Arduino	10 (100 %)	0 (0 %)
Sistemas de alimentación	10 (100 %)	0 (0 %)
Motores eléctricos	10 (100 %)	0 (0 %)
Luces LED	10 (100 %)	0 (0 %)
Conexiones eléctricas de Arduino	2 (20 %)	8 (80 %)
Primer programa con Arduino	6 (60 %)	4 (40 %)
Uso de los LED	10 (100 %)	0 (0 %)
Programando el semáforo	10 (100 %)	0 (0 %)

Además, al final del taller, se llevó a cabo una prueba posterior para obtener información sobre los resultados de los asistentes al taller.

En primer lugar, se hizo una batería de preguntas con el objetivo de recoger las opiniones de los asistentes sobre el taller. La Tabla 17 enumera los comentarios recopilados.

Tabla 17. Opinión de los asistentes acerca del taller.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	8	0	1	2
Hacer montajes con Crumble	2	4	3	4
Hacer montajes con Arduino	0	0	4	4
La parte teórica	0	0	2	0
Nada	0	6	0	0

Adicionalmente, la Tabla 18 enumera las respuestas a las otras preguntas relacionadas con el conocimiento adquirido en programación, robótica y el uso de las herramientas Scratch, Crumble y Arduino.

Tabla 18. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	2	7	1
Conocimientos de robótica	0	0	0	7	3
Interés por la robótica	0	0	0	4	5
Conocimiento de Scratch	0	0	0	9	1
Conocimiento de Crumble	0	0	0	10	0
Conocimiento de Arduino	0	0	0	6	4

Se obtuvieron unos resultados que confirman la validez del uso de las herramientas Scratch, Crumble y Arduino como adecuadas para introducir a los estuantes en la robótica de una manera divertida, económica y de forma sencilla.

4.2.5. CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2017/2018

Este curso es una evolución del curso de introducción a la robótica 2016/2017. En este caso, se han incluido en cada módulo varios niveles de dificultad. De esta manera, los estudiantes pueden experimentar con cada una de las herramientas a un mayor nivel de profundidad. Los estudiantes también disponen de más tiempo para poder asimilar mejor los conceptos de mayor complejidad.

Este curso se basa en tres herramientas ampliamente utilizadas en la enseñanza. Las herramientas se caracterizan por dos parámetros principales: el grado de simplicidad de su uso y el grado de complejidad de las aplicaciones que se pueden realizar con ellas.

Por otro lado, el curso consta de tres niveles de dificultad:

- Un primer nivel es muy sencillo y tiene como objetivo principal ser un contacto de los alumnos con la herramienta en cuestión
- A esto le sigue un nivel intermedio, este nivel es un puente entre una toma de contacto con la herramienta y el desarrollo de aplicaciones de cierto grado de complejidad
- Y, finalmente, un nivel de mayor complejidad con el que se pretende que los estudiantes puedan desarrollar aplicaciones complejas en el campo de la herramienta utilizada y en el marco de la robótica educativa.

Por tanto, el curso fue diseñado de tal forma que cada módulo se dividió a su vez en tres módulos de dificultad. Cada módulo está diseñado para ser desplegado en sesiones semanales. En la Figura 31 se muestra el conjunto de módulos de la matriz colocados por tipo de herramienta utilizada y nivel de dificultad.

		NIVELES DE DIFICULTAD		
		BÁSICO	INTERMEDIO	AVANZADO
HERRAMIENTAS DE ROBÓTICA EDUCATIVA		1.1. Scratch Básico	1.2. Scratch Intermedio	1.2. Scratch Avanzado
		2.1. Crumble Básico	2.2. Crumble Intermedio	2.2. Crumble Avanzado
		3.1. Arduino Básico	3.2. Arduino Intermedio	3.2. Arduino Avanzado

Figura 31. Matriz de contenidos del curso.

A continuación, se muestra cómo se puede desarrollar cada módulo en función de la herramienta usada y de la dificultad:

1. Scratch nivel básico: En la primera parte de este módulo, se muestra a los estudiantes la estructura de la interfaz de desarrollo y sus diferentes elementos. Se explican los paneles que componen la interfaz de desarrollo, la categorización de comandos, algunos comandos básicos y el panel de visualización. En este punto, pasamos a actividades en las que los alumnos tienen más participación. Primero, trabajas en la programación sin un ordenador. Los estudiantes deben definir comandos inventados, definir una lista de ellos y, finalmente, ejecutar ese conjunto de instrucciones. Entonces deben empezar a programar con Scratch. El primer reto es programar al gato Scratch para que se mueva de izquierda a derecha. El siguiente ejercicio tiene como objetivo conseguir que el personaje pinte el fondo a medida que avanza. Usando el programa generado hasta ahora, los estudiantes deben modificar el programa para que el personaje dibuje un triángulo y un cuadrado. Para terminar esta parte, los estudiantes deben definir una palabra, compartirla con otros y crear una historia con todas las palabras. Con esa historia definida, los estudiantes deben hacer un programa con Scratch. En la segunda parte trabajamos en la implementación de un semáforo con Scratch. El primer paso es crear una señal de tráfico con personajes. Para hacer esto, los estudiantes usarán

los personajes como semáforos. Luego crearán un cruce con dos señales de tráfico, y luego crearán un cruce con cuatro semáforos. En este punto, los estudiantes son desafiados a crear un circuito con dos señales de tráfico. Posteriormente, los estudiantes deben usar personajes para crear las luces para la parte del cruce peatonal. Finalmente, el objetivo final es implementar un semáforo para controlar el cruce de vehículos y peatones. En la tercera parte, se trabaja en la implementación de una barrera de acceso al garaje. Para hacer esto, los estudiantes deben usar un personaje para implementar la barrera. También deben incluir un sensor de color en la programación para que la barrera sepa cuando un vehículo está esperando, pasando o no hay ningún vehículo presente. Además, la barrera del garaje debe incluir un semáforo, para informarle cuando puede pasar, cuando se requiere precaución porque la barrera se está abriendo o cerrando y, finalmente, para informarle cuando no puede pasar porque la barrera está cerrada. En la última parte, los estudiantes deben usar un personaje que contenga detectores de color para que se pueda detectar una línea negra sobre un fondo blanco. También deben modificar el fondo para crear un circuito circular. Una vez desarrollados estos dos elementos y el programa personaje, los estudiantes deben modificar sus programas para asegurarse de que su robot virtual sigue las líneas lo más rápidamente posible sin salir del circuito. El siguiente paso es utilizar un circuito de forma rectangular. Finalmente, el personaje se utilizará para circular en circuitos más complejos que los propios alumnos deberán crear.

2. Scratch nivel intermedio: Al comienzo de este módulo, los estudiantes ya tienen cierta familiaridad con la herramienta Scratch. Saben cuáles son las partes importantes y conocen una variedad de comandos y su uso. En la primera parte de este módulo, trabajamos en la implementación de un coche policía con Scratch. Para ello, los estudiantes deben crear su personaje de coche policía. No tienen restricciones de formas ni de colores. Una vez hayan trabajado con el editor de imágenes y tengan su propio coche policía que incluya las luces que caracterizan un coche patrulla, los estudiantes comenzarán a trabajar con los comandos de movimiento y aspecto. También harán uso de los comandos de interacción de Scratch con el teclado del ordenador. De esta manera, el coche policía mostrará las luces de persecución y se moverá cuando un usuario lo indique pulsando las correspondientes teclas del teclado. En la segunda parte trabajamos en la creación de un coche que será aquel que tiene que ser apresado por el coche de policía. Esta fase del módulo es muy similar al desarrollo anterior, pero el coche no tendrá

las luces de persecución, será un coche normal. Este personaje también será controlado por las teclas del teclado. De esta manera, dos usuarios podrán interactuar con Scratch, y cada uno de los usuarios será capaz de manejar, o el coche policial, o el coche perseguido. En la tercera parte, pasamos al uso de los comandos de sensores que tiene Scratch. El coche perseguido debe detectar si ha sido alcanzado por el coche patrulla. De ser así, debe cambiar su aspecto para que aparezca enjaulado. Además, si el usuario que controla el coche perseguido quiere manejar este vehículo y ha sido apresado, el coche perseguido no se podrá mover, ya que está inmovilizado. En la última parte, los estudiantes deben incluir un circuito a través del cual se moverán, tanto el coche policía, como el coche perseguido. También deben incluir sensores en los dos personajes, ya que, si alguno de los vehículos alcanza los límites exteriores del circuito, ambos vehículos volverán a la posición inicial. Asimismo, los estudiantes crearán un contador para cada uno de los personajes, este contador representará el número de vidas que tiene cada vehículo. Si el coche policía apresa al coche perseguido, el primero ganará una vida y el segundo la perderá. Si cualquiera de los dos se sale del circuito, perderá una vida. Cuando alguno de los vehículos tenga un total de cero vidas, el juego habrá concluido y se deberá mostrar una pantalla de fin de juego indicando quién ha sido el jugador ganador.

3. Scratch nivel avanzado: Cuando los alumnos comienzan este módulo, ya tienen un manejo bastante elevado de Scratch. En este punto es interesante darles más grados de libertad a la hora de hacer desarrollos. De esta manera, pueden desarrollar la creatividad y la autonomía. En la primera parte de este módulo, se les explica a los estudiantes cuál es el objetivo del módulo. A lo largo de este módulo desarrollarán un videojuego de laberintos con varios niveles. El primer paso será crear un primer fondo, un personaje que debe escapar del laberinto y la casilla de inicio y fin del laberinto. El trazado del primer laberinto no debe ser muy complejo, de hecho, conviene que sea tan sencillo como una línea recta. El aspecto del personaje debe ser definido completamente por el alumno. Los alumnos deben programar al personaje para que se mueva en función de las teclas que pulse un usuario. En la segunda parte trabajamos en la creación de la pantalla de fin, donde se le dará la enhorabuena al usuario que haya sido capaz de llegar desde la posición de inicio hasta la posición de fin. Si el personaje tocara los bordes del trazado, el juego se reiniciará. Si el personaje alcanza la posición de fin, el aspecto del juego deberá mostrar una pantalla de fin de juego indicando que se ha logrado salir del laberinto. En la tercera parte, pasamos a añadir más niveles. Cada nivel

debe tener un trazado más complejo que el nivel anterior. Es importante que las posiciones de inicio y fin de cada laberinto esté en el mismo sitio. De esta manera, los usuarios que utilicen este juego irán avanzando en los niveles que cree el alumno hasta que alcancen la pantalla de fin de juego. En la última parte, los estudiantes deben incluir un contador de vidas del personaje. Este contador se irá reduciendo a medida que el personaje toque los bordes del trazado. Así, en lugar de empezar de cero cada vez que se toque un borde, el personaje empezará de nuevo sólo el laberinto donde tocó el borde. Además, el videojuego deberá almacenar el número de veces que se ha reiniciado cada laberinto y el número de veces que se ha reiniciado el juego. Entonces, cuando un usuario finalice las pruebas, en la pantalla final, se mostrarán: las vidas del personaje, los intentos en cada laberinto y el número de veces que se ha reiniciado el videojuego.

4. Crumble nivel básico: La primera parte de este módulo muestra a los estudiantes el hardware de Crumble. Se describe la tarjeta controladora y los elementos de hardware que se utilizarán a lo largo del módulo. Estos elementos de hardware son, por ejemplo, el soporte de la batería, los cables de conexión eléctrica, el cable USB para la programación de la tarjeta controladora, el Sparkle, el sensor ultrasónico, los motores y el sensor de detección de línea. A continuación, se describe la estructura de la interfaz de desarrollo y sus diferentes elementos. Se explican los paneles que componen la interfaz de desarrollo, la categorización de comandos y algunos comandos básicos. En este punto, pasamos a actividades en las que los alumnos tienen más participación. El primer reto es que programen el controlador para que el Sparkle tenga el color de su elección. El siguiente ejercicio está diseñado para hacer que el destello parpadee con un período de encendido de un segundo y un período de apagado de un segundo. Para terminar esta parte, los estudiantes deben programar el controlador para que Sparkle muestre una sucesión de colores de su elección. En la segunda parte, trabajamos en la implementación de un semáforo con Scratch. El primer paso es crear la secuencia de aspectos que serán mostrados por el Sparkle. Los estudiantes usarán la siguiente secuencia: dos segundos en rojo, dos segundos en amarillo y dos segundos en verde. Luego modificarán su programa para que la secuencia se ejecute cíclicamente. En el siguiente paso, los estudiantes deben modificar su horario para el aspecto amarillo, en lugar de ser un aspecto fijo, o un aspecto intermitente. En este punto, los estudiantes se enfrentan al reto de crear las luces en el lado del cruce de peatones, es decir, rojo fijo, verde fijo y verde intermitente. Finalmente, el objetivo final es implementar un semáforo para controlar el cruce de vehículos

y peatones. En la tercera parte, se trabaja en la implantación de una barrera de acceso al garaje. Para hacer esto, los estudiantes deben usar un servo para implementar la barrera. También deben incluir un sensor ultrasónico para que la barrera sepa cuando un vehículo está esperando, pasando o no hay ningún vehículo presente. Además, la barrera del garaje debe incluir un semáforo, para informarle cuando puede pasar, cuando se requiere precaución porque la barrera se está abriendo o cerrando y, finalmente, para informarle cuando no puede pasar porque la barrera está cerrada. En la última parte, los estudiantes deben usar un robot con ruedas que contenga detectores de línea para que se pueda detectar una línea negra sobre un fondo blanco. Se utilizará una superficie blanca y una cinta aislante negra para dibujar los diferentes circuitos. Una vez que se desarrolla un programa que hace que el robot siga el circuito circular, los estudiantes deben modificar sus programas para asegurarse de que su robot sigue las líneas lo más rápidamente posible sin salir del circuito. El siguiente paso es utilizar un circuito de forma rectangular. Finalmente, utilizará circuitos más complejos que los propios estudiantes deben crear.

5. Crumble nivel intermedio: Al comienzo de este módulo, los estudiantes ya tienen cierta familiaridad con la herramienta Crumble. Saben cómo es el entorno de desarrollo, conocen una variedad de comandos y cómo usarlos con diferentes actuadores y sensores. En la primera parte de este módulo, trabajamos en la implementación de un coche policía con Crumble. Para ello, los estudiantes deben crear su propio coche policía. No tienen restricciones de formas ni de colores. Pero deben montar un robot de ruedas basado en Crumble con motores y con Sparkles. También deben realizar una programación básica para comprobar que se mueve correctamente y que las luces funcionan como se espera. En la segunda parte trabajamos en la programación del robot para que se pueda mover a lo largo de diferentes circuitos. Un primer circuito será muy sencillo, un circuito circular. El segundo circuito será cuadrado. Un tercer circuito será rectangular. Por último, se pedirá a los alumnos que sus robots policías se muevan a lo largo de un circuito con un trazado más complejo que los mencionados anteriormente. En la tercera parte, pasamos al uso del sensor de ultrasonidos. En este caso, los robots deben detectar objetos a cierta distancia. Si el objeto está a una distancia mayor de veinte centímetros, el robot deberá circular por un circuito (se puede usar cualquiera de los mencionados en la parte anterior). En el caso que el robot detecte un objeto a menos de diez centímetros, el robot se deberá detener hasta que deje de detectarlo. En la última parte, los estudiantes retirarán el sensor de

ultrasonidos para usar un sensor de infrarrojos y unos sensores de detección de línea. La mecánica de funcionamiento es similar a la planteada en la etapa anterior. La diferencia es que ya no hará falta programar el trazado para seguir un circuito u otro. Los estudiantes dibujarán el circuito con cinta aislante negra sobre un fondo blanco. El robot debe ser programado para que siga la línea negra. Las condiciones de arranque y parada en función de los obstáculos detectados por el sensor de distancia serán las mismas que las planteadas en la parte anterior.

6. Crumble nivel avanzado: Un alumno que inicia este módulo ya tiene una destreza bastante elevada de Crumble. Ahora los estudiantes deben potenciar su creatividad y autonomía en el desarrollo de aplicaciones avanzadas. En la primera parte de este módulo, se les explica a los estudiantes cuál es el objetivo del módulo. A lo largo de este módulo implementarán una grúa controlada por Crumble. El primer paso será crear un robot con ruedas y con un brazo manipulador basado en servos. El brazo de la grúa tendrá un grado de libertad, subir el brazo o bajarlo. También tendrá una pinza para poder agarrar objetos. Además de montar la construcción, los alumnos también deben realizar una programación básica para comprobar que se mueve todo correctamente. En la segunda parte los alumnos añadirán un sensor para detectar obstáculos. Cuando la grúa detecte un objeto, debe cogerlo con la pinza, retirarlo a un lado y continuar por donde circulaba. Después se probará el robot grúa para que se pueda mover a lo largo de diferentes circuitos. Un primer circuito será muy sencillo, un circuito circular. El segundo circuito será cuadrado. Un tercer circuito será rectangular. Por último, se pedirá a los alumnos que sus robots policías se muevan a lo largo de un circuito con un trazado más complejo que los mencionados anteriormente. En la tercera parte, los alumnos modificarán la programación para que el robot grúa recoja piezas que forman una columna, y el propio robot, monte con esas piezas una columna en otro lugar que se le indica. En la última parte, los estudiantes participarán en una competición para ver quién ha desarrollado la grúa que es capaz de recolocar las columnas. En este caso habrá equipos de robots grúa. De manera que un robot le coloca una columna situada en una posición de inicio en otra posición donde se encuentra esperando otro robot. Este último recolocará la columna en la zona de meta.
7. Arduino nivel básico: En la primera parte de este módulo, se muestra a los estudiantes el hardware de Arduino. Se describe la tarjeta controladora y los elementos de hardware que se utilizarán a lo largo del módulo. Estos elementos de hard-

ware son, por ejemplo, los cables de conexión eléctrica, el cable USB para la programación de la tarjeta controladora, los LED (diodos emisores de luz), el sensor ultrasónico, los motores y el sensor de detección de línea. A continuación, se describe la estructura de la interfaz de desarrollo y sus diferentes elementos. Explica la estructura de los programas de Arduino, la sintaxis general del lenguaje, cómo están configurados los puertos, y cómo se escriben y se leen los puertos de la tarjeta. También se incluyen explicaciones sobre cómo elegir el puerto de programación COM y qué tarjeta elegir antes de programar. Finalmente, detalla cómo se cargará un programa en la tarjeta Arduino. En este punto, pasamos a actividades en las que los alumnos tienen más participación. El primer reto es programar el controlador para que el LED integrado en la placa parezca fijo. El siguiente ejercicio tiene por objeto hacer que el LED parpadee durante un segundo de encendido y un segundo de apagado. Para terminar esta parte, los estudiantes deben integrar un LED externo en la placa y utilizar el programa creado previamente. En la segunda parte, trabajamos en la implementación de un semáforo con Arduino. El primer paso es conectar un LED para cada aspecto de la señal de tráfico. Segundo, los estudiantes deben programar la secuencia de aspectos que se mostrarán en los LED. Los estudiantes usarán la siguiente secuencia: dos segundos en rojo, dos segundos en amarillo y dos segundos en verde. En el siguiente paso, los estudiantes deben modificar su horario para el aspecto amarillo, en lugar de ser un aspecto fijo, o un aspecto intermitente. En este punto, los estudiantes se enfrentan al reto de crear las luces en el lado del cruce de peatones, es decir, rojo fijo, verde fijo y verde intermitente. Finalmente, el objetivo final es implementar un semáforo para controlar el cruce de vehículos y peatones. En la tercera parte, se trabaja en la implantación de una barrera de acceso al garaje. Para hacer esto, los estudiantes deben usar un servo para implementar la barrera. También deben incluir un sensor ultrasónico para que la barrera sepa cuando un vehículo está esperando, pasando o no hay ningún vehículo presente. Además, la barrera del garaje debe incluir un semáforo, para informarle cuando puede pasar, cuando se requiere precaución porque la barrera se está abriendo o cerrando y, finalmente, para informarle cuando no puede pasar porque la barrera está cerrada. En la última parte, los estudiantes deben usar un robot con ruedas que contenga detectores de línea para que se pueda detectar una línea negra sobre un fondo blanco. Se utilizará una superficie blanca y una cinta aislante negra para dibujar los diferentes circuitos. Una vez que se desarrolla un programa que hace que el robot

siga el circuito circular, los estudiantes deben modificar sus programas para asegurarse de que su robot sigue las líneas lo más rápidamente posible sin salir del circuito. El siguiente paso es utilizar un circuito de forma rectangular. Finalmente, utilizará circuitos más complejos que los propios estudiantes deben crear.

8. **Arduino nivel intermedio:** Al comienzo de este módulo, los estudiantes ya tienen cierta familiaridad con la herramienta Arduino. Saben cómo es el entorno de desarrollo, conocen una variedad de comandos y cómo usarlos con diferentes actuadores y sensores. En la primera parte de este módulo, trabajamos en la implementación de un coche policía con Arduino. Para ello, los estudiantes deben crear su propio coche policía. No tienen restricciones de formas ni de colores. Pero deben montar un robot de ruedas basado en Arduino con motores y con LED. También deben realizar una programación básica para comprobar que se mueve correctamente y que las luces funcionan como se espera. En la segunda parte trabajamos en la programación del robot para que se pueda mover a lo largo de diferentes circuitos. Un primer circuito será muy sencillo, un circuito circular. El segundo circuito será cuadrado. Un tercer circuito será rectangular. Por último, se pedirá a los alumnos que sus robots policías se muevan a lo largo de un circuito con un trazado más complejo que los mencionados anteriormente. En la tercera parte, pasamos al uso del sensor de ultrasonidos. En este caso, los robots deben detectar objetos a cierta distancia. Si el objeto está a una distancia mayor de veinte centímetros, el robot deberá circular por un circuito (se puede usar cualquiera de los mencionados en la parte anterior). En el caso que el robot detecte un objeto a menos de diez centímetros, el robot se deberá detener hasta que deje de detectarlo. En la última parte, los estudiantes incluirán unos sensores de detección de línea. La mecánica de funcionamiento es similar a la planteada en la etapa anterior. La diferencia es que ya no hará falta programar el trazado para seguir un circuito u otro. Los estudiantes dibujarán el circuito con cinta aislante negra sobre un fondo blanco. El robot debe ser programado para que siga la línea negra. Las condiciones de arranque y parada en función de los obstáculos detectados por el sensor de distancia serán los mismos que los planteados en la parte anterior.
9. **Arduino nivel avanzado:** Cuando un alumno inicia este módulo y ha pasado por los ocho módulos anteriores, ya tiene una gran destreza con la programación, uso de componentes electrónicos, montajes mecánicos e implementación de aplicaciones robóticas de cierta complejidad. En la primera parte de este módulo, se le explica a los estudiantes cuál es el objetivo del módulo. A lo largo de este módulo

implementarán un bípedo basado en servos manejado por Bluetooth. El primer paso será montar las piernas del robot y conectar la controladora Arduino. También deben realizar una programación básica para comprobar que se mueve correctamente dando pasos sencillos. En la segunda parte los alumnos añadirán un módulo Bluetooth para poder controlar el robot desde un dispositivo móvil (Smartphone, Tablet, portátil) o un ordenador de sobremesa. Cuando el dispositivo conectado por Bluetooth le mande un comando al módulo Bluetooth del robot, el robot debe dar un paso hacia adelante con cada una de sus piernas. En la tercera parte, los alumnos ampliarán la programación para que el robot obedezca comandos de avance, retroceso y giro hacia ambos lados. En la última parte, los estudiantes añadirán un sensor de distancia. Este sensor de distancia se usará para detectar objetos. Si el robot bípedo basado en servos manejado por Bluetooth detecta un obstáculo a menos de 10 centímetros, dejará de obedecer los comandos y se agachará. Pasados unos segundos, se volverá a levantar. Si vuelve a detectar un obstáculo repetirá el procedimiento. En caso de no detectar obstáculos, volverá a obedecer los comandos recibidos por Bluetooth.

Métricas e indicadores del curso

Para evaluar si un estudiante ha adquirido las habilidades y destrezas necesarias, se han propuesto diferentes métricas e indicadores que deben utilizarse para saber si el estudiante está listo para pasar al siguiente módulo. Se han propuesto dos tipos de herramientas de medición: en primer lugar, el análisis del instructor a lo largo de las sesiones y, en segundo lugar, el uso de encuestas de control al principio y al final de cada módulo.

La edad de los estudiantes debe ser considerada al hacer la transición de un módulo a otro. Esto es importante, especialmente debido a la edad mínima recomendada para cada una de las herramientas.

También es importante conocer las preocupaciones de los estudiantes. Un estudiante puede completar el módulo intermedio de Scratch y la transición natural sería al módulo avanzado de Scratch. Pero si el estudiante está interesado en comenzar el módulo Desmenuzar, sería aconsejable tener en cuenta esta opinión a la hora de elegir el siguiente módulo a tomar por el estudiante.

Progreso de los estudiantes

Dada la estructura del curso, es posible hacer muchas transiciones entre módulos. La Figura 32 muestra diferentes transiciones posibles. Primero, puedes hacer un tour desde el Scratch básico avanzado de Arduino hasta todos los niveles de dificultad y todas las

herramientas. Esta ruta está representada por la Figura 32 como flechas azules marcadas con el identificador 1. También es posible, si la edad del estudiante es igual o mayor que la edad recomendada para cada una de las herramientas, que el estudiante asista primero a los niveles básicos de cada herramienta, luego a los niveles intermedios y termine los niveles avanzados de cada una de las herramientas. Esta trayectoria se representa en la Figura 32 como flechas verdes marcadas con el identificador 2.

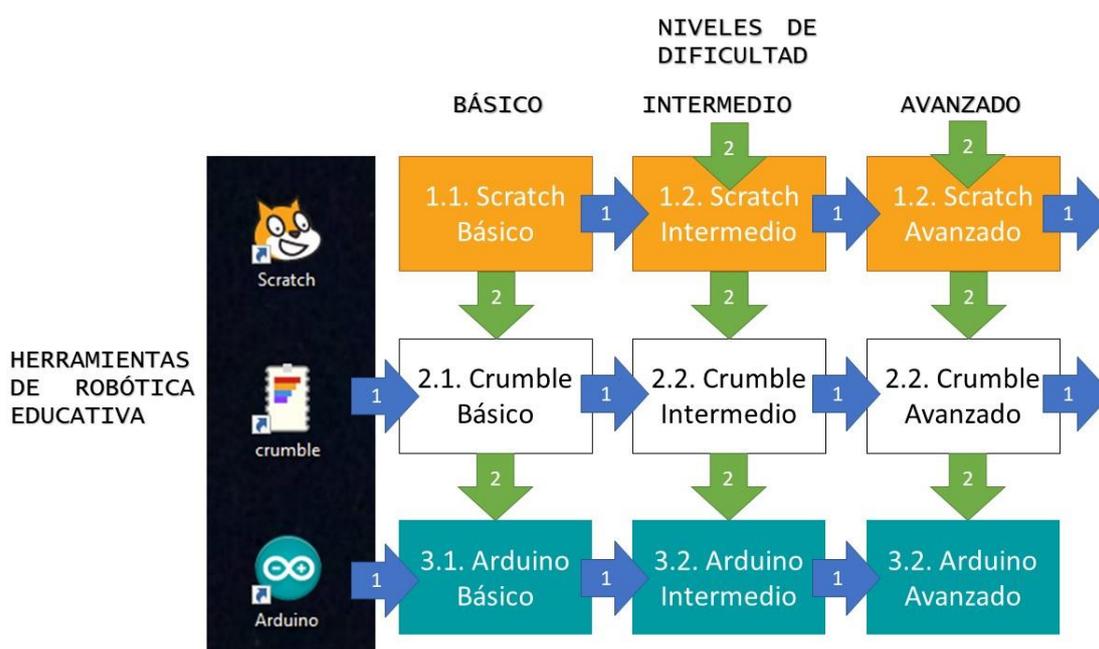


Figura 32. Transiciones para cada módulo.

Además, no es un requisito previo para comenzar en el nivel básico de Scratch. Si un estudiante tiene experiencia previa en programación y robótica o tiene la edad y el conocimiento de que herramientas como Scratch y Crumble son demasiado triviales para él o ella, este estudiante podría empezar con el módulo básico de Arduino.

Resultados obtenidos en La Estera para Scratch básico

El grupo de estudiantes estaba formado por catorce estudiantes de entre 6 y 11 años. Cuatro estudiantes eran menores de 8 años y el resto tenía entre 9 y 11 años.

Uno de los estudiantes es una niña (10 años) y los otros trece son niños. La Tabla 19 resume las calificaciones de los estudiantes. Se incluye la correspondencia entre el nivel educativo español y el nivel ISCED.

Tabla 19. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	14

Antes de comenzar el módulo, se realizó una prueba previa para obtener las inquietudes acerca de su asistencia al módulo. También se les preguntó acerca de sus conocimientos previos sobre programación, robótica y el uso de la herramienta Scratch.

La Tabla 20 recoge las respuestas a las otras preguntas relacionadas con los conocimientos previos sobre programación, robótica y uso de la herramienta Scratch. La mayoría de los estudiantes no tenían experiencia en programación, robótica y Scratch. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 20. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	14	0	0	0	0
Conocimientos de programación	14	0	0	0	0
Conocimientos de robótica	14	0	0	0	0
Experiencia con Scratch	12	1	1	0	0
Conocimiento de Scratch	12	1	1	0	0

En la primera parte, la mayoría de los estudiantes tuvieron su primer contacto con Scratch. Esta parte es eminentemente teórica. Para evitar la falta de implicación de los alumnos durante las explicaciones teóricas, se hizo a los estudiantes parte activa mediante el planteamiento de preguntas. El resto de las actividades requirieron ciertas aclaraciones adicionales y, en algunos casos, fue necesario ayudar a los estudiantes para que pudiesen finalizar la correspondiente actividad con ayuda. La Tabla 21 resume la información relativa a las actividades realizadas por los alumnos.

Tabla 21. Resultados de las actividades de las sesiones de la primera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Entorno de Scratch	14 (100 %)	0 (0 %)	0 (0 %)
Programación sin ordenador	0 (0 %)	2 (14 %)	12 (86 %)
Programación básica	14 (100 %)	0 (0 %)	0 (0 %)
Programando una historia sencilla	4 (29 %)	10 (71 %)	0 (0 %)

La Tabla 22 se resume la información relativa a las actividades realizadas por los alumnos a lo largo de la segunda parte. Durante las primeras actividades, los alumnos necesitaban ayuda para alcanzar los objetivos que se habían fijado. Sin embargo, los estudiantes completaron las dos últimas actividades sin tener que ser ayudados.

Tabla 22. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Trabajando con personajes	0 (0 %)	2 (14 %)	12 (86 %)
Luces de semáforo básicas (I)	0 (0 %)	4 (29 %)	10 (71 %)
Luces de semáforo básicas (II)	14 (100 %)	0 (0 %)	0 (0 %)
Luces de semáforo complejas	14 (100 %)	0 (0 %)	0 (0 %)

En la tercera parte, de nuevo los estudiantes necesitaron ayuda durante las primeras actividades. Al igual que en la segunda parte del módulo, en esta tercera parte los alumnos realizaron las últimas actividades de forma independiente. La Tabla 23 resume la información relativa a las actividades que fueron completadas por los estudiantes.

Tabla 23. Resultados de las actividades de las sesiones de la tercera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Barrera de garaje sencilla	2 (14 %)	12 (86 %)	0 (0 %)
Barrera de garaje con detección	4 (29 %)	10 (71 %)	0 (0 %)
Barrera de garaje con luces	14 (100 %)	0 (0 %)	0 (0 %)
Barrera de garaje compleja	14 (100 %)	0 (0 %)	0 (0 %)

La Tabla 24 resume la información relativa a las actividades realizadas por los alumnos durante la cuarta parte. Como se desprende de los resultados obtenidos, todos los alumnos mostraron una mayor independencia en la resolución de los retos a los que se habían enfrentado.

Tabla 24. Resultados de las actividades de las sesiones de la cuarta parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Detector de color	10 (71 %)	2 (14 %)	2 (14 %)
Circuito	12 (86 %)	2 (14 %)	0 (0 %)
Incrementando la velocidad	12 (86 %)	2 (14 %)	0 (0 %)
Circuito complejo	14 (100 %)	0 (0 %)	0 (0 %)

Además, al final del módulo, se llevó a cabo un cuestionario para obtener la opinión del estudiante sobre los resultados del módulo. En primer lugar, se hizo una batería de preguntas con el objetivo de obtener la opinión de los estudiantes sobre el módulo. La Tabla 25 recoge la opinión de los estudiantes sobre el módulo.

Tabla 25. Opinión de los asistentes acerca del curso.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	3	2	10	3
Crear historias	0	12	3	6
Hacer aplicaciones de la vida real	11	0	1	5
La parte teórica	0	0	0	0
Nada	0	0	0	0

Todas las preguntas relacionadas con la verificación de que los estudiantes habían adquirido los conocimientos que se pretendía transmitir a lo largo del curso fueron contestadas correctamente por todos los estudiantes.

Por otro lado, la Tabla 26 recoge las respuestas a las otras preguntas relacionadas con los conocimientos adquiridos sobre programación, robótica y uso de la herramienta Scratch. La mayoría de los estudiantes aumentaron su percepción sobre programación, robótica y conocimientos de Scratch. Además, la mayoría de los estudiantes mostraron un aumento en su curiosidad por la robótica. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 26. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	2	7	5
Conocimientos de robótica	0	0	3	3	8
Interés por la robótica	0	1	3	2	8
Conocimiento de Scratch	0	0	2	7	5

Al final del módulo pudimos ver cómo todos los estudiantes aumentaron su motivación para programar y crear proyectos sencillos. Como se ha demostrado a lo largo de estos resultados, la combinación de la pedagogía STEM con la robótica educativa presenta oportunidades prometedoras para el desarrollo de las habilidades y competencias necesarias para los futuros profesionales.

Resultados obtenidos en La Estera para Scratch intermedio

Al comienzo de este módulo se unieron dos estudiantes más. Por lo tanto, el grupo estaba formado por 16 estudiantes. Como se ha hecho en las diferentes experiencias educativas mostradas anteriormente, antes de comenzar el módulo, se realizó una prueba previa

para obtener las inquietudes acerca de su asistencia al módulo. Los dos alumnos que se incorporaron al curso comenzaron el curso no tenían ningún conocimiento ni experiencia previa en programación, robótica o Scratch.

La Tabla 27 recoge las respuestas a las otras preguntas relacionadas con los conocimientos previos sobre programación, robótica y uso de la herramienta Scratch. La mayoría de los estudiantes no tenían experiencia en programación, robótica y Scratch. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 27. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	2	0	14	0	0
Conocimientos de programación	2	0	14	0	0
Conocimientos de robótica	2	0	14	0	0
Experiencia con Scratch	2	0	12	1	1
Conocimiento de Scratch	2	0	12	1	1

En la primera parte, la mayoría de los estudiantes estaban ya familiarizados con Scratch. Esta parte tiene una componente más práctica que el módulo de Scratch básico. La Tabla 28 resume la información relativa a las actividades realizadas por los alumnos.

Tabla 28. Resultados de las actividades de las sesiones de la primera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Personaje coche policía	2 (12,5 %)	12 (75 %)	2 (12,5 %)
Programación básica	3 (18,75 %)	9 (56,25 %)	4 (25 %)
Interacción con usuario	13 (81,25 %)	2 (12,5 %)	1 (6,25 %)
Luces y movimiento	15 (93,75 %)	0 (0 %)	1 (6,25 %)

La Tabla 29 se resume la información relativa a las actividades realizadas por los alumnos a lo largo de la segunda parte.

Tabla 29. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Personaje coche perseguido	1 (6,25 %)	13 (81,25 %)	2 (12,5 %)
Programación básica	0 (0 %)	12 (75 %)	4 (25 %)
Interacción con usuario	15 (93,75 %)	0 (0 %)	1 (6,25 %)
Movimiento	15 (93,75 %)	0 (0 %)	1 (6,25 %)

La Tabla 30 resume la información relativa a las actividades que fueron completadas por los estudiantes.

Tabla 30. Resultados de las actividades de las sesiones de la tercera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Sensores de personajes	0 (0 %)	12 (75 %)	4 (25 %)
Detección de personajes	4 (25 %)	10 (62,5 %)	2 (12,5 %)
Interacción entre personajes	15 (93,75 %)	1 (6,25 %)	0 (0 %)
Interacción personajes-usuarios	15 (93,75 %)	1 (6,25 %)	0 (0 %)

La Tabla 31 resume la información relativa a las actividades realizadas por los alumnos durante la cuarta parte. Como se desprende de los resultados obtenidos, todos los alumnos mostraron una mayor independencia en la resolución de los retos a los que se habían enfrentado.

Tabla 31. Resultados de las actividades de las sesiones de la cuarta parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Circuito básico	10 (62,5 %)	4 (25 %)	2 (12,5 %)
Interacción personajes-fondo	14 (87,5 %)	2 (12,5 %)	0 (0 %)
Uso de variables	14 (87,5 %)	2 (12,5 %)	0 (0 %)
Programación compleja	16 (100 %)	0 (0 %)	0 (0 %)

Además, al final del módulo, se llevó a cabo un test de fin para obtener la opinión del estudiante sobre los resultados del módulo. La Tabla 32 recoge la opinión de los estudiantes sobre el módulo.

Tabla 32. Opinión de los asistentes acerca del curso.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	4	3	1	6
Hacer aplicaciones de la vida real	2	3	2	1
Trabajar con videojuegos	12	0	13	1
La parte teórica	0	4	0	0
Nada	0	6	0	8

Todas las preguntas relacionadas con la verificación de que los estudiantes habían adquirido los conocimientos que se pretendía transmitir a lo largo del curso fueron contestadas correctamente por todos los estudiantes.

Por otro lado, la Tabla 33 recoge las respuestas a las otras preguntas relacionadas con los conocimientos adquiridos sobre programación, robótica y uso de la herramienta Scratch. La mayoría de los estudiantes aumentaron su percepción sobre programación, robótica y conocimientos de Scratch. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 33. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	1	13	2
Conocimientos de robótica	0	0	2	10	4
Interés por la robótica	0	0	1	11	4
Conocimiento de Scratch	0	0	0	9	6

Al final del módulo pudimos ver cómo todos los estudiantes aumentaron su motivación para programar y crear proyectos de una dificultad mayor que las actividades que desarrollaron durante el módulo de Scratch básico.

Resultados obtenidos en La Estera para Scratch avanzado

El grupo de estudiantes estaba formado por cinco estudiantes de entre 7 y 10 años. Todos los estudiantes son niños. La Tabla 34 resume las calificaciones de los estudiantes. Se incluye la correspondencia entre el nivel educativo español y el nivel ISCED.

Tabla 34. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	5

Antes de comenzar el módulo, se realizó una prueba previa para obtener las inquietudes acerca de su asistencia al módulo. Todos los estudiantes habían cursado ya el módulo de Scratch intermedio.

La Tabla 35 recoge las respuestas a las otras preguntas relacionadas con los conocimientos previos sobre programación, robótica y uso de la herramienta Scratch. La mayoría de los estudiantes tenían experiencia en programación, robótica y Scratch. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 35. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	0	0	1	4	0
Conocimientos de programación	0	0	2	3	0
Conocimientos de robótica	0	0	1	4	0
Experiencia con Scratch	0	0	0	5	0
Conocimiento de Scratch	0	0	1	4	0

En la primera parte, la mayoría de los estudiantes estaban ya bastante familiarizados con Scratch. Este nivel de dificultad es eminentemente práctico. Este nivel de dificultad tam-

bién promueve la creatividad de los estudiantes a través de la resolución de retos mediante el aprendizaje basado en proyectos. La Tabla 36 resume la información relativa a las actividades realizadas por los alumnos.

Tabla 36. Resultados de las actividades de las sesiones de la primera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Laberinto básico	2 (40 %)	2 (40 %)	1 (20 %)
El protagonista del escape	3 (60 %)	1 (20 %)	1 (20 %)
Interacción con usuario	3 (60 %)	1 (20 %)	1 (20 %)
Probando el nivel básico	5 (100 %)	0 (0 %)	0 (0 %)

La Tabla 37 resume la información relativa a las actividades realizadas por los alumnos a lo largo de la segunda parte.

Tabla 37. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Pantalla de fin	1 (20 %)	2 (40 %)	2 (40 %)
Interacción personajes-fondo (I)	2 (40 %)	1 (20 %)	2 (40 %)
Interacción personajes-fondo (II)	5 (100 %)	0 (0 %)	0 (0 %)
Probando el nivel intermedio	5 (100 %)	0 (0 %)	0 (0 %)

La Tabla 38 resume la información relativa a las actividades que fueron completadas por los estudiantes.

Tabla 38. Resultados de las actividades de las sesiones de la tercera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Laberintos complejos	3 (60 %)	1 (20 %)	1 (20 %)
Interacción personajes-fondo	3 (60 %)	1 (20 %)	1 (20 %)
Interacción usuarios-fondo (I)	4 (80 %)	1 (20 %)	0 (0 %)
Interacción usuarios-fondo (II)	4 (80 %)	1 (20 %)	0 (0 %)

La Tabla 39 resume la información relativa a las actividades realizadas por los alumnos durante la cuarta parte.

Tabla 39. Resultados de las actividades de las sesiones de la cuarta parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Vidas del personaje	5 (100 %)	0 (0 %)	0 (0 %)
Interacción personajes-fondo	4 (80 %)	1 (20 %)	0 (0 %)
Interacción usuarios-fondo	5 (100 %)	0 (0 %)	0 (0 %)
Interacción usuarios-videojuego	5 (100 %)	0 (0 %)	0 (0 %)

Además, al final del módulo, se llevó a cabo un cuestionario para obtener la opinión del estudiante sobre los resultados del módulo. La Tabla 40 recoge la opinión de los estudiantes sobre el módulo.

Tabla 40. Opinión de los asistentes acerca del curso.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	0	2	2	3
Hacer aplicaciones de la vida real	0	0	0	0
Trabajar con videojuegos	5	0	3	0
La parte teórica	0	0	0	0
Nada	0	3	0	2

Por otro lado, la Tabla 41 recoge las respuestas a las otras preguntas relacionadas con los conocimientos adquiridos sobre programación, robótica y uso de la herramienta Scratch. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 41. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	1	1	3
Conocimientos de robótica	0	0	2	3	0
Interés por la robótica	0	0	0	0	5
Conocimiento de Scratch	0	0	0	1	4

Al final del módulo pudimos ver cómo todos los estudiantes aumentaron su motivación para programar y crear proyectos de una dificultad todavía mayor que las actividades que desarrollaron durante el módulo de Scratch intermedio.

Resultados obtenidos en La Estera para Crumble básico

El grupo de estudiantes estaba formado por catorce estudiantes de entre 9 y 12 años. Dos de los estudiantes son niñas (9 y 10 años) y los otros doce son niños. La Tabla 42 resume las calificaciones de los estudiantes.

Tabla 42. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	14

Antes de comenzar el módulo, se realizó una prueba previa para obtener las inquietudes acerca de su asistencia al módulo. También se les preguntó acerca de sus conocimientos previos sobre programación, robótica y el uso de la herramienta Crumble.

La Tabla 43 recoge las respuestas a las otras preguntas relacionadas con los conocimientos previos sobre programación, robótica y uso de la herramienta Crumble. Ningún estudiante había tenido experiencias previas con Crumble, ni tenían conocimientos sobre esta herramienta. Sin embargo, la mayoría de los estudiantes tenían experiencia en programación y robótica. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 43. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	1	2	0	11	0
Conocimientos de programación	1	2	0	11	0
Conocimientos de robótica	2	1	0	11	0
Experiencia con Crumble	14	0	0	0	0
Conocimiento de Crumble	14	0	0	0	0

En la primera parte, todos los estudiantes tuvieron su primer contacto con Crumble. Por otro lado, un importante número de estudiantes ya habían cursado los módulos de Scratch básico y Scratch intermedio. La metodología usada fue igual a la empleada en el módulo de Scratch básico, pero con los contenidos adaptados a la herramienta Crumble. La Tabla 44 resume la información relativa a las actividades realizadas por los alumnos.

Tabla 44. Resultados de las actividades de las sesiones de la primera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Entorno de Crumble	14 (100 %)	0 (0 %)	0 (0 %)
Hardware de Crumble	2 (14 %)	12 (86 %)	0 (0 %)
Programación básica	0 (0 %)	2 (14 %)	12 (86 %)
Programación intermedia	10 (72 %)	4 (28 %)	0 (0 %)

La Tabla 45 se resume la información relativa a las actividades realizadas por los alumnos a lo largo de la segunda parte.

Tabla 45. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Trabajando con Sparkles	2 (14 %)	12 (86 %)	0 (0 %)
Luces de semáforo básicas (I)	11 (79 %)	2 (14 %)	1 (7 %)
Luces de semáforo básicas (II)	14 (100 %)	0 (0 %)	0 (0 %)
Luces de semáforo complejas	14 (100 %)	0 (0 %)	0 (0 %)

La Tabla 46 resume la información relativa a las actividades que fueron completadas por los estudiantes.

Tabla 46. Resultados de las actividades de las sesiones de la tercera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Barrera de garaje sencilla	1 (7 %)	12 (86 %)	1 (7 %)
Barrera de garaje con detección	3 (21 %)	10 (72 %)	1 (7 %)
Barrera de garaje con luces	14 (100 %)	0 (0 %)	0 (0 %)
Barrera de garaje compleja	14 (100 %)	0 (0 %)	0 (0 %)

La Tabla 47 resume la información relativa a las actividades realizadas por los alumnos durante la cuarta parte.

Tabla 47. Resultados de las actividades de las sesiones de la cuarta parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Detector de color	10 (71 %)	3 (21 %)	1 (7 %)
Circuito	10 (71 %)	4 (28 %)	0 (0 %)
Incrementando la velocidad	13 (93 %)	1 (7 %)	0 (0 %)
Circuito complejo	14 (100 %)	0 (0 %)	0 (0 %)

Además, al final del módulo, se llevó a cabo un cuestionario para obtener la opinión del estudiante sobre los resultados del módulo. La Tabla 48 recoge la opinión de los estudiantes sobre el módulo.

Tabla 48. Opinión de los asistentes acerca del curso.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	1	0	1	1
Encender las luces	12	0	13	0
Mover los motores	1	2	0	4
Usar los sensores	0	5	0	9
Nada	0	7	0	0

Por otro lado, la Tabla 49 recoge las respuestas a las otras preguntas relacionadas con los conocimientos adquiridos sobre programación, robótica y uso de la herramienta Crumble. La mayoría de los estudiantes aumentaron su percepción sobre programación, robótica y conocimientos de Crumble. Además, la mayoría de los estudiantes mostraron un aumento en su curiosidad por la robótica. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 49. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	2	7	5
Conocimientos de robótica	0	0	3	3	8
Interés por la robótica	0	1	3	2	8
Conocimiento de Crumble	0	0	2	7	5

Al final del módulo pudimos ver cómo todos los estudiantes aumentaron su motivación para programar, crear proyectos sencillos y trabajar con elementos físicos de robótica educativa.

4.2.6. TALLER DE ROBÓTICA EDUCATIVA PARA LA INTEGRACIÓN Y LA INCLUSIÓN EDUCON 2018

Este taller se realizó dentro del congreso EDUCON y tuvo una duración de 90 minutos. La agenda se dividió en cuatro bloques principales:

1. Bloque destinado a la robótica se presenta como una herramienta para promover la inclusión e integración de los estudiantes
2. Bloque destinado a presentar la pedagogía STEM y la educación robótica como el trasfondo subyacente de este taller
3. Bloque destinado a presentar las herramientas Raspberry Pi y Arduino, y el contenido se despliega basado en ambas herramientas:
 - Preparar una Raspberry Pi: Pixel OS.
 - Configurar el servidor LAMP (Linux, Apache, MySQL y PHP).
 - Configurar WordPress.
 - Manejar Arduino a través de la web.
4. Bloque destinado a discutir sobre el contenido y las experiencias.

En el último bloque del taller, la discusión se centra en cómo la robótica educativa puede ser buena para motivar a los jóvenes estudiantes en las disciplinas básicas, como las matemáticas y la física. Los estudiantes piensan que estas disciplinas son demasiado complicadas y abstractas, pero con las aplicaciones basadas en robótica, los estudiantes pueden ver lo útiles que pueden ser estas disciplinas en la vida.

Toda la información sobre los detalles del contenido del taller y los resultados obtenidos están descritos en [Peixoto, 2018b].

4.3. Actividades realizadas en 2018/2019

En el curso académico 2018/2019 se usaron los conocimientos adquiridos en las actividades realizadas del curso 2017/2018 para seguir probando los contenidos basados en los del curso anterior. Se trabajó en un curso anual tal y como se ha definido en la sección 8.2.5. A lo largo de esta sección se detallan los resultados obtenidos.

4.3.1. RESULTADOS OBTENIDOS EN ACADEMIA PIXEL PARA SCRATCH BÁSICO

El curso se realizó en el centro Academia PIXEL [PIXEL, 2018], es un centro de estudios que está ubicado en la localidad de Rivas Vaciamadrid, zona sureste de la Comunidad de Madrid.

El grupo de estudiantes estaba formado por siete estudiantes de entre 7 y 13 años. Cuatro estudiantes eran menores de 8 años y el resto tenía entre 9 y 11 años. Todos los estudiantes son niños. La Tabla 50 resume las calificaciones de los estudiantes. Además, incluye la correspondencia entre el nivel educativo español y el nivel ISCED.

Tabla 50. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	6
Educación Secundaria Obligatoria	2 y 3	1

Antes de comenzar el módulo, se realizó una prueba previa para obtener las inquietudes acerca de su asistencia al módulo. También se les preguntó acerca de sus conocimientos previos sobre programación, robótica y el uso de la herramienta Scratch. La Tabla 51 recoge las respuestas de los estudiantes.

Tabla 51. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	3	0	1	2	1
Conocimientos de programación	4	1	0	1	1
Conocimientos de robótica	3	1	1	1	2
Experiencia con Scratch	4	0	1	1	1
Conocimiento de Scratch	4	0	1	1	1

En la primera parte, la mayoría de los estudiantes tuvieron su primer contacto con Scratch. La Tabla 52 resume la información relativa a las actividades realizadas por los alumnos.

Tabla 52. Resultados de las actividades de las sesiones de la primera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Entorno de Scratch	7 (100 %)	0 (0 %)	0 (0 %)
Programación sin ordenador	0 (0 %)	0 (0 %)	7 (100 %)
Programación básica	2 (29 %)	1 (14 %)	4 (57 %)
Programando una historia sencilla	2 (29 %)	2 (29 %)	3 (42 %)

La Tabla 53 se resume la información relativa a las actividades realizadas por los alumnos a lo largo de la segunda parte.

Tabla 53. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Trabajando con personajes	2 (29 %)	1 (14 %)	4 (57 %)
Luces de semáforo básicas (I)	2 (29 %)	1 (14 %)	4 (57 %)
Luces de semáforo básicas (II)	4 (57 %)	2 (29 %)	1 (14 %)
Luces de semáforo complejas	7 (100 %)	0 (0 %)	0 (0 %)

En la tercera parte los alumnos realizaron las últimas actividades de forma independiente. La Tabla 54 resume la información relativa a las actividades que fueron completadas por los estudiantes.

Tabla 54. Resultados de las actividades de las sesiones de la tercera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Barrera de garaje sencilla	4 (57 %)	2 (29 %)	1 (14 %)
Barrera de garaje con detección	4 (57 %)	2 (29 %)	1 (14 %)
Barrera de garaje con luces	7 (100 %)	0 (0 %)	0 (0 %)
Barrera de garaje compleja	7 (100 %)	0 (0 %)	0 (0 %)

La Tabla 55 resume la información relativa a las actividades realizadas por los alumnos durante la cuarta parte.

Tabla 55. Resultados de las actividades de las sesiones de la cuarta parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Detector de color	5 (56 %)	0 (0 %)	1 (14 %)
Circuito	5 (56 %)	0 (0 %)	1 (14 %)
Incrementando la velocidad	5 (56 %)	0 (0 %)	1 (14 %)
Circuito complejo	6 (86 %)	1 (14 %)	0 (0 %)

Además, al final del módulo, se llevó a cabo un cuestionario para obtener la opinión del estudiante sobre los resultados del módulo. La Tabla 56 recoge la opinión de los estudiantes sobre el módulo.

Tabla 56. Opinión de los asistentes acerca del curso.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	3	0	6	0
Crear historias	0	0	0	0
Hacer aplicaciones de la vida real	4	0	1	3
La parte teórica	0	0	0	0
Nada	0	7	0	1

Por otro lado, la Tabla 57 recoge las respuestas a las otras preguntas relacionadas con los conocimientos adquiridos sobre programación, robótica y uso de la herramienta Scratch.

Tabla 57. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	3	3	1
Conocimientos de robótica	0	0	2	3	2
Interés por la robótica	0	1	3	0	3
Conocimiento de Scratch	0	0	3	0	4

Se obtuvieron unos resultados que confirman la validez del uso de la herramienta Scratch como adecuada para introducir a los estuantes en la robótica de una manera divertida, económica y de forma sencilla.

4.3.2. RESULTADOS OBTENIDOS EN ACADEMIA PIXEL PARA CRUMBLE BÁSICO

El módulo se desarrolló en la Academia PIXEL [PIXEL, 2018] y el grupo de estudiantes es el mismo que el descrito en la sección 4.3.1.

Antes de comenzar el módulo, se realizó una prueba previa para obtener las inquietudes acerca de su asistencia al módulo. También se les preguntó acerca de sus conocimientos previos sobre programación, robótica y el uso de la herramienta Crumble.

La Tabla 58 recoge las respuestas a las otras preguntas relacionadas con los conocimientos previos sobre programación, robótica y uso de la herramienta Crumble. Ningún estudiante había tenido experiencias previas con Crumble, ni tenían conocimientos sobre esta herramienta. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 58. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	0	0	1	6	0
Conocimientos de programación	0	0	2	5	0
Conocimientos de robótica	0	0	2	5	0
Experiencia con Crumble	7	0	0	0	0
Conocimiento de Crumble	7	0	0	0	0

En la primera parte, todos los estudiantes tuvieron su primer contacto con Crumble. Por otro lado, un importante número de estudiantes ya habían cursado los módulos de Scratch básico y Scratch intermedio. La metodología usada fue igual a la empleada en el módulo de Scratch básico, pero con los contenidos adaptados a la herramienta Crumble. La Tabla 59 resume la información relativa a las actividades realizadas por los alumnos.

Tabla 59. Resultados de las actividades de las sesiones de la primera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Entorno de Crumble	0 (0 %)	7 (100 %)	0 (0 %)
Hardware de Crumble	0 (0 %)	0 (0 %)	7 (100 %)
Programación básica	2 (29 %)	4 (57 %)	1 (14 %)
Programación intermedia	2 (29 %)	3 (42 %)	2 (29 %)

La Tabla 60 se resume la información relativa a las actividades realizadas por los alumnos a lo largo de la segunda parte.

Tabla 60. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Trabajando con Sparkles	3 (42 %)	4 (57 %)	0 (0 %)
Luces de semáforo básicas (I)	3 (42 %)	3 (42 %)	1 (14 %)
Luces de semáforo básicas (II)	7 (100 %)	0 (0 %)	0 (0 %)
Luces de semáforo complejas	7 (100 %)	0 (0 %)	0 (0 %)

La Tabla 61 resume la información relativa a las actividades que fueron completadas por los estudiantes.

Tabla 61. Resultados de las actividades de las sesiones de la tercera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Barrera de garaje sencilla	2 (28 %)	4 (57 %)	1 (14 %)
Barrera de garaje con detección	3 (42 %)	3 (42 %)	1 (14 %)
Barrera de garaje con luces	7 (100 %)	0 (0 %)	0 (0 %)
Barrera de garaje compleja	7 (100 %)	0 (0 %)	0 (0 %)

La Tabla 62 resume la información relativa a las actividades realizadas por los alumnos durante la cuarta parte.

Tabla 62. Resultados de las actividades de las sesiones de la cuarta parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Detector de color	4 (58 %)	2 (28 %)	1 (14 %)
Circuito	5 (72 %)	2 (28 %)	0 (0 %)
Incrementando la velocidad	6 (86 %)	1 (14 %)	0 (0 %)
Circuito complejo	7 (100 %)	0 (0 %)	0 (0 %)

Además, al final del módulo, se llevó a cabo un cuestionario para obtener la opinión del estudiante sobre los resultados del módulo. La Tabla 63 recoge la opinión de los estudiantes sobre el módulo.

Tabla 63. Opinión de los asistentes acerca del curso.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	1	0	1	1
Encender las luces	5	0	6	0
Mover los motores	1	1	0	1
Usar los sensores	0	2	0	5
Nada	0	4	0	0

Por otro lado, la Tabla 64 recoge las respuestas a las otras preguntas relacionadas con los conocimientos adquiridos sobre programación, robótica y uso de la herramienta Crumble. La mayoría de los estudiantes aumentaron su percepción sobre programación, robótica y conocimientos de Crumble. Además, la mayoría de los estudiantes mostraron un aumento en su curiosidad por la robótica. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 64. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	1	4	2
Conocimientos de robótica	0	0	1	3	3
Interés por la robótica	0	1	1	2	3
Conocimiento de Crumble	0	0	1	4	2

Al final del módulo pudimos ver cómo todos los estudiantes aumentaron su motivación para programar, crear proyectos sencillos y trabajar con elementos físicos de robótica educativa.

CAPÍTULO 5

5. ARQUITECTURA DE LA PLATAFORMA

Como se ha podido comprobar, existe una gran variedad de plataformas de desarrollo para aplicaciones IoT (Internet of Things). A pesar de esto, todavía no se ha podido encontrar una propuesta que integre una red de dispositivos para desplegar contenidos educativos de robótica para proyectos enmarcados en STEM (Science, Technology, Engineering and Mathematics). Por ello, se ha definido el desarrollo de una plataforma que permite elaborar programas educativos en un entorno de desarrollo colaborativo para potenciar la innovación y la motivación de los alumnos durante el proceso de aprendizaje de áreas de conocimiento STEM combinadas con robótica educativa. Una herramienta educativa modular, reconfigurable, flexible, versátil y con un coste que puede ser fácilmente adaptado a las necesidades y al poder adquisitivo de la entidad interesada en incluir esta herramienta en sus programas educativos.

El objetivo de esta plataforma es aportar a los usuarios una experiencia de aprendizaje que va más allá de la adquisición de conocimientos o de habilidades concretas, puesto que ofrece oportunidades reales gracias a la libertad de reconfiguración, flexibilidad, versatilidad y escalabilidad. Además, posibilita la elaboración de prácticas metodológicas activas de aprendizaje centradas principalmente en el usuario y en la potenciación de las relaciones de grupo. Los usuarios pueden configurarla de manera que se creen espacios de aprendizaje propios o espacios de aprendizaje grupales. También posibilita a los educadores la opción de crear actividades creativas, divergentes y abiertas.

Dada la complejidad de la arquitectura de la plataforma, se incluye desarrollo hardware, firmware y software, que se ha separado en cuatro capítulos. Este capítulo recoge las especificaciones generales de la arquitectura de la plataforma y la metodología de alto nivel que se lleva a cabo a lo largo del documento. La plataforma se divide en tres grandes áreas: el hardware, el firmware y el software. Para describir cada una de estas áreas se elabora un capítulo independiente con la descripción del desarrollo de la plataforma enmarcándola dentro de cada una de ellas.

5.1. Herramienta Colaborativa Robótica Educativa

La propuesta consta de cinco PCB (Printed Circuit Board). La arquitectura del sistema, desde el punto de vista del hardware, se compone de un módulo principal y de cuatro módulos secundarios. Por un lado, el módulo principal es el PCB al que se conectan los

cuatro módulos secundarios. El módulo principal, además de comunicarse con los módulos secundarios, tiene la capacidad de comunicarse por Bluetooth con Smart Devices separados físicamente de la plataforma. Los Smart Devices son dispositivos tales como un Smart Phone, Tablets, SBC (System Board Computer), o incluso, ordenadores personales. Asimismo, al módulo principal se pueden ensamblar Shields de Arduino. Por otro lado, los módulos secundarios son compatibles con Shields de Arduino, con lo que se pueden ensamblar estos últimos a los módulos secundarios con facilidad. Los módulos secundarios tienen compatibilidad con Elementos Externos. Los Elementos Externos son placas que harán de interfaz entre la plataforma y diferentes tipos de circuitos electrónicos o plataformas robóticas para que puedan ser integradas al sistema. Por último, y gracias a la comunicación Bluetooth, la plataforma puede tener acceso, o ser accedida a través de Internet. Este tipo de conectividad se consigue a través del módulo principal usando Smart Devices para tal efecto. Los Smart Devices, en este caso, actúan como puentes entre el usuario y la plataforma. La Figura 33 muestra los distintos elementos que componen la plataforma y los diferentes dispositivos que se pueden añadir al sistema.

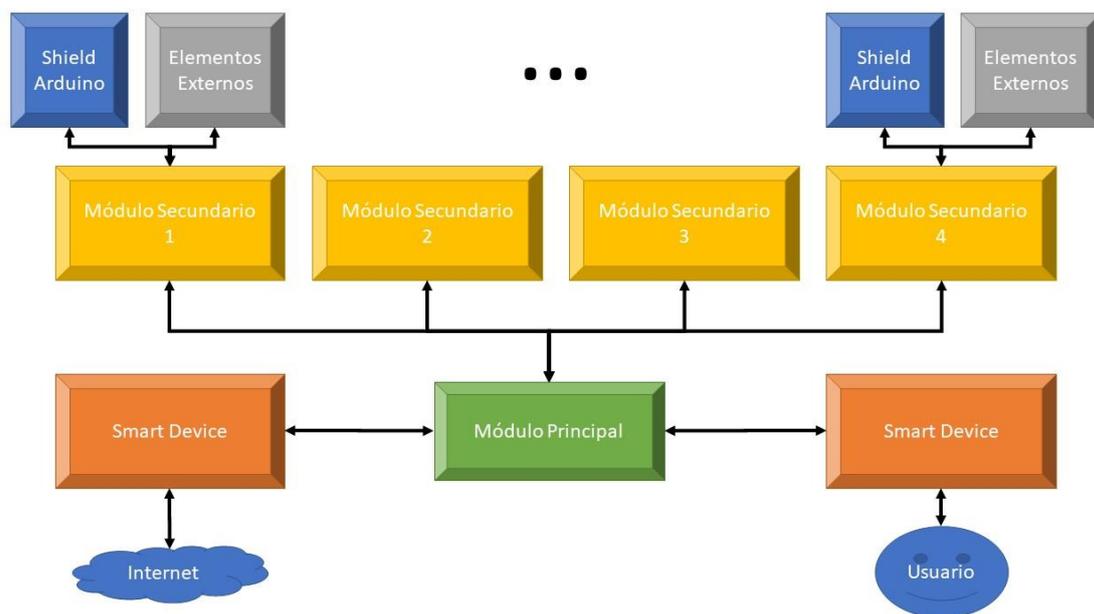


Figura 33. Diagrama de bloques de la plataforma y sus periféricos.

La Figura 34 muestra la Herramienta Colaborativa Robótica Educativa basada en Lógica Programable y Arduino. Como se aprecia en la Figura 34 incluye un módulo principal y un módulo secundario. La Herramienta Colaborativa Robótica Educativa puede estar compuesta por un máximo de cuatro módulos secundarios. Cada uno de estos módulos está formado por otros componentes que son definidos en los siguientes subapartados

de esta sección. En el ejemplo representado por la Figura 34 se pueden ver conectados al módulo principal: un módulo de Bluetooth, una Arduino Nano, un Shield de Bluetooth compatible con Arduino y un módulo secundario. También se puede observar una Arduino Nano y un Shield de Interconexiones compatible con Arduino, conectados ambos un módulo secundario.

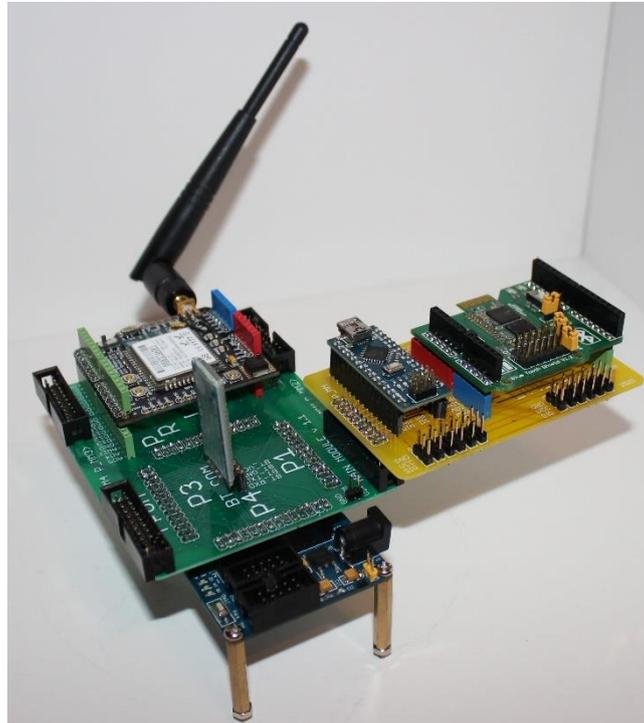


Figura 34. Herramienta Robótica Educativa.

5.1.1. FUNCIONALIDADES

La plataforma creada pretende cubrir una serie de funcionalidades fundamentales como:

- Reconfiguración. Gracias a la arquitectura y estructura de conexiones arriba expuestas, la plataforma descrita es capaz de ser reconfigurada de tal manera que pueda ser fácilmente adaptada para poder configurar uno u otro sistema, en función de las necesidades del uso que se le vaya a dar. La reconfiguración se puede realizar previamente a su uso o durante el mismo.
- Complejidad y coste. También es posible modificar el número de elementos que forman la plataforma. Así, el grado de complejidad de la plataforma puede ser reducido o incrementado, haciendo posible elaborar un sistema escalable en función de las necesidades o de la capacidad económica con la que se cuente.
- Conexión con otros dispositivos. Otra funcionalidad que aporta la plataforma es la posibilidad de conexión con otros dispositivos como puede ser un PC o Smart

Devices. En la actualidad el acceso a un Smartphone, Tablet o PC de tamaño reducido es muy común, por lo que se ha tenido en cuenta, que pueda incluirse como parte del sistema, siendo la conexión sin cables. Esta característica da la posibilidad de trabajar con un sistema IoT como herramienta educativa [Brown, 2002].

- Comunicación concurrente. Al estar basada la plataforma en FPGA, las comunicaciones internas de la misma se realizarán de manera concurrente, para así reducir el riesgo de tener un cuello de botella en este punto. De esta manera se evitan retardos indeseados y se obtiene un control casi total de los tiempos de ejecución y del flujo de los datos manejados.
- Funciones de protección. Además, se incluyen funcionalidades de protección del sistema, siendo monitorizadas la tensión y la corriente en diferentes puntos. También se cuenta con comprobaciones de estado, tanto en el arranque de la plataforma, como durante su uso.
- Redes. Permite configurar una red de Arduino para un control de robots colaborativo dentro de un grupo de alumnos.
- Protecciones menores. Adicionalmente, la plataforma puede ser configurada para añadir protecciones menores a periféricos que se le puedan conectar, para evitar que manipulaciones malintencionadas puedan provocar un deterioro de la plataforma o de los propios periféricos conectados a ellos.
- Prototipado. Se permite el uso de prototipos que puedan ser conectados como un módulo más. Con este grado de flexibilidad se potencia la innovación del usuario a la hora de trabajar con la plataforma, ya sea por parte del profesorado, o bien por parte de los alumnos que estén desarrollando la actividad educativa.

5.2. Metodologías de desarrollo

La arquitectura de la plataforma se enmarca en tres grandes áreas: el hardware, el firmware y el software. Por lo tanto, se debe definir una metodología que sea lo suficientemente flexible y eficaz para cada una de ellas.

En [Metodología, 2018a] se define metodología como la “ciencia del método y como un conjunto de métodos que se siguen en una investigación científica, o en una exposición doctrinal”. Por lo tanto, se podría definir como aquella guía que hace referencia al camino o al conjunto de procedimientos racionales utilizados para alcanzar el objetivo o la variedad de objetivos que rige una investigación científica [Metodología, 2018b]. También se entiende por metodología el enfoque que permite observar un problema de una

forma total, sistemática y disciplinada [Metodología, 2018c]. Como se muestra, es posible encontrar varias definiciones de metodología. Aun así, todas estas definiciones conducen a la misma idea

Existen multitud de metodologías para el análisis y diseño de sistemas, por esta razón, se ha hecho una investigación sobre las diferentes metodologías existentes, y las ventajas y desventajas que presentan, con el fin de elegir la más adecuada.

5.2.1. METODOLOGÍA DEL DESARROLLO RÁPIDO DE APLICACIONES

Esta metodología de desarrollo de software conocida como Metodología RAD (Rapid Application Development), fue creada por el gurú de computación James Martin en 1991. Está orientada a disminuir radicalmente el tiempo necesario para diseñar e implementar Sistemas de Información. El RAD cuenta con una participación intensa del usuario, sesiones JAD (Joint Application Design), prototipado, herramientas CASE (Computer Aided Software Engineering) integradas y generadores de código.

JAD es una técnica de definición de requisitos y de diseño de la interfaz de usuario, basada en reuniones participativas entre clientes, directiva y desarrolladores – ver [Davidson, 1999]. En dichas reuniones los temas a tratar se centran más en el negocio que en el asunto técnico. Lógicamente está más orientado a proyectos de cliente (o bien sistemas a medida, como también se los conoce), y permite recolectar requisitos eficientemente.

Las herramientas CASE son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de estas en términos de tiempo y de dinero – como se indica en [Martin, 1991]. Además, se añade que estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, etc.

La metodología JAD se compone de cuatro etapas [Cline, 1999]:

1. Etapa de Planificación de Requisitos: esta etapa requiere que los usuarios con un vasto conocimiento de los procesos de la compañía determinen cuáles serán las funciones del sistema. Debe darse una discusión estructurada sobre los problemas de la compañía que necesitan solución.
2. Etapa de Diseño: consiste en un análisis detallado de las actividades de la compañía en relación con el sistema propuesto. Los usuarios participan activamente

en talleres bajo la tutela de los profesionales de la informática. En ellos descomponen funciones y definen entidades asociadas con el sistema. Una vez se completa el análisis se crean los diagramas que definen las alteraciones entre los procesos y la data.

3. Construcción: en la etapa de construcción el equipo de desarrolladores trabajando de cerca con los usuarios finalizan el diseño y la construcción del sistema. La construcción de la aplicación consiste en una serie de pasos donde los usuarios tienen la oportunidad de afirmar los requisitos y repasar los resultados.
4. Implementación: esta última etapa envuelve la implementación del nuevo producto y el manejo de cambio del viejo al nuevo sistema. Se hacen pruebas comprensivas y se adiestran los usuarios.

Las ventajas y desventajas de esta metodología son las siguientes:

- Ahorro de tiempo durante el desarrollo del sistema. Mayor velocidad y menores costes pueden repercutir en la calidad del sistema (por ejemplo, debido a falta de atención en controles internos).
- Estrecha correspondencia entre las especificaciones del usuario y las especificaciones del sistema.
- Pueden producirse inconsistencias entre diseños internos y entre sistemas.
- Es muy aconsejable cuando la velocidad de desarrollo es importante (cambios rápidos de las condiciones del negocio). Es probable incurrir en violaciones de estándares de programación relacionadas con nomenclaturas inconsistentes e insuficiente documentación.
- Permite cambiar rápidamente el diseño de los sistemas cuando los usuarios lo demandan. Dificultades para reutilizar los módulos para futuros sistemas.
- Los sistemas son optimizados por los usuarios involucrados en la metodología del RAD. Carencia de un diseño escalable dentro del sistema.
- Se concentra en los elementos esenciales del sistema, desde el punto de vista del usuario. Falta de atención de la futura administración del sistema dentro de los sistemas existentes (por ejemplo, falta de integración con el modelo de datos organizacional y facilidades de recuperación del sistema). El usuario se compromete y se hace propietario del sistema. Altos costes de compromiso por parte del personal clave.

5.2.2. METODOLOGÍA DEL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado se describe en [Jacobson, 2000] como un marco de desarrollo de software que se caracteriza

por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de Rational, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP (Rational Unified Process). Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto [Rational, 1998].

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales, los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. Los flujos de trabajo definidos en RUP son los siguientes:

- **Modelo del Negocio:** describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimiento:** define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y Diseño:** describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba:** busca los defectos a lo largo del ciclo de vida.
- **Instalación o despliegue:** produce una versión del producto y realiza actividades (empaquetado, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos y control de versiones.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Las fases de esta metodología se describen a continuación:

- Fase de concepción o inicio tiene por finalidad definir la visión, los objetivos y el alcance del proyecto, tanto desde el punto de vista funcional como del técnico, obteniéndose como uno de los principales resultados una lista de los casos de uso y una lista de los factores de riesgo del proyecto. El principal esfuerzo está radicado en el Modelamiento del Negocio y el Análisis de Requerimientos. Es la única fase que no necesariamente culmina con una versión ejecutable.
- Fase de elaboración tiene como principal finalidad completar el análisis de los casos de uso y definir la arquitectura del sistema, además se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo con el alcance definido.
- Fase de construcción está compuesta por un ciclo de varias iteraciones, en las cuales se van incorporando sucesivamente los casos de uso, de acuerdo con los factores de riesgo del proyecto. Este enfoque permite por ejemplo contar en forma temprana con versiones del sistema que satisfacen los principales casos de uso. Los cambios en los requerimientos no se incorporan hasta el inicio de la próxima iteración.
- Fase de transición se inicia con una versión “beta” del sistema y culmina con el sistema en fase de producción.

5.2.3. METODOLOGÍA DE KENDALL Y KENDALL

Según esta metodología el ciclo de vida de un sistema consta de siete partes: siendo la primera la identificación del problema, la segunda identificación de requisitos de información, la tercera es el análisis de las necesidades del sistema, la cuarta es el diseño del sistema recomendado, la quinta desarrollo y documentación del sistema, la sexta prueba y mantenimiento y la última implementación y evaluación. Cada fase se explica por separado, pero nunca se realizan como pasos aislados, más bien es posible que algunas actividades se realicen de manera simultánea, y algunas de ellas podrían repetirse.

“El ciclo de vida del desarrollo de sistemas (SDLC, Systems Development Life Cycle) es un enfoque por fases para el análisis y el diseño cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades del analista y el usuario.” (Kendall & Kendall) [Kendall, 2014]

La metodología de Kendall y Kendall es un ciclo de desarrollo de los sistemas, y se desarrolla en siete etapas las cuales son:

1. Identificación de problemas, oportunidades y objetivos: Esta fase es crucial para el éxito del resto del proyecto. Requiere que se observe de forma objetiva lo que ocurre en una organización, luego en conjunto con otros miembros de la organización hacer notar los problemas. Las oportunidades son aquellas situaciones que se considera que pueden mejorarse, perfeccionarse mediante el uso de los sistemas de información. También es un componente importante de la primera fase. En esta etapa se deberá descubrir lo que la organización intenta realizar, luego determinar si el uso de los sistemas de información apoyaría a la organización para alcanzar sus metas.
2. Determinación de los requerimientos de información: Esto se hace a partir de los usuarios particularmente involucrados. Para determinar los requerimientos de información dentro de una organización pueden utilizarse diversos instrumentos, los cuales incluyen: muestreo, el estudio de los datos y formas usadas para la organización, la entrevista, los cuestionarios; la observación de la conducta de quien tomó la decisión, así como de su ambiente. Se hace todo lo posible por identificar qué información requiere el usuario para desempeñar sus tareas.
3. Análisis de las necesidades del sistema: Se analizan las necesidades propias del sistema, para ello existen herramientas y técnicas diseñadas para tal fin. Estas herramientas pueden ser, entre otras, el uso de los diagramas de flujo de datos. Mediante el uso de esta técnica se representa de forma gráfica y estructurada la entrada de datos a la organización y los procesos de salida de la información. También se analizan las decisiones estructuradas por realizar, que son decisiones donde las condiciones, condiciones alternativas, acciones y reglas de acción podrán determinarse.
4. Diseño del sistema recomendado: Se usa la información recolectada con anterioridad y se elabora el diseño lógico de sistemas de información, se diseña también procedimientos precisos de captura de datos, con la finalidad de que los datos que se introducen en el sistema de información sean los correctos. Esta etapa también incluye el diseño de los archivos o la base de datos que almacenará aquellos datos requeridos por quien toma las decisiones en la organización.
5. Desarrollo y documentación del software: Dentro de las técnicas estructuradas para el diseño y documentación del software se tienen: el método HIPO (Hierarchy-Input-Process-Output), los diagramas de flujo, los diagramas Nassi Schneiderman, los diagramas Warnier-Orr y el pseudocódigo. Es aquí donde se transmite al programador los requerimientos de programación.
6. Pruebas y mantenimiento del sistema: Todo sistema de información debe probarse antes de ser utilizado, ya que el costo es menor si se detectan los problemas antes de que entre en funcionamiento. En un principio, se hace una serie de pruebas, con datos tipo, para identificar las posibles fallas del sistema, más adelante, se utilizarán los datos del sistema real.

7. Implantación y evaluación del sistema: Esta es la última etapa del desarrollo del sistema, esto incluye el adiestramiento que el usuario requerirá. Aunque la evaluación del sistema se plantea como parte integrante de la última etapa del ciclo de desarrollo de los sistemas; realmente la evaluación toma parte de cada una de las etapas. Uno de los criterios fundamentales que debe satisfacerse, es que el futuro usuario utilice el sistema desarrollado.

5.2.4. METODOLOGÍA ORIENTADA A OBJETOS

De acuerdo con [Booch, 2007], el análisis y diseño orientado a objetos (ADOO) es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Este enfoque representa un dominio absoluto en términos de conceptos compuestos por verbos y sustantivos, clasificados de acuerdo con su dependencia funcional. En este método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el lenguaje unificado de modelado, UML (Unified Modeling Language). Una parte de la notación permite dibujar clases y objetos, pero UML es más completo que para ser solo usado en POO (Programación Orientada a Objetos). ADOO aplica técnicas de modelado de objetos para analizar los requerimientos para un contexto - por ejemplo, un sistema de negocio, un conjunto de módulos de software - y para diseñar una solución para mejorar los procesos involucrados. No está restringido al diseño de programas de computadora, sino que cubre sistemas enteros de distinto tipo. Las metodologías de análisis y diseño más modernas son casos de uso guiados a través de requerimientos, diseño, implementación, pruebas, y despliegue.

La Metodología Orientada a Objetos ha derivado de las metodologías anteriores a ésta. Así como los métodos de diseño estructurado realizados guían a los desarrolladores que tratan de construir sistemas complejos utilizando algoritmos como sus bloques fundamentales de construcción, similarmente los métodos de diseño orientado a objetos han evolucionado para ayudar a los desarrolladores a explotar el poder de los lenguajes de programación basados en objetos y orientados a objetos, utilizando las clases y objetos como bloques de construcción básicos. Las clases son ideas generales, los objetos son instancias de esas clases. Fue un cambio radical a la programación estructurada. Las clases tienen atributos y métodos. Al menos, un método constructor y un destructor. Este tipo de programación contempla conceptos como herencia, sobrecarga, polimorfismo y demás.

Actualmente el modelo de objetos ha sido influenciado por un número de factores no sólo de la Programación Orientada a Objetos (POO). Además, el modelo de objetos ha

probado ser un concepto uniforme en las ciencias de la computación, aplicable no sólo a los lenguajes de programación sino también al diseño de interfaces de usuario, bases de datos y arquitectura de computadoras por completo. La razón de ello es, simplemente, que una orientación a objetos ayuda a hacer frente a la inherente complejidad de muchos tipos de sistemas.

Se define a un objeto como "una entidad tangible que muestra alguna conducta bien definida" [Conceptos, 2018a]. Un objeto "es cualquier cosa, real o abstracta, acerca de la cual almacenamos datos y los métodos que controlan dichos datos" [Conceptos, 2018b]. Las clases son ideas generales, los objetos son instancias de esas clases.

Los objetos tienen una cierta "integridad" la cual no deberá ser violada. En particular, un objeto puede solamente cambiar estado, conducta, ser manipulado o estar en relación con otros objetos de manera apropiada a este objeto.

Actualmente, el Análisis Orientado a Objetos (AOO) va progresando como método de análisis de requisitos por derecho propio y como complemento de otros métodos de análisis. En lugar de examinar un problema mediante el modelo clásico de entrada-proceso-salida (flujo de información) o mediante un modelo derivado exclusivamente de estructuras jerárquicas de información, el AOO introduce varios conceptos nuevos.

Una clase es una plantilla para objetos múltiples con características similares. Las clases comprenden todas esas características de un conjunto particular de objetos. Cuando se escribe un programa en lenguaje orientado a objetos, no se definen objetos verdaderos sino se definen clases de objetos.

Una instancia de una clase es otro término para un objeto real. Si la clase es la representación general de un objeto, una instancia es su representación concreta. A menudo se utiliza indistintamente la palabra objeto o instancia para referirse, precisamente, a un objeto.

En los lenguajes orientados a objetos, cada clase está compuesta de dos cualidades: atributos (estado) y métodos (comportamiento o conducta). Los atributos son las características individuales que diferencian a un objeto de otro (ambos de la misma clase) y determinan la apariencia, estado u otras cualidades de ese objeto. Los atributos de un objeto incluyen información sobre su estado.

Los métodos de una clase determinan el comportamiento o conducta que requiere esa clase para que sus instancias puedan cambiar su estado interno o cuando dichas instancias son llamadas para realizar algo por otra clase o instancia. El comportamiento es la

única manera en que las instancias pueden hacerse algo a sí mismas o tener que hacerles algo. Los atributos se encuentran en la parte interna mientras que los métodos se encuentran en la parte externa del objeto.

Algunas ventajas de esta metodología son las siguientes:

- Reutilización: las clases están diseñadas para que se reutilicen en muchos sistemas. Para maximizar la reutilización, las clases se construyen de manera que se puedan adaptar a los otros sistemas. Un objetivo fundamental de las técnicas orientadas a objetos es lograr la reutilización masiva al construir el software [Reutilización, 2018].
- Estabilidad: las clases diseñadas para una reutilización repetida se vuelven estables, de la misma manera que los microprocesadores y otros chips se hacen estables [Estabilidad, 2018].
- Calidad: los diseños suelen tener mayor calidad, puesto que se integran a partir de componentes probados, que han sido verificados y pulidos varias veces [Calidad, 2018].
- Un diseño más rápido: las aplicaciones se crean a partir de componentes ya existentes. Muchos de los componentes están contruidos de modo que se pueden adaptar para un diseño particular [Teoría, 2018].
- Mantenimiento más sencillo: el programador encargado del mantenimiento cambia un método de clase a la vez. Cada clase efectúa sus funciones independientemente de las demás [Teoría, 2018].
- Interacción: el software de varios proveedores puede funcionar como conjunto. Un proveedor utiliza clases de otros. Existe una forma estándar de localizar clases e interactuar con ellas. El software desarrollado de manera independiente en lugares ajenos debe poder funcionar en forma conjunta y aparecer como una sola unidad ante el usuario [Teoría, 2018].
- Migración: las aplicaciones ya existentes, sean orientadas a objetos o no, pueden preservarse si se ajustan a un contenedor orientado a objetos, de modo que la comunicación con ella sea a través de mensajes estándar orientados a objetos [Teoría, 2018].

5.2.5. METODOLOGÍA DE SISTEMAS BLANDOS

Esta metodología fue creada por Peter Checkland y un grupo del Departamento de Sistemas de la Universidad Británica de Lancaster, quienes trabajaron por varios años en el desarrollo de una metodología sistémica flexible, diseñada para hacer frente a situaciones problemáticas, las cuales son difíciles de definir, tienen un componente social y política grande.

La metodología de los sistemas blandos, SSM, (Soft Systems Methodology), tal y como se define en [Checkland, 2006], ha sido desarrollada a partir de este ciclo continuo de la intervención en las malas estructuras de gestión de los problemas y aprender de los resultados. Los Sistemas Blandos es una rama de la teoría de sistemas diseñados específicamente para su uso y aplicación en una variedad de contextos del mundo real.

La Metodología de Sistemas Blandos consta de las siguientes etapas:

1. Primera fase, Situación no estructurada: se observan acontecimientos que suceden en la situación - problema, aún sin tener una idea clara de las interrelaciones de los elementos que la conforman. En esta etapa se debe empezar a delimitar el sistema a cuyo estudio se aboca, así como a definir el entorno de este.
2. Segunda fase, situación estructurada: se expresa la situación problemática. En esta etapa se concatenan los elementos que integran la situación - problema, haciendo una descripción del pasado - presente y su consecuencia en el futuro, y recogiendo aspiraciones, intereses y necesidades del sistema contenedor del problema. Todo esto contribuirá a lograr el objetivo de expresar pictográficamente la situación-problema. Representar la situación problema mediante diagramas "visiones enriquecidas", donde se muestren:
 - a. Estructura.
 - b. Procesos.
 - c. Hechos de la organización que puedan ser relevantes para la definición del problema.
 - d. Clima de la situación: relación entre estructura y proceso.
3. Tercera fase, la elaboración de definiciones básicas: la información que se reúne en la segunda etapa permite identificar posibles "candidatos a problemas" y buscar la correspondiente "solución". Dicha solución, que implica un cambio (un proceso de transformación) de la realidad social, se expresa a través de lo que en la SSM se denomina definición básica. Los elementos básicos especificados conforman una lista de verificación útil para probar que se tiene una definición raíz válida correspondiente a una situación planteada.
4. Cuarta fase, la elaboración de modelos conceptuales: una vez descrita la definición básica, en esta etapa se genera un modelo conceptual de lo expresado en ella, es decir construir un Modelo Sistema de Actividades (MSA) necesarias para lograr la transformación descrita en la definición. Este modelo conceptual permitirá llevar a cabo lo que se especifica en la Definición Básica, convirtiéndose adecuadamente en un reporte de las actividades que el sistema debe hacer para convertirse en el sistema nombrado en la definición. La elaboración del Modelo Conceptual y modelos se realiza de manera que la definición expresa un sistema de actividad a realizar para llevar a cabo el proceso de transformar la realidad social, sus elementos serán expresados a través de acciones a efectuar, y esto es

posible a través de palabras que expresen acción, es decir, mediante verbos. En esta etapa se aplica la parte técnica de la Metodología de Sistemas Blandos, es decir el "cómo" llevar a cabo la transformación definida a través del "qué" anteriormente, para ello la técnica del modelado consiste en ensamblar sistemáticamente una agrupación mínima de verbos que describen actividades que son necesarias en un sistema especificado en la Definición Básica y que están unidas gráficamente en una secuencia de acuerdo con la lógica.

5. Quinta fase, comparación de la cuarta fase y la segunda fase: el objetivo de esta etapa es comparar los modelos conceptuales elaborados en la etapa 4 con la situación problema analizada en la etapa 2 de Percepción Estructurada, esto se debe hacer junto con los participantes interesados en la situación problema, con el objeto de generar un debate acerca de posibles cambios que se podrían introducir para así aliviar la condición del problema. Además, es necesario comparar para determinar si el modelo requiere ser mejorado su conceptualización, elaborada en la etapa anterior.
6. Sexta fase, cambios factibles y deseables: una vez concluida la comparación de los Modelos Conceptuales con la situación de la realidad problemática estructurada y determinando las diferencias, se procede a ejecutar aquellas medidas propuestas en la etapa anterior que lleva a mejorar la situación del problema. Estos posibles cambios pueden hacerse en diversos planos; en estructura, en procedimientos y en actitudes. A propósito de la etapa anterior de comparación esta consistía en usar la comparación entre los Modelos Conceptuales y "lo que es", para generar la discusión de los cambios de cualquiera de las tres formas descritas anteriormente.
7. Séptima fase, implantación de los cambios en el mundo real: una vez que se han acordado los cambios, la habilitación en el mundo real quizás sea inmediata. O su introducción quizá cambie la situación, de forma que, aunque el problema generalmente percibido ha sido eliminado, emergen nuevos problemas y quizás a estos nuevos problemas se enfrenten con la ayuda de la SSM.

5.2.6. METODOLOGÍA DEL MODELO EN V

El estándar ISO/IEC 61508 de seguridad funcional para sistemas eléctricos, electrónicos y electrónica programable define en [Safety, 2010] el modelo en V como una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño. La codificación forma el vértice de la V, con el análisis y el diseño a la izquierda y las pruebas y el mantenimiento a la derecha. La Figura 35 detalla los niveles típicos de un diagrama de modelo en V. También se representan las transiciones entre las diferentes etapas que son comúnmente utilizadas.

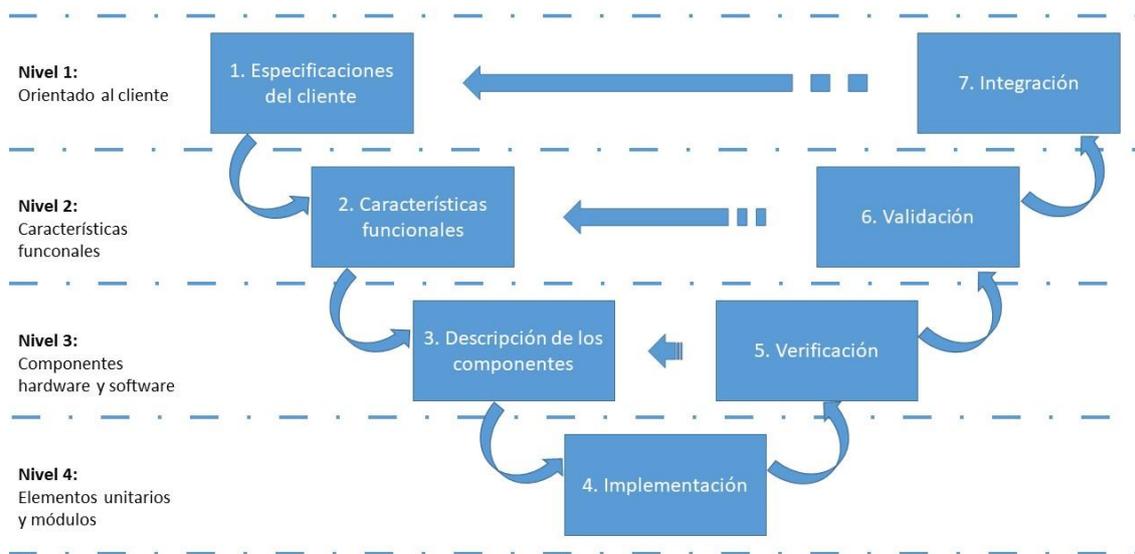


Figura 35. Diagrama de Modelo en V.

La unión mediante líneas discontinuas entre las fases de la parte izquierda y las pruebas de la derecha representa una doble información. Por un lado, sirve para indicar en qué fase de desarrollo se deben definir las pruebas correspondientes. Por otro sirve para saber a qué fase de desarrollo hay que volver si se encuentran fallos en las pruebas correspondientes.

En los niveles lógicos del 1 al 4, para cada fase del desarrollo, existe una fase correspondiente o paralela de verificación o validación. Esta estructura obedece al principio de que para cada fase del desarrollo debe existir un resultado verificable.

En la misma estructura se advierte también que la proximidad entre una fase del desarrollo y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V. La longitud de esta separación intenta ser proporcional a la distancia en el tiempo entre una fase y su homóloga de verificación.

El nivel 1 está orientado al “cliente”. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones.

El nivel 2 se dedica a las características funcionales del sistema propuesto. Puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.

El nivel 3 define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.

El nivel 4 es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

Por lo tanto, el modelo en V hace más explícita la parte del desarrollo que conlleva las iteraciones y repeticiones de trabajo. En contraposición con el modelo en cascada, el modelo en V sí da visibilidad a las tareas asociadas a las iteraciones y repeticiones necesarias tras las diferentes fases de verificación del desarrollo del producto. Mientras el foco del modelo en cascada se sitúa en los documentos y productos desarrollados, el modelo en V se centra en las actividades y la corrección.

Las ventajas de esta metodología son las siguientes:

- La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos.
- Es un modelo sencillo y de fácil aprendizaje.
- Hace explícito parte de la iteración y trabajo que hay que revisar.
- Especifica bien los roles de los distintos tipos de pruebas a realizar.
- Involucra al usuario en las pruebas.

Por otro lado, esta metodología presenta las siguientes desventajas:

- Es difícil que el cliente exponga explícitamente todos los requisitos.
- El cliente debe tener paciencia pues obtendrá el producto al final del ciclo de vida.
- Las pruebas pueden ser caras y, a veces, no lo suficientemente efectivas.
- El producto final obtenido puede que no refleje todos los requisitos del usuario.

5.2.7. METODOLOGÍA EXTREME PROGRAMMING

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, tal y como se define en [Beck, 2000]. Promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Extreme Programming (XP) se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y que no presente inconvenientes a la hora de trabajar sobre cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Las fases de esta metodología son las siguientes:

- 1) Fase de Planificación del proyecto: en esta primera fase se debe hacer una recopilación de todos los requerimientos del proyecto, también debe haber una interacción con el usuario, y se debe planificar bien entre los desarrolladores del proyecto que es lo que se quiere para el proyecto para así lograr los objetivos finales.
- 2) Fase de Diseño: se sugiere que hay que conseguir diseños simples y sencillos. Para procurar hacerlo todo lo menos complicado posible para el usuario o cliente, para conseguir un diseño fácil de entender e implementar. Porque a la larga costará menos tiempo y esfuerzo desarrollarlo. En esta fase se logrará crear parte del proyecto, la parte física (lo bonito) y la interfaz que tendrá el usuario o cliente con el proyecto.
- 3) Fase de Codificación: a la hora de codificar la presencia del usuario es aún más necesaria. No se debe olvidar que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. En esta fase de la codificación los clientes y los desarrolladores del proyecto deben estar en comunicación para que los desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere, en esta fase está incluido todo lo de codificación o programación por parte de los desarrolladores del proyecto.
- 4) Fase de Pruebas: uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que se vayan implementando. Para esta fase lo que se implementa son pruebas que se le hacen al proyecto o como ya se dijo a los códigos que se vayan implementando.

5.2.8. METODOLOGÍA SCRUM

Tal y como se define en [Rising, 2000], SCRUM es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa, ROI, (Return of Investment). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, autogestión e innovación.

El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio.

Los roles involucrados en SCRUM son los siguientes:

- Scrum master: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI.
- Product owner: Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del ROI del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las vuelve a priorizar de forma regular.
- Team: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint.

A continuación, se indican algunas definiciones propias de esta metodología:

- Product Backlog: Conjunto de requisitos denominados historias descritos en un lenguaje no técnico y priorizados por valor de negocio, o lo que es lo mismo, por retorno de inversión considerando su beneficio y coste. Los requisitos y prioridades se revisan y ajustan durante el curso del proyecto a intervalos regulares.
- Sprint Planning: Reunión durante la cual el Product Owner presenta las historias del backlog por orden de prioridad. El equipo determina la cantidad de historias que puede comprometerse a completar en ese sprint, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir.
- Sprint: Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del Product Backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo.
- Sprint Backlog: Lista de las tareas necesarias para llevar a cabo las historias del sprint.
- Daily sprint meeting: Reunión diaria de, como máximo 15 minutos en la que el equipo se sincroniza para trabajar de forma coordinada. Cada miembro comenta qué hizo el día anterior, qué hará hoy y si hay impedimentos.
- Demo y retrospectiva: Reunión que se celebra al final del sprint y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, en la retrospectiva, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discute acerca de cómo perfeccionarlos.

5.3. Metodología de desarrollo de la plataforma

Tras el análisis mostrado en la sección anterior, y después de haber analizado todas esas metodologías, se ha decidido usar como base la metodología del Modelo en V. Los motivos por los que se ha optado por usar esta metodología son los siguientes:

- El modelo en V hace más explícita la tarea parte de la iteración de las actividades del proceso.
- Las pruebas de cada fase ayudaran a corregir posibles errores sin tener que esperar a que sean rectificadas en las etapas posteriores.
- Con las pruebas unitarias y de integración se consigue obtener mejores resultados en los correspondientes desarrollos.
- Este tipo de metodología se adapta mejor a las características de la plataforma de robótica educativa. El desarrollo de esta plataforma comprende desarrollo hardware, firmware y software.
- Por último, esta metodología es ampliamente utilizada para el desarrollo de productos que combinan componentes hardware y software,

Para aplicar la metodología de Modelo en V para el desarrollo de la plataforma de robótica educativa se han definido las siguientes etapas:

1. Primera etapa: especificaciones de la plataforma.
2. Segunda etapa: diseño de la arquitectura.
3. Tercera etapa: descripción de los componentes.
4. Cuarta etapa: implementación.
5. Quinta etapa: pruebas de integración.
6. Sexta etapa: pruebas de sistema.
7. Séptima etapa: pruebas con alumnos.

La Figura 36 representa los diferentes niveles que componen la metodología de desarrollo, las etapas en las que se ha dividido, y las transiciones lógicas que pueden ocurrir en cada una de las etapas:

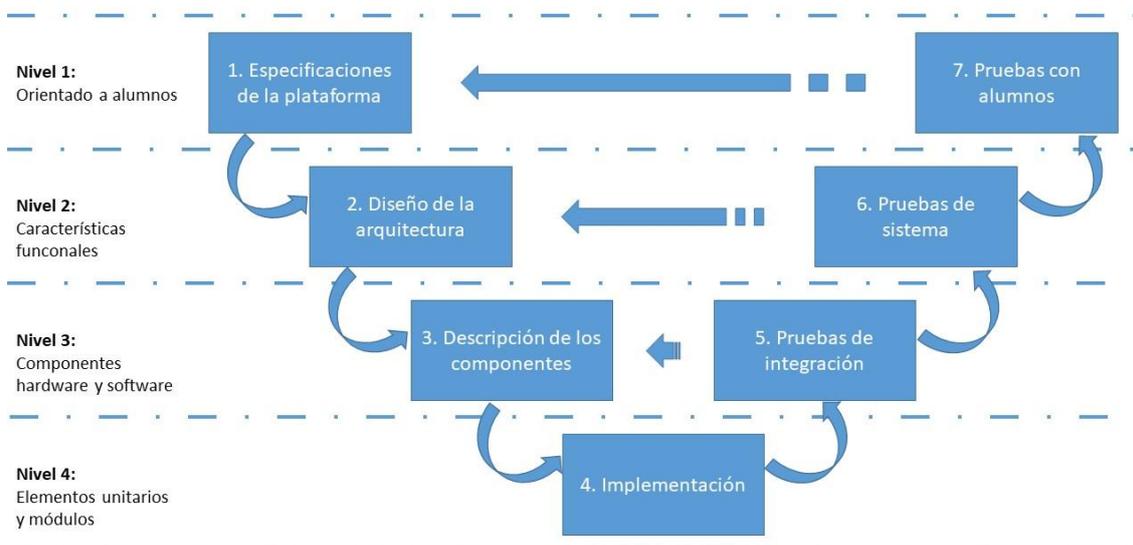


Figura 36. Modelo en V del diseño de la plataforma.

El nivel 1 está orientado a los alumnos. Se compone del análisis de las especificaciones y de las pruebas de aceptación de la plataforma.

El nivel 2 se dedica a las características funcionales del sistema propuesto. Puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.

El nivel 3 define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.

El nivel 4 es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

A lo largo de la aplicación de la metodología arriba expuesta, se elaborará toda la documentación relacionada con cada una de las etapas indicadas.

5.3.1. PRIMERA ETAPA: ESPECIFICACIONES DE LA PLATAFORMA

El primer paso al aplicar la metodología del Modelo en V es definir las especificaciones del producto. El producto es la plataforma de robótica educativa. Las especificaciones generales de la plataforma son la libertad de reconfiguración, flexibilidad, versatilidad y escalabilidad. Además, la plataforma debe permitir la elaboración de prácticas metodológicas activas de aprendizaje centradas principalmente en el alumno. Asimismo, la propia plataforma debe potenciar las relaciones de grupo y el trabajo colaborativo.

La Tabla 65 contiene un resumen de las especificaciones. Esta Tabla lista las funcionalidades y una breve descripción de estas.

Tabla 65. Resumen de las especificaciones de la plataforma.

Funcionalidades	Descripción
Reconfiguración	Las conexiones o los parámetros de configuración pueden ser modificados antes de su uso, o mientras es usada.
Escalabilidad	La complejidad de la plataforma puede ser modificada de acuerdo con las necesidades de uso.
Compatibilidad	Las conexiones con Shields de Arduino y Smart Devices son posibles y sencillas.
Concurrencia	El procesado y las comunicaciones pueden ser paralelados.

Funcionalidades	Descripción
Protección	Protecciones de integridad y protecciones menores están incluidas. Además, se permite la inclusión de protecciones por parte del usuario.
Prototipado	Prototipos electrónicos del usuario se pueden incluir, tanto por profesores, como por alumnos, con el fin de completar la plataforma.
Flexibilidad	La plataforma es susceptible de modificaciones o adaptaciones.

La plataforma puede ser usada por los alumnos directamente. También podría ser reconfigurada por los propios alumnos durante el proceso de aprendizaje. Además, la plataforma podría ser configurada por los instructores previamente a la sesión de aprendizaje. Las configuraciones o reconfiguraciones pueden ser realizadas “in situ” o a través de Internet.

5.3.2. SEGUNDA ETAPA: DISEÑO DE LA ARQUITECTURA

Se define plataforma como el conjunto de módulos que la forman. Los módulos son los elementos hardware que implementan diferentes funcionalidades genéricas como parte de la plataforma y cumplen con unas conexiones estándar para poder ser incluidas sencillamente como parte de la plataforma mediante unos conectores normalizados. La Figura 37 muestra los diferentes componentes que se van a desarrollar para formar la plataforma.



Figura 37. Componentes de la plataforma a desarrollar.

Los módulos que conforman la plataforma son incluidos en el plano horizontal. Adicionalmente, a los módulos se les pueden conectar periféricos para incluir funcionalidades más específicas de control o comunicación con elementos externos a la plataforma, pero que forman parte del sistema completo. Los periféricos se añaden en el eje vertical de la plataforma. Estos elementos externos a la plataforma son los dispositivos auxiliares que conforman la herramienta de aprendizaje. Elementos externos pueden ser sensores o actuadores. Finalmente se define sistema como el conjunto global de módulos, periféricos y elementos externos. La Figura 38 muestra los diferentes elementos que componen la plataforma.

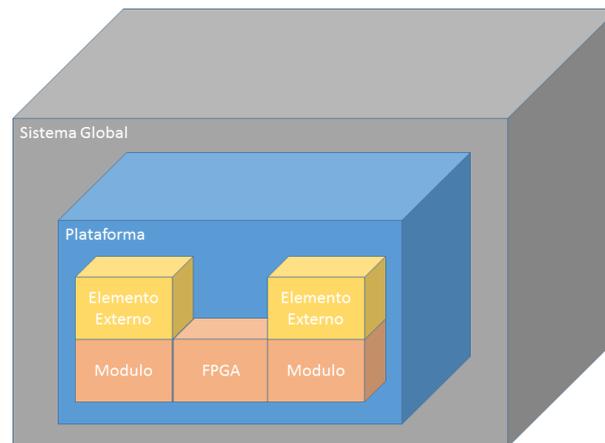


Figura 38. Arquitectura de la plataforma.

De esta manera se posibilita un grado de flexibilidad, versatilidad y escalabilidad en función del número de módulos y periféricos que conforman la plataforma. El tamaño de la plataforma dependerá de la complejidad del sistema completo, tanto del número de elementos externos que se quieran manejar o monitorizar, como de las necesidades educativas.

El pilar básico de la arquitectura es un diseño modular, es decir, es un conjunto formado por módulos claramente diferenciados, que puedan ser intercambiados. Es decir, la plataforma la pueden formar varios módulos iguales, o módulos de diferentes tipos. Otra característica importante de la plataforma es que sea capaz de poder reconfigurarse durante su uso de una manera sencilla. Por ello se ha escogido una FPGA como eje y base del diseño.

La plataforma es capaz de manejar robots y obtener información de estado de estos. Por otro lado, envía la información de estado a un equipo que tenga conexión a Internet. Este equipo podría ser un ordenador personal, una Raspberry Pi o cualquier otro equipo que disponga de un puerto de comunicaciones Bluetooth y tenga conexión a Internet. Aunque también se dispone de compatibilidad para dispositivos tales como Smartphone o Tablet para ser controlado por el usuario o usuarios que estén trabajando con la plataforma sin una conexión física con ella. La Figura 39 muestra estas conexiones entre los diferentes elementos.

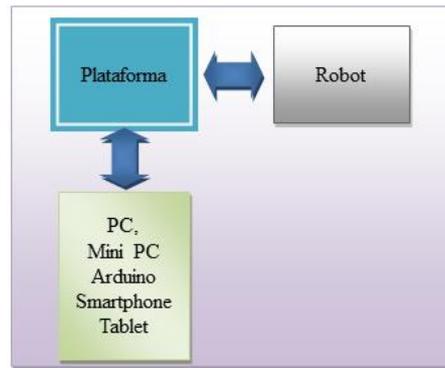


Figura 39. Comunicaciones de la plataforma.

5.3.3. TERCERA ETAPA: DESCRIPCIÓN DE LOS COMPONENTES

La propuesta consta de dos tipos de componentes: el módulo principal y el módulo secundario. Estos componentes son los siguientes:

- Un módulo principal al que se conectan cuatro módulos secundarios.
- El módulo principal, además de comunicarse con los módulos secundarios, tiene la capacidad de comunicarse por Bluetooth con Smart Devices separados físicamente de la plataforma y se le pueden ensamblar Shields de Arduino.
- Los módulos secundarios son también compatibles con Shields de Arduino, con lo que se pueden ensamblar estos últimos a los módulos secundarios con facilidad.

Hay otro tipo de componentes que no forman parte del desarrollo de la plataforma, pero ésta debe ser compatible con ellos. Estos componentes, los Elementos Externos, son dispositivos que harán de interfaz entre la plataforma y diferentes tipos de circuitos electrónicos o plataformas robóticas para que puedan ser integradas al sistema. Por último, y gracias a la comunicación Bluetooth, la plataforma puede tener acceso, o ser accedida a través de Internet, usando Smart Devices para tal efecto.

5.3.4. CUARTA ETAPA: IMPLEMENTACIÓN

De acuerdo con la arquitectura planteada, la etapa de implementación contiene tres grandes desarrollos: el desarrollo hardware, el desarrollo firmware y el desarrollo software.

El desarrollo hardware contempla la elaboración de los PCB del módulo principal y del módulo secundario. El desarrollo firmware consiste en la implementación de la lógica incluida dentro de la FPGA. El desarrollo software contiene el diseño de los programas generados para las placas de Arduino Nano.

Por la complejidad de esta etapa, se ha decidido usar una metodología a medida para cada uno de los desarrollos que requiere esta cuarta etapa. Por lo tanto, se ha incluido una sección propia para documentar esas metodologías y los trabajos realizados en cada uno de dichos desarrollos.

5.3.5. QUINTA ETAPA: PRUEBAS DE INTEGRACIÓN

Para el inicio de esta etapa es necesario finalizar la etapa previa de implementación, la cual consiste en tres desarrollos que se realizan de manera secuencial. Por este motivo, las pruebas de integración se realizarán a la finalización de cada uno de los desarrollos, siendo el desarrollo hardware incluido como parte del desarrollo firmware, cuando este último haya finalizado. Estas pruebas de integración son definidas al final del desarrollo firmware. De la misma manera, las pruebas de integración de la plataforma se hacen a la finalización del desarrollo software, incluyendo en ellas, tanto el desarrollo hardware, como el desarrollo firmware. Por lo tanto, las pruebas de integración de la plataforma se definen al final del desarrollo software.

5.3.6. SEXTA ETAPA: PRUEBAS DE SISTEMA

Las pruebas de sistema se basan en escenarios educativos. Dadas las características anteriormente expuestas, la plataforma es totalmente válida para poder ser empleada en actividades educativas. Un claro ejemplo es el prototipado electrónico en grupos de hasta cuatro alumnos.

Es posible elaborar prácticas de programación de una red de Arduino incluyendo hasta cuatro de estos dispositivos, pudiendo ser complementada con el uso de sensores y actuadores. De esta manera se pueden implementar proyectos de redes de sensores, gestionando los datos de la red de sensores para analizar la información obtenida y realizar acciones en función de los valores adquiridos.

Los escenarios educativos se han considerado como entidades propias que requieren un diseño electromecánico adaptado a unas necesidades educativas. Por el contenido del desarrollo de los escenarios educativos, su descripción se ha incluido en una sección independiente.

5.3.7. SÉPTIMA ETAPA: PRUEBAS CON ALUMNOS

Esta última etapa consiste en la realización de sesiones educativas en las que se use la plataforma, con el fin de probarla, y los escenarios educativos elaborados a lo largo de la etapa anterior.

CAPÍTULO 6

6. DESARROLLO HARDWARE

En este capítulo se recoge toda la información recopilada durante diseño hardware del desarrollo de la arquitectura de la plataforma de robótica educativa. De acuerdo con la arquitectura definida en el capítulo 5, es necesario el desarrollo de dos tipos de componentes hardware. En primer lugar, el módulo principal, cuya finalidad es la conexión de diferentes elementos tales como la tarjeta que contiene la FPGA (Field Programmable Gate Array), un Shield Arduino, el módulo de comunicaciones Bluetooth y los módulos secundarios. Por otro lado, se requiere el desarrollo de unos módulos secundarios que contendrán elementos como la tarjeta Arduino Nano, un Shield Arduino y conectores para elementos externos a la plataforma. Estos elementos externos pueden ser servos, sensores u otros dispositivos de uso en aplicaciones robóticas.

6.1. Metodología de desarrollo hardware

El desarrollo hardware de la Plataforma se encuentra ubicado en la etapa 5, referente a la implementación de la Metodología de Modelo en V usado a lo largo de la elaboración de la arquitectura de la Plataforma, tal y como se describe en el capítulo 5 a lo largo de la sección 3.

Como parte de la Metodología de Modelo en V se ha elaborado una metodología de desarrollo de hardware. Esta metodología sigue una estructura similar, pero se incluyen detalles propios de un desarrollo hardware. La metodología hardware seguida consiste en varios pasos que se siguen tanto para el diseño hardware del módulo principal, como para el diseño hardware del módulo secundario. Los pasos por seguir son los siguientes:

- En primer lugar, se indican las especificaciones identificadas que tienen relación con el hardware de cada diseño, siendo comunes para ambos módulos.
- Posteriormente se identifican los elementos hardware que van a formar parte de la Plataforma y que serán fundamentales a lo largo del diseño de los esquemáticos.
- En tercer lugar, se procede a la elaboración de los esquemáticos.
- Tras la elaboración de los esquemáticos, en un siguiente paso, se procede al diseño físico de la tarjeta, es decir, dimensiones, conexiones y ubicación de los distintos componentes.

- Una vez concluidas los pasos anteriormente mencionados, se generan los archivos de fabricación, GERBER, para su envío al fabricante de PCB.
- Tras la recepción del PCB se ensamblan todos los componentes.
- Y al final se realizan pruebas de verificación para comprobar que el diseño hardware se ajusta a las especificaciones.

Para acometer los pasos 3, 4 y 5, relacionados con los esquemáticos, el diseño del PCB y los archivos de fabricación, la herramienta CAD EAGLE [EAGLE, 2018] ha sido usada. Se trata de un software que dispone de una versión gratuita para su uso en fines educativos, como es el caso particular en el que se enmarca esta tesis. Este es un tipo de software que pertenece a la categoría de los EDA (Electronic Design Automation) que sirven para la automatización de diseño electrónico como puede ser el diseño de placas de circuitos impresos o PCB.

El software básicamente dispone de tres tipos de archivos:

- **Proyectos:** se utilizan para organizar los archivos de diseño. Un proyecto puede tener un número ilimitado de esquemáticos y un archivo de diseño de PCB. EAGLE es capaz de producir archivos de salida como GERBER y Excellon. Estos archivos estándar son aceptados por fabricantes de PCB y se utilizan para fabricar una tarjeta de circuito impreso.
- **Diseño de esquemáticos:** se utilizan para elaborar diagramas de circuitos y conexiones. En un proyecto, se combinan todos juntos para formar el diseño completo.
- **Diseño de PCB:** Los esquemáticos se traducen en un archivo de diseño de PCB con un Asistente de PCB. Un diseño de PCB necesitar varias iteraciones antes de pasar a una tarjeta de circuito impreso finalizada para la producción.

Si se detectase algún error de diseño o incumplimiento de cualquier especificación, se evaluará si es posible continuar con la siguiente etapa, en este caso, la etapa del desarrollo de firmware. Si por el contrario no fuese posible avanzar hasta la siguiente etapa de desarrollo, se realizarán las tareas de rediseño necesarias hasta que se obtenga un prototipo que cumpla con las especificaciones hardware fijadas.

Las sucesivas secciones muestran toda la información de diseño obtenida a lo largo del desarrollo de los prototipos hardware, incluyendo errores de diseño o funcionalidades que no se cumplen, paso en el que se detectó y soluciones propuestas para que sean subsanados.

6.1.1. PASO 1: ESPECIFICACIONES HARDWARE

Las especificaciones hardware consisten en una serie de restricciones que se usan para la toma de decisiones de diseño en las siguientes etapas. Estas decisiones de diseño se orientan, por un lado, en la elección de los componentes que integran la PBA (Printed Board Assembly) y, por otro lado, las simplificaciones que se pueden realizar, en función de la etapa de diseño de hardware en la que se encuentre.

Módulo principal

El módulo principal es un PCB que hace de conexión entre el EP2C5/EP2C8 MINI BOARD y los módulos secundarios. Contiene los conectores, el rutado de las líneas y conexiones con el módulo de comunicaciones Bluetooth y el Shield de Arduino.

Los conectores que comunican el módulo principal con los módulos secundarios son 4 conectores exactamente iguales. La Tabla 66 detalla la correspondencia entre las entradas/salidas de la tarjeta Arduino Nano y el conector P_MM:

Tabla 66. Relación entre las señales del módulo principal y el conector P_MM.

	Conector P_MM									
Pin Conector	2	4	6	8	10	12	14	16	18	20
Señal módulo principal	VIN	D0	D2	D4	D6	D8	D10	D12	RST	GND
Pin Conector	1	3	5	7	9	11	13	15	17	19
Señal módulo principal	GND	RST	D1	D3	D5	D7	D9	D11	D13	VIN

Módulo secundario

El módulo secundario es un PCB que hace de conexión entre el módulo principal, la Arduino Nano y los Shield de Arduino. Por lo tanto, el conector del módulo secundario es el descrito en la Tabla 66. También se pasan señales de reset del módulo principal a los módulos secundarios para poder hacer un reset a las Arduino Nano desde la FPGA.

En el módulo principal se pueden incluir hasta cuatro módulos secundarios, en función de las necesidades de uso relativas a sistemas robóticos que se quieran manejar, o grupos de alumnos que se creen para realizar diferentes actividades educativas. Por ejemplo, si se usa la plataforma para el control de un robot de ruedas con sensores de distancia, es posible emplear un módulo principal con dos módulos secundarios. Uno de los módulos secundarios se puede usar para la gestión de los motores del robot y el otro

módulo secundario se puede usar para adquirir las medidas de distancia a objeto adquiridas por los sensores. El segundo módulo secundario también se puede usar para hacer un procesamiento de señal, además de enviar información al otro módulo secundario sobre diferentes obstáculos encontrados.

Elementos Externos

La plataforma será compatible con tarjetas que pueden ser directamente conectadas al PCB de Arduino UNO. De esta manera, se consigue aumentar las capacidades funcionales de la plataforma de una manera económica y sencilla. Hoy en día existen multitud de Shields de Arduino con funcionalidades tales como: manejo de servos, memoria, manejo de motores, comunicaciones Bluetooth, comunicación Wi-Fi, pantalla LCD (Liquid Crystal Display), pantalla táctil TFT (Thin Film Transistor) y otras tantas.

Una gran ventaja de los Shield Arduino, además de poder ser apilados en una Arduino para añadirle funcionalidades, es que la mayoría de los Shield se pueden apilar unos encima de otros. La Figura 40 muestra un ejemplo de Shields Arduino apilados. En el ejemplo mostrado se incluye, de abajo a arriba, un Shield WiFi, dos Shield de control de motores de corriente continua y un Shield de pantalla táctil TFT.



Figura 40. Shields Arduino apilados.

Conexión con Smart Devices

La conexión entre la plataforma y el Smart Device se realizará por Bluetooth. Este tipo de comunicación lo incluye cualquiera de estos dispositivos. La plataforma será la encargada de incluir el módulo de comunicaciones Bluetooth. Para interactuar con la plataforma, es posible realizar el envío de comandos a través de una aplicación como puede ser un terminal de comunicaciones Bluetooth. Por otro lado, la plataforma puede enviar mensajes con información de estado al propio Smart Device. Un ejemplo de terminal Bluetooth es BlueTerm [BlueTerm, 2018].

Conexión a Internet

Basándose en la conexión mencionada en el anterior apartado, se añade conexión a Internet mediante una conexión puente Bluetooth, para poder comunicar con un dispositivo que tenga conexión a Internet mediante cable de red o mediante conexión WiFi. Ejemplos de estos dispositivos son la Raspberry Pi [Raspberry Pi, 2018] configurada como servidor u otros dispositivos similares, como pueden ser Beaglebone Black [Beagleboard, 2018] o cualquier PC configurado como servidor.

6.1.2. PASO 2: ELEMENTOS HARDWARE

Los elementos hardware son enumerados, añadiendo en cada uno de ellos sus características más importantes, según las necesidades que cubren en relación con las especificaciones fijadas para la PBA en cuestión, y una enumeración de los puertos de entrada/salida con su correspondiente descripción.

A) EP2C5/EP2C8 MINI BOARD

Se trata de un sistema básico de desarrollo FPGA basado en Altera Cyclone II EP2C5T144. Este sistema está compuesto por:

- 1) FPGA EP2C5T144 – ver [Altera, 2018].
- 2) Memoria EPROM de configuración EPCS4 con capacidad de 4Mbit.
- 3) Reloj SMD de 50 MHz.
- 4) Entrada externa de alimentación de 5 Vdc hasta 12 Vdc:
 - a) Conector Jack hembra, componente P7.
 - b) Pines de alimentación, componente P8.
- 5) LED SMD de indicación de alimentación, componente D3.
- 6) Botón de Reset, componente key.
- 7) 3 LED SMD de propósito general, componentes D2, D4 y D5.
- 8) Pines de entrada/salida, VCC, GND y 1.2 V. Componentes P1, P2, P3 y P4.

A continuación, se describirán en un conjunto de tablas con las siguientes características:

- Tabla 67 detalla la correspondencia entre las entradas/salidas (E/S) de la FPGA y el conector P1.
- Tabla 68 detalla la correspondencia entre las entradas/salidas de la FPGA y el conector P2.

- Tabla 69 detalla la correspondencia entre las entradas/salidas de la FPGA y el conector P3.
- Tabla 70 detalla la correspondencia entre las entradas/salidas de la FPGA y el conector P4.

Tabla 67. Relación entre las E/S de la FPGA y el conector P2.

		Conector P1													
Pin Conector		2	4	6	8	10	12	14	16	18	20	22	24	26	28
Pin FPGA		41	43	45	48	52	55	58	60	64	67	70	72	GND	VCC
Pin Conector		1	3	5	7	9	11	13	15	17	19	21	23	25	27
Pin FPGA		40	42	44	47	51	53	57	59	63	65	69	71	GND	VCC

Tabla 68. Relación entre las E/S de la FPGA y el conector P2.

		Conector P2													
Pin Conector		2	4	6	8	10	12	14	16	18	20	22	24	26	28
Pin FPGA		GND	74	76	80	86	88	90	92	94	97	100	103	GND	VCC
Pin Conector		1	3	5	7	9	11	13	15	17	19	21	23	25	27
Pin FPGA		GND	73	75	79	81	87	89	91	93	96	99	101	104	GND

Tabla 69. Relación entre las E/S de la FPGA y el conector P3.

		Conector P3													
Pin Conector		2	4	6	8	10	12	14	16	18	20	22	24	26	28
Pin FPGA		GND	113	115	119	121	125	129	133	135	137	141	143	GND	VCC
Pin Conector		1	3	5	7	9	11	13	15	17	19	21	23	25	27
Pin FPGA		GND	112	114	118	120	122	126	132	134	136	139	142	GND	GND

Tabla 70. Relación entre las E/S de la FPGA y el conector P4.

		Conector P4													
Pin Conector		2	4	6	8	10	12	14	16	18	20	22	24	26	28
Pin FPGA		GND	4	8	GND	GND	18	21	24	26	1.2	28	31	GND	VCC
Pin Conector		1	3	5	7	9	11	13	15	17	19	21	23	25	27
Pin FPGA		GND	3	7	9	GND	17	22	25	27	1.2	30	32	GND	VCC

B) Shield de Arduino

La plataforma será compatible con tarjetas que pueden ser directamente conectadas al PCB de Arduino UNO. De esta manera, se consigue aumentar las capacidades funcionales de la plataforma de un modo económico y siendo sencillo su montaje. Hoy en día existen multitud de Shields de Arduino con funcionalidades tales como: manejo de servos, memoria, manejo de motores, comunicaciones Bluetooth, comunicación Wi-Fi, pantalla LCD, pantalla táctil TFT y otras tantas.

Los Shields de Arduino se caracterizan por tener 4 puertos de comunicaciones:

- 1) Conector de puertos digitales 1.

- 2) Conector de puertos digitales 2.
- 3) Conector de puertos analógicos.
- 4) Conector de alimentación.

A continuación, se describirán en un conjunto de tablas las siguientes características:

- Tabla 71 detalla la correspondencia entre las entradas/salidas de Arduino y el conector de puertos digitales 1.
- Tabla 72 detalla la correspondencia entre las entradas/salidas de Arduino y el conector de puertos digitales 2.
- Tabla 73 detalla la correspondencia entre las entradas/salidas de Arduino y el conector de puertos analógicos.
- Tabla 74 detalla la correspondencia entre las entradas/salidas de Arduino y el conector de alimentación.

Tabla 71. Relación entre las E/S de Arduino y el conector de puertos digitales 1.

	Conector de puertos digitales 1							
Pin Conector	1	2	3	4	5	6	7	8
Puerto Arduino	D0	D1	D2	D3	D4	D5	D6	D7

Tabla 72. Relación entre las E/S de Arduino y el conector de puertos digitales 2.

	Conector de puertos digitales 2									
Pin Conector	1	2	3	4	5	6	7	8	9	10
Puerto Arduino	D8	D9	D10	D11	D12	D13	GND	AREF	AD4	AD5

Tabla 73. Relación entre las E/S de Arduino y el conector de puertos analógicos.

	Conector de puertos analógicos					
Pin Conector	1	2	3	4	5	6
Puerto Arduino	AD0	AD1	AD2	AD3	AD4	AD5

Tabla 74. Relación entre las E/S de Arduino y el conector de alimentación.

	Conector de alimentación							
Pin Conector	1	2	3	4	5	6	7	8
Puerto Arduino	NC	VCC	RST	3V3	VCC	GND	GND	VIN

C) Módulo de comunicación Bluetooth

El módulo de comunicación Bluetooth es una sencilla placa que se comunica a través de sus puertos de entrada/salida con un protocolo de comunicación serie basado en el estándar RS-232 usando dos puertos, uno de recepción y otro de transmisión. Se alimenta

con 3,3 Vdc e implementa comunicaciones inalámbricas usando el protocolo de comunicaciones Bluetooth.

Este módulo de comunicaciones se conecta directamente a la FPGA y su alimentación proviene de la placa que alberga a la FPGA. La Tabla 75 detalla la correspondencia entre las entradas/salidas de la FPGA, del Módulo de Comunicaciones Bluetooth y el conector BT_COM.

Tabla 75. Relación entre las E/S de Arduino y el conector BT_COM.

Pin Conector	Conector BT_COM			
	1	2	3	4
Puerto Módulo Bluetooth	3V3	GND	TXD	RXD
Pin FPGA	VCC	GND	104	102

D) Arduino Nano

La Arduino Nano es una placa pequeña, completa y fácil de usar basada en la ATmega328. Tiene más o menos la misma funcionalidad que el Arduino Duemilanove, pero en un paquete diferente. Sólo carece de un conector de alimentación de CC y funciona con un cable USB Mini-B en lugar de uno estándar. Se trata de una tarjeta compacta (ver [Arduino, 2018i]) que cuenta con las siguientes características:

- 1) Tensión de alimentación de 5 Vdc desde USB o un puerto de alimentación.
- 2) Entrada de reset mediante pulsador o dos puertos con esta funcionalidad.
- 3) Un LED de estado.
- 4) 14 puertos digitales, de los cuales 6 pueden ser usados para generar PWM. Son los puertos 3, 5, 6, 9, 10 y 11.
- 5) 8 entradas analógicas, cada una proporciona una resolución de 10 bits (1024 valores). Se puede configurar la tensión de referencia de las entradas analógicas.
- 6) Canal de comunicaciones serie, usado para la programación del microcontrolador y para comunicarse con otros dispositivos. El puerto 0 es el de recepción y el puerto 1 es el de transmisión.
- 7) 2 interrupciones externas.
- 8) Comunicaciones SPI e I2C.

6.1.3. PASO 3: DISEÑO DE LOS ESQUEMÁTICOS

Los diseños de esquemáticos deben contener todos los componentes que forman la PBA. Además, incluyen las conexiones entre los diferentes componentes. Por claridad del esquemático y simplicidad se opta por usar nombres para las líneas de conexión, gracias a esta medida se pueden representar todos los componentes sin la necesidad de

unir directamente los puertos de entrada/salida de todos y cada uno de los componentes. En este paso se fijan los nombres de cada una de las líneas de conexión, y para ello, se siguen las siguientes normas de nomenclatura:

- Si existe una conexión directa con otra PBA, y ya existe un nombre en dicha PBA, se utilizará el mismo nombre, siempre que EAGLE lo permita. Es decir, no se usen caracteres especiales que no acepta EAGLE.
- Si se trata de líneas que conectan puertos de PBA distintas y el nombre en ambas ya está fijado, se elegirá un nombre que caracterice la funcionalidad que desempeña.

6.1.4. PASO 4: DISEÑO FÍSICO DEL PCB

El diseño se realizará en dos capas para la tarjeta, este criterio de diseño se sustenta en las características de la tarjeta a desarrollar:

- No muy altas frecuencias en las señales transmitidas por las pistas.
- Pistas que conducirán corrientes no muy elevadas.
- Densidad de componentes baja.

En el proceso de diseño de tarjetas de desarrollo se cuenta con un gran número de opciones para realizar la etapa de procesamiento de todo el sistema. Empresas como Microchip con su línea 32M [Microchip, 2018], ST Microelectronics [ST, 2018] o Texas Instruments [Texas, 2018] cuentan cada uno con suites de desarrollo y MCU (Microcontroller Unit) con grandes capacidades de desempeño. Una de las tendencias más grandes en el diseño electrónico actual es el de usar arquitecturas de ARM (Advanced Risc Machine) [ARM, 2018] entonces se delimitan las opciones a lo que ofrecen Microchip, ST Microelectronics y Texas Instruments.

6.1.5. PASO 5: ARCHIVOS DE FABRICACIÓN

Una vez completado el diseño físico del PCB, es el momento de preparar los archivos de fabricación. La herramienta EAGLE no genera automáticamente los GERBER, por lo que se deben seguir los pasos indicados en esta sección.

Además de EAGLE, la herramienta Gerbv [Gerbv, 2018] será de gran utilidad. Se trata de una herramienta de código abierto y multiplataforma para realizar la revisión de los archivos de fabricación GERBER que se generen.

La herramienta EAGLE cuenta con una funcionalidad de DRC (Design Rule Check) mediante la que se comprueba que el diseño no excede las capacidades de producción

DFRobot [DFRobot, 2018] proporciona un archivo DRU para poder ejecutar el DRC. Basándose en los resultados de la comprobación se podrán realizar las modificaciones oportunas al diseño, de manera que se puedan corregir problemas comunes. En el caso de especificaciones particulares, ciertos errores del informe pueden ser ignorados.

Para realizar la comprobación automática de las reglas de diseño, en primer lugar, se inicia el software de diseño de PCB, EAGLE, y se abre el archivo de diseño con extensión brd. Posteriormente se selecciona “Tools” y “DRC...”. Se presiona “load” para abrir el archivo DRU de DFRobot. Una vez cargado el archivo, se presiona “check” para obtener el informe de errores.

Una vez realizadas las pertinentes comprobaciones, se procede a generar los archivos fabricación con la herramienta EAGLE. Los archivos GERBER son típicamente producidos por diseñadores de PCB usando software de PCB general. Estos archivos se mandan a fabricantes de PCB, y son cargados en un sistema CAM (Computer-Aided Manufacturing) usando CAM350 para preparar los datos para cada paso del proceso de fabricación de PCB. En este flujo de trabajo se transfiere la información de las capas desde el archivo PCB/brd al archivo CAM.

1. Primeramente, en la ventana Eagle Board, se selecciona “File” y “CAM Processor”.
2. Seleccionar “File”, “Open” y “Job”.
3. Seleccionar “Gerb274x.cam” y pulsar “open”.
4. Ahora es importante asegurarse que la opción “Mirror” no está marcada en ninguna de las pestañas. Entonces pulsar “Process Job...”
5. En unos segundos se generarán todos los archivos, una vez haya finalizado el proceso, se cierra la ventana de diálogo.

Nota: El trabajo (gerb274x) del procesador sólo exportará las capas Dimension, tPlace y tNames. Por lo tanto, no exportará la capa inferior de serigrafía. Si fuese necesario imprimir la capa tValues y la cara inferior de serigrafía, se podría modificar el trabajo gerb274x y guardar el trabajo modificado.

Por otro lado será necesario generar los archivos de taladrado Excellon. El formato Excellon es usado para intercambiar datos de taladros y rutas entre los sistemas CAD/CAM, además del control de las máquinas CNC. Los pasos para generar estos archivos son los siguientes:

1. Abrir el procesador CAM de nuevo y pulsar “File”, “Open” y “Job”
2. Seleccionar excellon.cam, y pulsar “open”

3. Pulsar “process Job”. Entonces se generarán instantáneamente los archivos de taladrado

Después de los pasos arriba indicados, los archivos de producción están listos para ser enviados a fabricación. Los archivos son los siguientes:

- drd: Excellon drill description
- dri: Excellon drill tool description
- cmp: Component side data
- sol: Solder side data
- plc: Component side silk screen data
- stc: Component side solder stop mask data
- sts: Solder side solder stop mask data
- gpi: Gerber photoplotter information data

Para comprobar los archivos GERBER generados se pueden importar a la herramienta GerbV.

Ya solo queda añadir todos los archivos de fabricación generados a un archivo zip y enviárselos al fabricante.

6.1.6. PASO 6: ENSAMBLADO DE LA PBA

El proceso de ensamblado se realiza mediante el montaje de los componentes y la soldadura de estos al PCB. Durante este proceso se sigue el siguiente orden: primero se ensamblarán los componentes de menor tamaño para ensamblar en último lugar los componentes que tengan un mayor tamaño.

Los conectores que se montan son de tipo agujero pasante. Para su montaje se utiliza un soldador de estaño como el que la Figura 41 muestra. Esta herramienta tiene las siguientes características principales:

- Permite realizar todo tipo de trabajo de soldadura con estaño de componentes de agujero pasante y algunos componentes SMD (Surface Mount Devices)
- Su consumo de potencia eléctrica es de 25 vatios
- La temperatura máxima soportada es de 380 ° C
- La fuente de alimentación es de 230 V_{AC}



Figura 41. Soldador de estaño.

6.1.7. PASO 7: PRUEBAS DE INTEGRIDAD

Las pruebas de integridad son pruebas básicas que tratan de detectar la ausencia de errores de diseño y fabricación tales como: pistas no conectadas entre dos puntos, pistas o contactos con una conexión no deseada entre ambos y otros tipos de fallos similares a estos dos.

Las pruebas de integridad que se han de pasar son las siguientes:

1. Prueba de continuidad: se mide continuidad o resistencia eléctrica entre puntos de la tarjeta que estén unidos por una pista. También se mide la no continuidad entre puntos del PCB contiguos.
2. Prueba de alimentación: se aplica alimentación y se mide tensión en los puntos donde debe haber tensión de alimentación.

6.2. Primer prototipo, versión 1.0

Se trata de la primera versión de los PCB del módulo principal y del módulo secundario. Para ser colocado encima del sistema básico de desarrollo basado en la FPGA (Field Programmable Gate Array), se ha elaborado un PCB (Printed Circuit Board) que conecte dicha placa con los módulos secundarios, con un módulo de comunicaciones Bluetooth, además de la alimentación que se distribuirá a los módulos secundarios. También incluye una conexión compatible con Shields de Arduino. Durante el desarrollo de la primera versión se han aplicado todos los pasos descritos en la metodología expuesta en la sección anterior.

Para la elaboración de este primer prototipo se ha usado la información indicada en las lecciones 5.1.1 y 5.1.2 de este capítulo 5. Como resultado de la ejecución del paso 3, diseño de los esquemáticos, se han obtenido los esquemáticos para la versión 1.0. La Figura 42 y la Figura 43 muestran los esquemáticos de esta primera versión del módulo

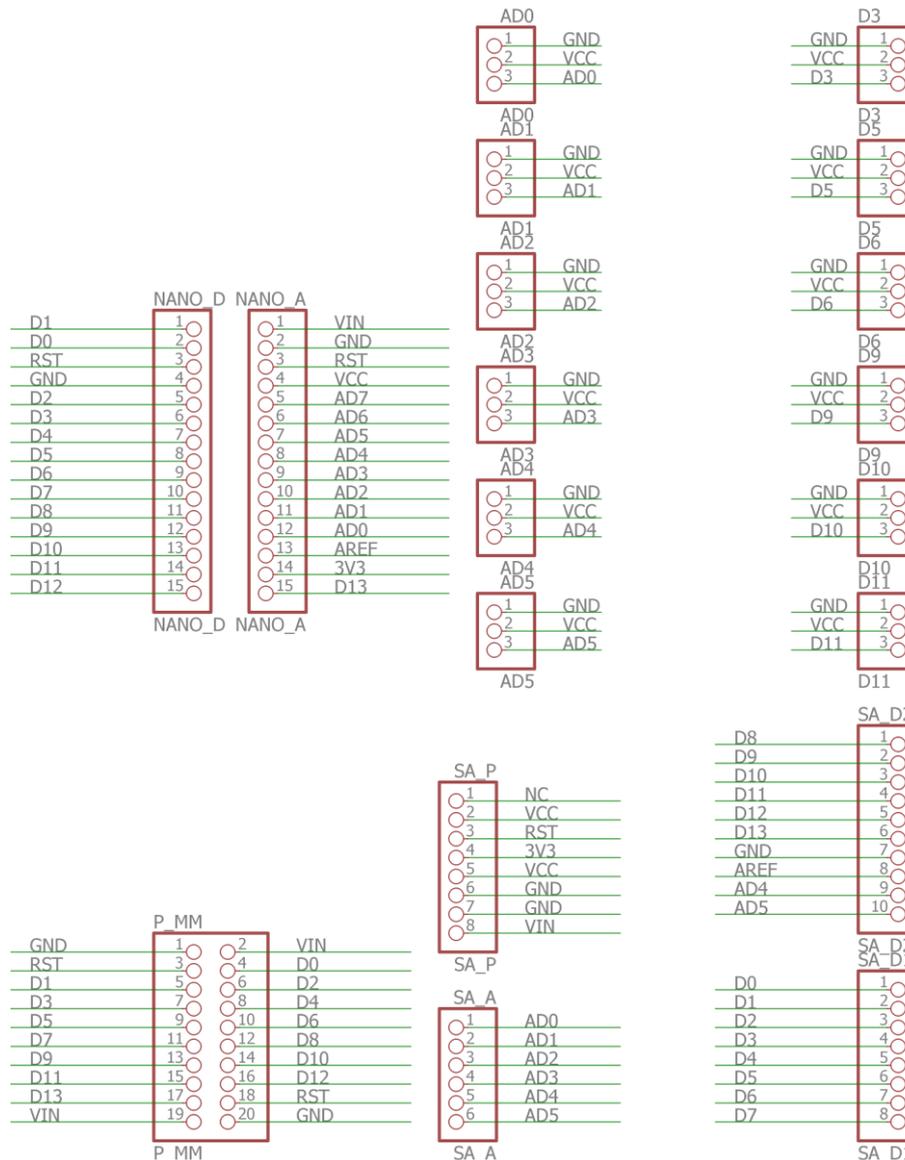


Figura 43. Esquemáticos de la versión 1.0 del módulo secundario.

Una vez obtenidos los esquemáticos para esta versión se procede al siguiente paso 4, diseño físico del PCB. En este caso, se han colocado todos los elementos de la tarjeta en la localización adecuada y se han posicionado todas las conexiones entre los diferentes elementos de manera que no quede ninguna conexión sin trazar. El resultado de este cuarto paso se muestra en la Figura 44 y en la Figura 45 para el módulo principal y para el módulo secundario.

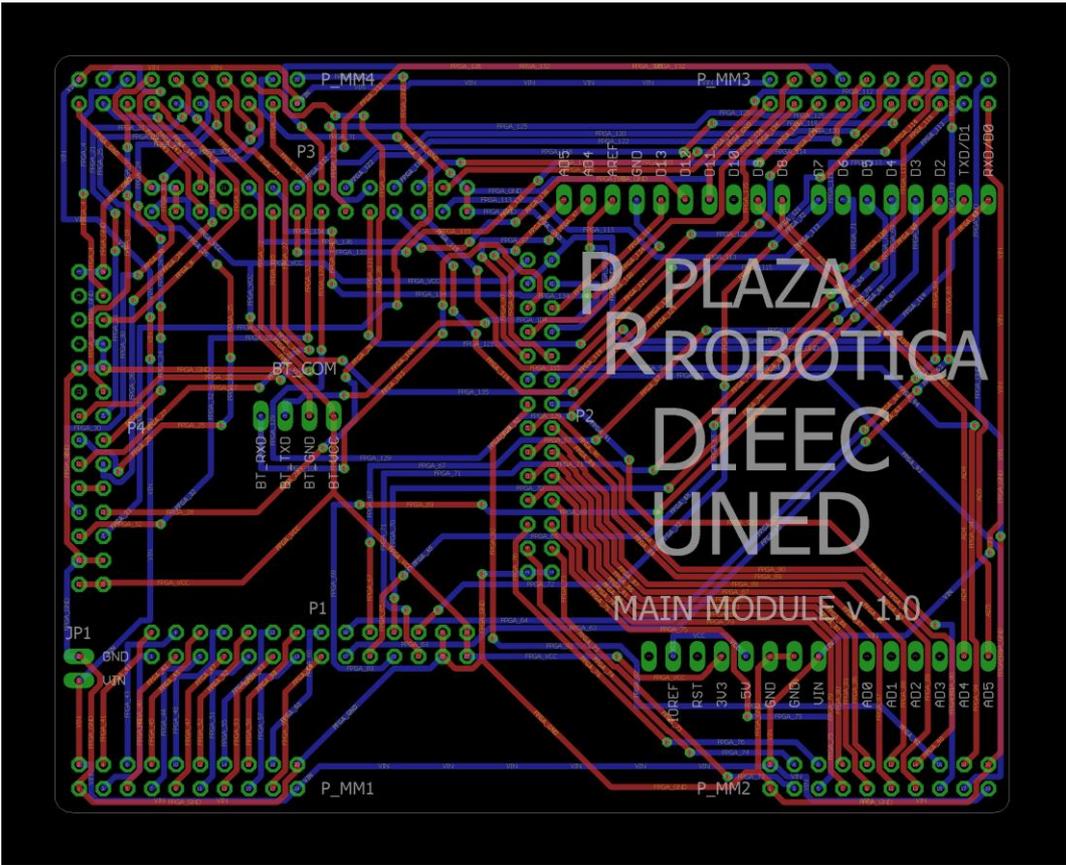


Figura 44. Layout de la versión 1.0 del módulo principal.

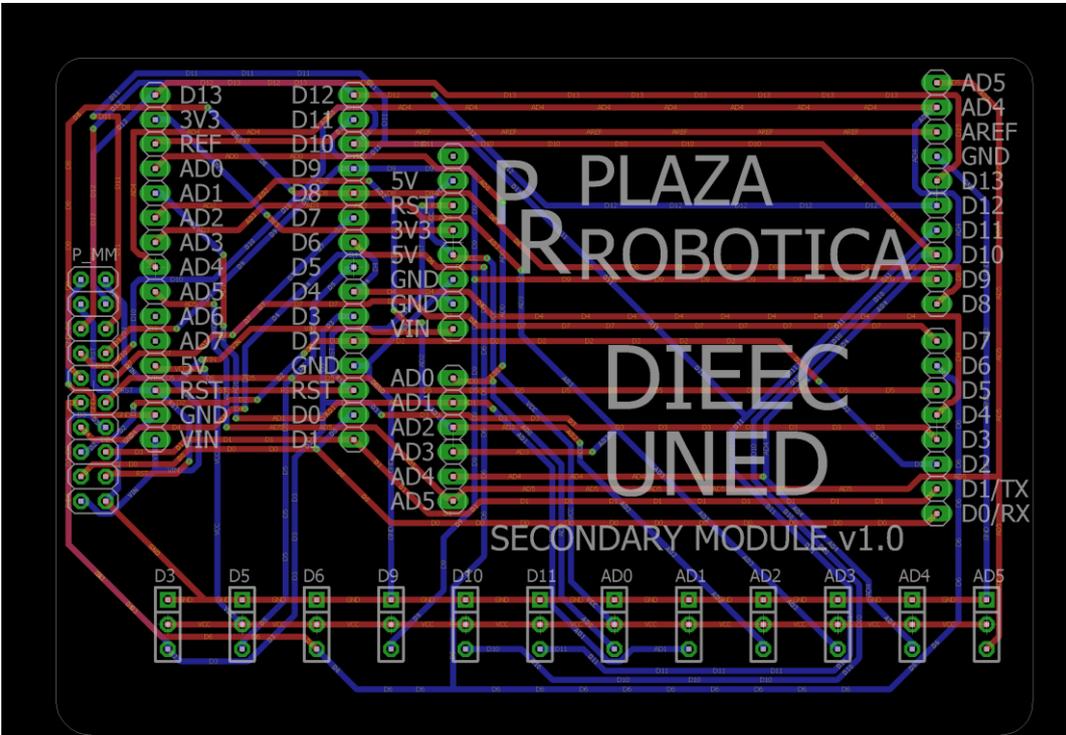


Figura 45. Layout de la versión 1.0 del módulo secundario.

A lo largo del paso 5 se generan los archivos de fabricación. Para la primera versión del módulo principal se han obtenido los siguientes archivos:

- MAIN_MODULE.drd
- MAIN_MODULE.dri
- MAIN_MODULE.cmp
- MAIN_MODULE.sol
- MAIN_MODULE.plc
- MAIN_MODULE.stc
- MAIN_MODULE.sts
- MAIN_MODULE.gpi

Para la primera versión del módulo secundario se han obtenido los siguientes archivos:

- SECONDARY_MODULE.drd
- SECONDARY_MODULE.dri
- SECONDARY_MODULE.cmp
- SECONDARY_MODULE.sol
- SECONDARY_MODULE.plc
- SECONDARY_MODULE.stc
- SECONDARY_MODULE.sts
- SECONDARY_MODULE.gpi

Los archivos de fabricación arriba indicados se mandaron al fabricante. El fabricante tarda en torno a una semana en realizar una revisión de los archivos antes de fabricar los PCB y en la propia fabricación. Una vez fabricadas las tarjetas, el fabricante realiza el envío. La recepción de las tarjetas desde que las manda el fabricante suele tardar una semana.

La Figura 46 muestra la versión 1.0 del PCB del módulo principal. En la parte superior de la figura se puede ver la cara superior de la tarjeta y en la parte inferior de la figura se puede ver la cara inferior de la tarjeta.

Una vez recibidas las tarjetas se comienza con el paso 6, ensamblado de las PBA. La Figura 48 muestra la versión 1.0 del módulo principal con todos los conectores ya montados. La PBA cuenta con conectores inferiores para comunicar la FPGA con los módulos secundarios. En la parte superior se pueden observar puertos para conectar hasta cuatro módulos secundarios, conexión para Shields de Arduino, y, por último, un conector para poder añadir un módulo Bluetooth.



Figura 48. Versión 1.0 del módulo principal.

Las señales que se intercambian entre el módulo principal, la FPGA, y los módulos secundarios, las Arduino Nano, son los puertos de entradas/salidas digitales de las Arduino Nano.

La Figura 49 muestra la versión 1.0 del módulo secundario con todos los conectores ya montados. La PBA cuenta con conectores en la parte superior para conectar el módulo secundario con el módulo principal, conexión para Shields de Arduino, conectores para entradas/salidas digitales y señales PWM (Pulse Width Modulation) y, por último, conectores para entradas/salidas analógicas que pueden ser usadas para incluir sensores.

- Modificación del tamaño de la serigrafía. Se incrementa su tamaño para que se vea mejor.
- Se agrupan los seis conectores de puertos digitales para facilitar su montaje.
- Se agrupan los seis conectores de puertos analógicos para facilitar su montaje.

Por último, se propone modificar el tamaño del módulo secundario para que la siguiente versión de este PCB sea compatible con la versión 1.1 del módulo principal.

6.3. Segundo prototipo, versión 1.1

Esta versión contiene los cambios mencionados en la sección anterior. Para la elaboración de este segundo prototipo no se han modificado los archivos generados durante la realización del primer prototipo hasta este paso, es decir, no se han modificado los esquemáticos, ni del módulo principal, ni del módulo secundario.

De acuerdo con la metodología descrita en el capítulo 5, los cambios aplicados para la segunda versión del prototipo, versión 1.1 son motivados por las pruebas de integración de hardware. La Figura 50 muestra la transición que proviene del desarrollo de hardware y afecta al desarrollo de hardware.

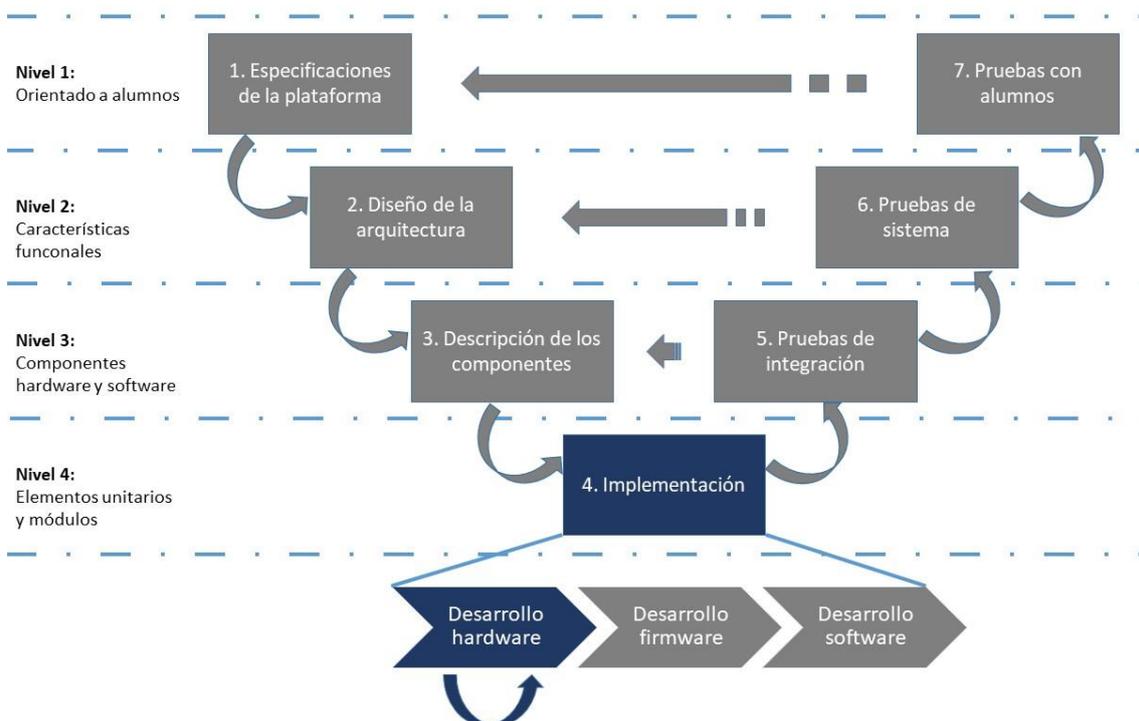


Figura 50. Diagrama de bloques del primer cambio hardware.

Para aplicar los cambios que contiene esta segunda versión de tarjetas, se procede al paso 4, diseño físico del PCB. En este caso, se han recolocado todos los elementos de la tarjeta en la localización correcta y se han posicionado todas las conexiones entre los

diferentes elementos de manera que no quede ninguna conexión sin trazar. El resultado de este cuarto paso lo muestra la Figura 51 y la Figura 52 para el módulo principal y para el módulo secundario.

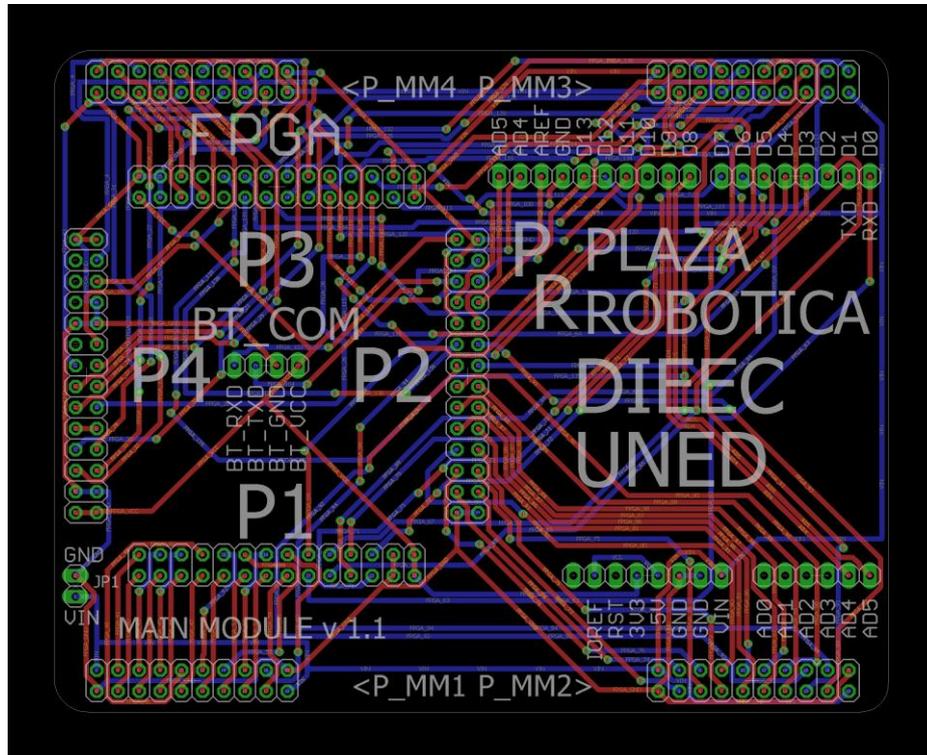


Figura 51. Layout de la versión 1.1 del módulo principal.

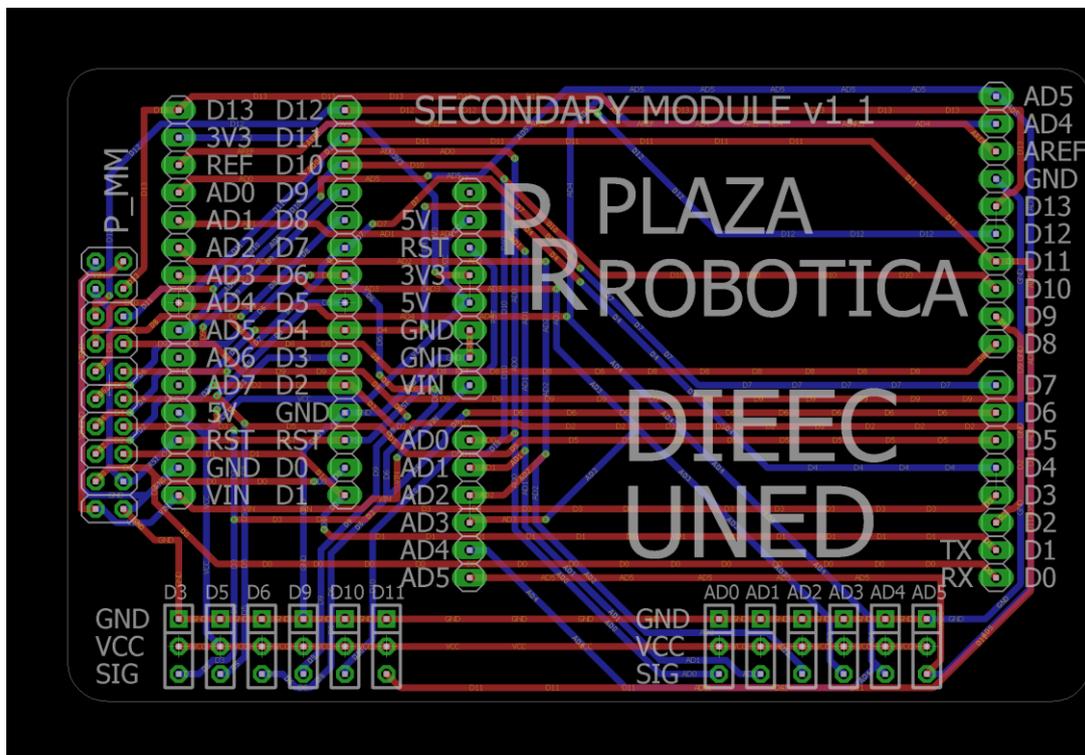


Figura 52. Layout de la versión 1.1 del módulo secundario.

A lo largo del paso 5 se generan los archivos de fabricación. Para la segunda versión del módulo principal se han obtenido los siguientes archivos:

- MAIN_MODULE.drd
- MAIN_MODULE.dri
- MAIN_MODULE.cmp
- MAIN_MODULE.sol
- MAIN_MODULE.plc
- MAIN_MODULE.stc
- MAIN_MODULE.sts
- MAIN_MODULE.gpi

Para la segunda versión del módulo secundario se han obtenido los siguientes archivos:

- SECONDARY_MODULE.drd
- SECONDARY_MODULE.dri
- SECONDARY_MODULE.cmp
- SECONDARY_MODULE.sol
- SECONDARY_MODULE.plc
- SECONDARY_MODULE.stc
- SECONDARY_MODULE.sts
- SECONDARY_MODULE.gpi

Por problemas con la máquina de fabricación de PCB se retrasa la fecha de entrega. El fabricante indica que la máquina no estará operativa hasta el 14 de febrero de 2016, por lo que se estima que se reciba la nueva versión de los PCB la semana comprendida entre el 22 de febrero de 2016 y el 28 de febrero de 2016.

La Figura 53 muestra la versión 1.1 del PCB del módulo principal. En la parte superior de la figura se puede ver la cara superior de la tarjeta y en la parte inferior de la figura se puede ver la cara inferior de la tarjeta.

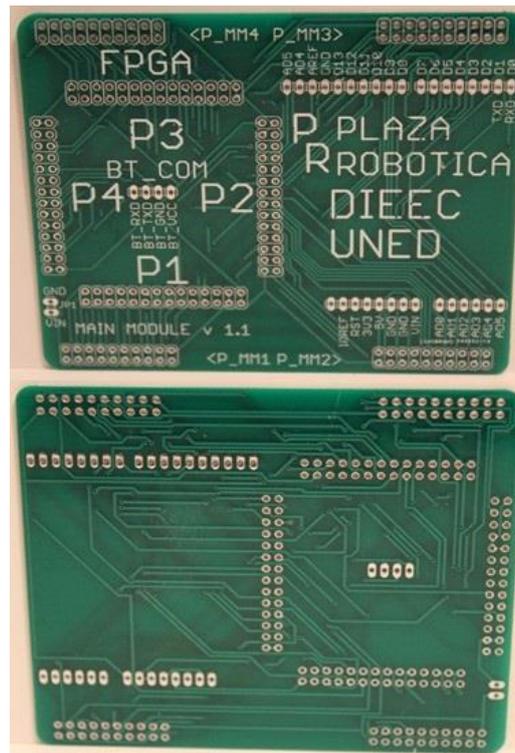


Figura 53. Versión 1.1 del PCB del módulo principal.

La Figura 54 muestra una comparativa de la cara superior de la versión 1.0 del PCB del módulo principal y de la nueva versión 1.1.

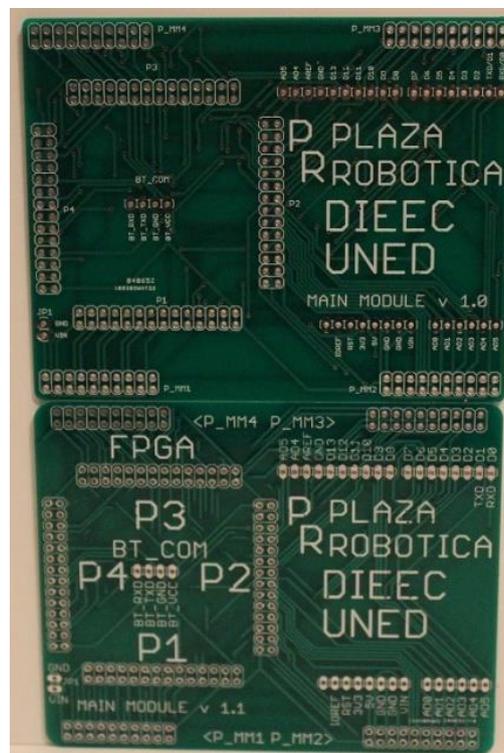


Figura 54. Versiones 1.0 (arriba) y 1.1 (abajo) del PCB del módulo principal.

La Figura 55 muestra la versión 1.1 del PCB del módulo secundario. En la parte inferior de la figura se puede ver la cara inferior de la tarjeta.

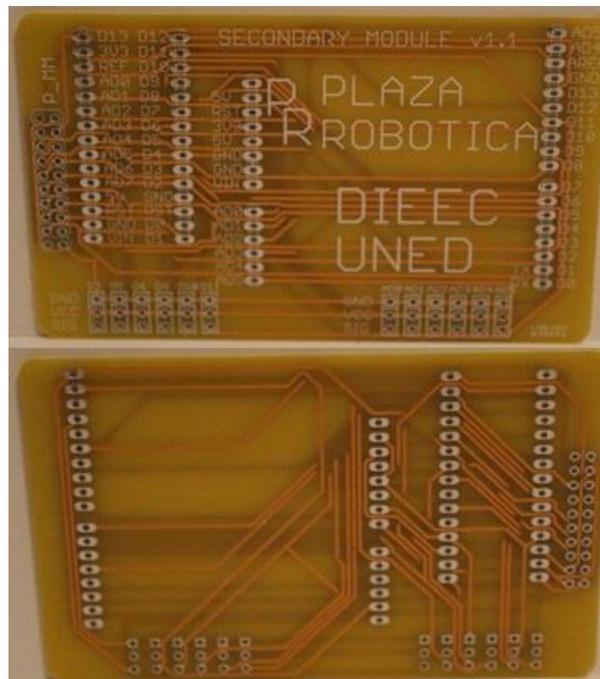


Figura 55. Versión 1.1 del PCB del módulo secundario.

La Figura 56 muestra una comparativa de la cara superior de la versión 1.0 del PCB del módulo secundario y de la nueva versión 1.1.



Figura 56. Versiones 1.0 (arriba) y 1.1 (abajo) del PCB del módulo secundario.

Una vez recibidas las tarjetas se comienza con el paso 6, ensamblado de las PBA. La Figura 57 muestra la versión 1.1 del módulo principal con todos los conectores ya montados. En la parte inferior de la figura se puede ver la cara inferior de la tarjeta.



Figura 57. Versión 1.1 del módulo principal.

La Figura 58 muestra una comparativa de la cara superior de la versión 1.0 de la PBA del módulo principal y de la nueva versión 1.1.



Figura 58. Versiones 1.0 (arriba) y 1.1 (abajo) de la PBA del módulo principal.

La Figura 59 muestra la versión 1.1 del módulo secundario con todos los conectores ya montados.

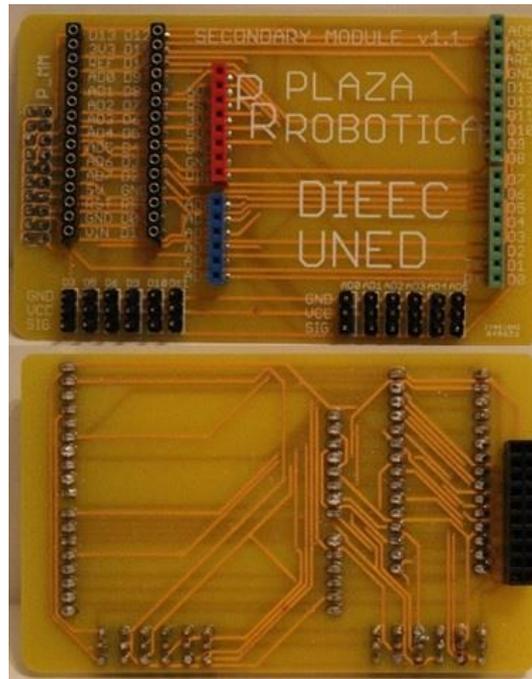


Figura 59. Versión 1.1 del módulo secundario.

La Figura 60 muestra una comparativa de la cara superior de la versión 1.1 de la PBA del módulo secundario y de la nueva versión 1.0.



Figura 60. Versiones 1.1 (arriba) y 1.0 (abajo) de la PBA del módulo secundario.

Durante el montaje de los ensamblajes, tanto del módulo principal, como del módulo secundario, se detectan mejoras de diseño que pueden ser subsanados en una versión posterior.

El diseño del PCB del módulo principal se puede mejorar realizando las siguientes modificaciones:

- Los conectores P_MM aún sobresalen un poco del PCB una vez montados. Con este cambio no se requiere reducir el ancho del PCB del módulo secundario.

El diseño del PCB del módulo secundario se puede mejorar para facilitar su montaje de la siguiente manera:

- Acercar más los seis conectores de puertos digitales para facilitar su montaje.
- Acercar más los seis conectores de puertos analógicos para facilitar su montaje.

Esta versión ya permite montar todos los elementos. La Figura 61 muestra la versión 1.1 del montaje completo que incluye:

- Sistema básico de desarrollo FPGA basado en Altera Cyclone II EP2C5T144
- Módulo principal versión 1.1
- Módulo Bluetooth
- Shield Wifi
- Shield TFT LCD Táctil
- 4 módulos secundarios versión 1.1
- 4 Arduino Nano
- 4 Shield de prototipado



Figura 61. Versión 1.1 de la Plataforma.

Tras obtener la versión 1.1 del módulo principal y del módulo secundario se procede a realizar las pruebas de integridad del hardware descritas en la sección 5.1.7:

1. Se comprueba la continuidad o resistencia eléctrica entre puntos de la tarjeta que estén unidos por una pista. También se mide la no continuidad entre puntos del PCB contiguos.
2. En un segundo paso se aplica alimentación y se mide tensión en los puntos donde debe haber tensión de alimentación.

Una vez concluidas las pruebas se comprueba que la versión 1.1 del módulo principal y del módulo secundario está lista para ser usada en las pruebas de integración hardware-firmware – ver sección 6.2.

6.4. Tercer prototipo, versión 1.2

Esta versión contiene los cambios mencionados en la sección anterior, además de las correcciones propuestas como resultado de la integración hardware-firmware detectados en la versión 1 del firmware – sección 6.3. De acuerdo con la metodología descrita en el capítulo 5, los cambios aplicados para la tercera versión del prototipo, versión 1.2 son motivados por las pruebas de integración de hardware-firmware. La Figura 62 muestra la transición que proviene del desarrollo de hardware y afecta al desarrollo de hardware.

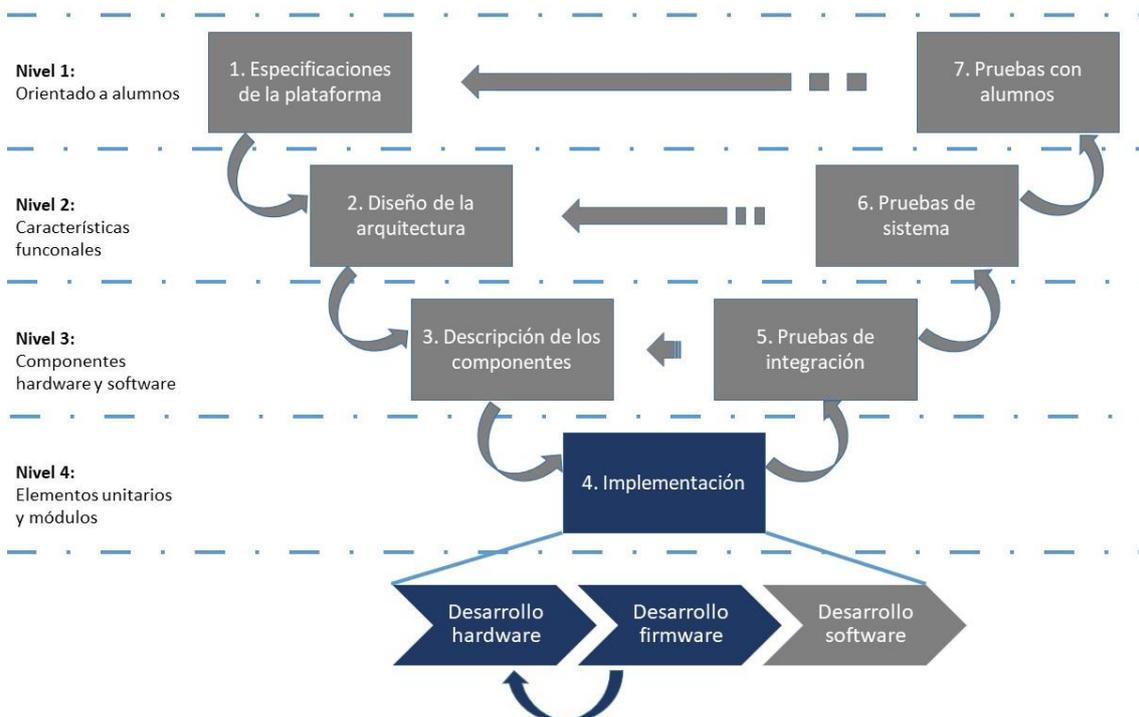


Figura 62. Diagrama de bloques del segundo cambio hardware.

Los cambios son los siguientes:

- El diseño del PCB del módulo principal se mejora realizando las siguientes modificaciones:
 - Los conectores P_MM se recolocan a una posición más interna del PCB.
 - El PIN_144 de la FPGA se conecta los pines de reset de los conectores de los módulos secundarios.
 - Se reorganizan las asignaciones de pines de la FPGA con los conectores del módulo principal.
- El diseño del PCB del módulo secundario se puede mejorar para facilitar su montaje de la siguiente manera:
 - Se acercan más los seis conectores de puertos digitales para facilitar su montaje.
 - Se acercan más los seis conectores de puertos analógicos para facilitar su montaje.

La reorganización de las asignaciones de pines de la FPGA con los conectores del módulo principal queda de la siguiente manera:

- Se sustituye el PIN_76 por el PIN_40.
- El PIN_76 deberá ser clasificado como No Usado.
- Se sustituye el PIN_80 por el PIN_88.
- Se sustituye el PIN_81 por el PIN_94.
- Se sustituye el PIN_88 por el PIN_73.

Para aplicar los cambios que contiene esta segunda versión de tarjetas, se realizan las modificaciones pertinentes utilizando la metodología de desarrollo hardware a partir del paso 3, diseño de los esquemáticos. En este caso, sólo ha sido necesario modificar los esquemáticos del módulo principal, por lo tanto, los esquemáticos del módulo secundario se mantienen tal cual se obtuvieron para la primera versión de hardware. Como resultado de la ejecución de este paso, se han obtenido los esquemáticos para la versión 1.2 del módulo principal. La Figura 63 muestra los esquemáticos de esta primera versión del módulo principal y del módulo secundario respectivamente. Como se puede ver en dichas figuras, quedan representados los conectores para cada uno de los elementos de dichas tarjetas y las conexiones que hay entre los diferentes conectores.

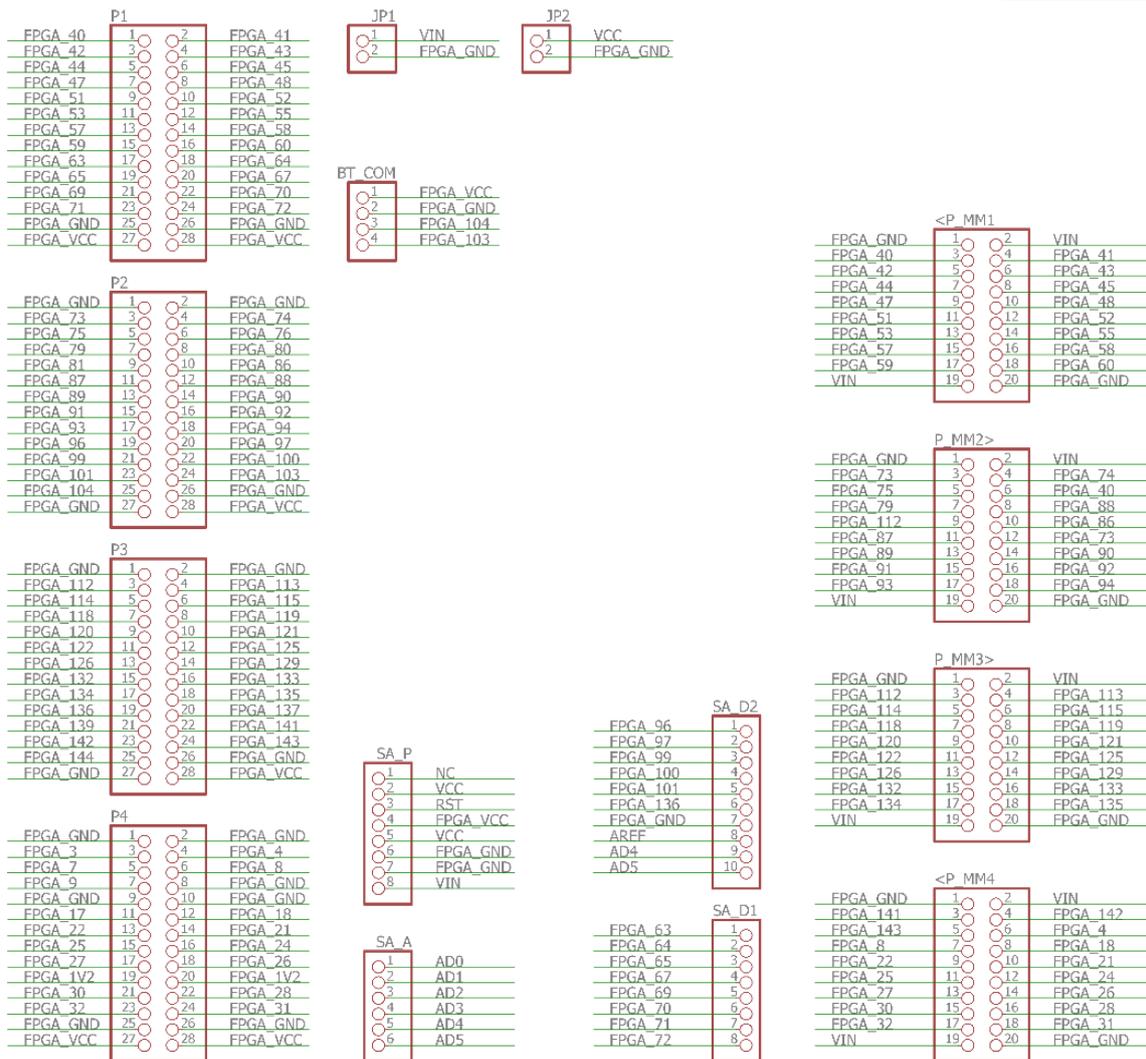


Figura 63. Esquemáticos de la versión 1.2 del módulo principal.

Posteriormente se procede al paso 4, diseño físico del PCB. En este caso, se han relocalizado todos los elementos de la tarjeta en la localización correcta y se han posicionado todas las conexiones entre los diferentes elementos de manera que no quede ninguna conexión sin trazar. El resultado de este cuarto paso lo muestra la Figura 64 y la Figura 65 para el módulo principal y para el módulo secundario.

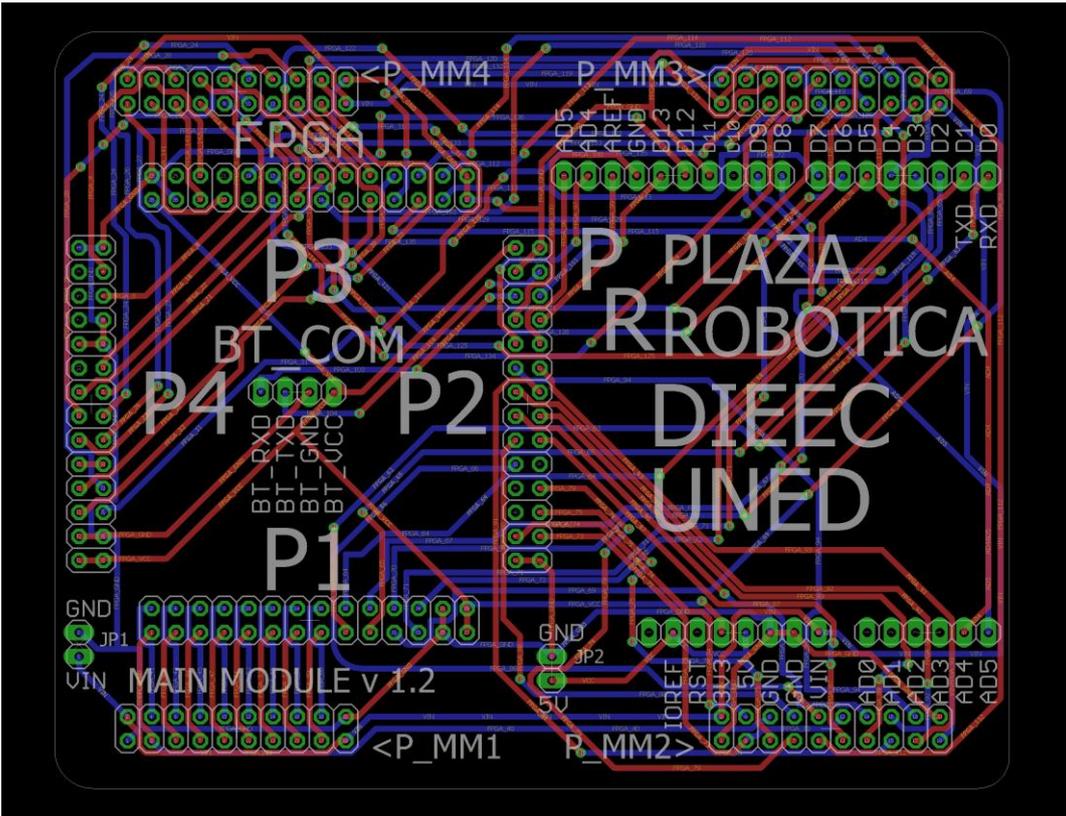


Figura 64. Layout de la versión 1.2 del módulo principal.

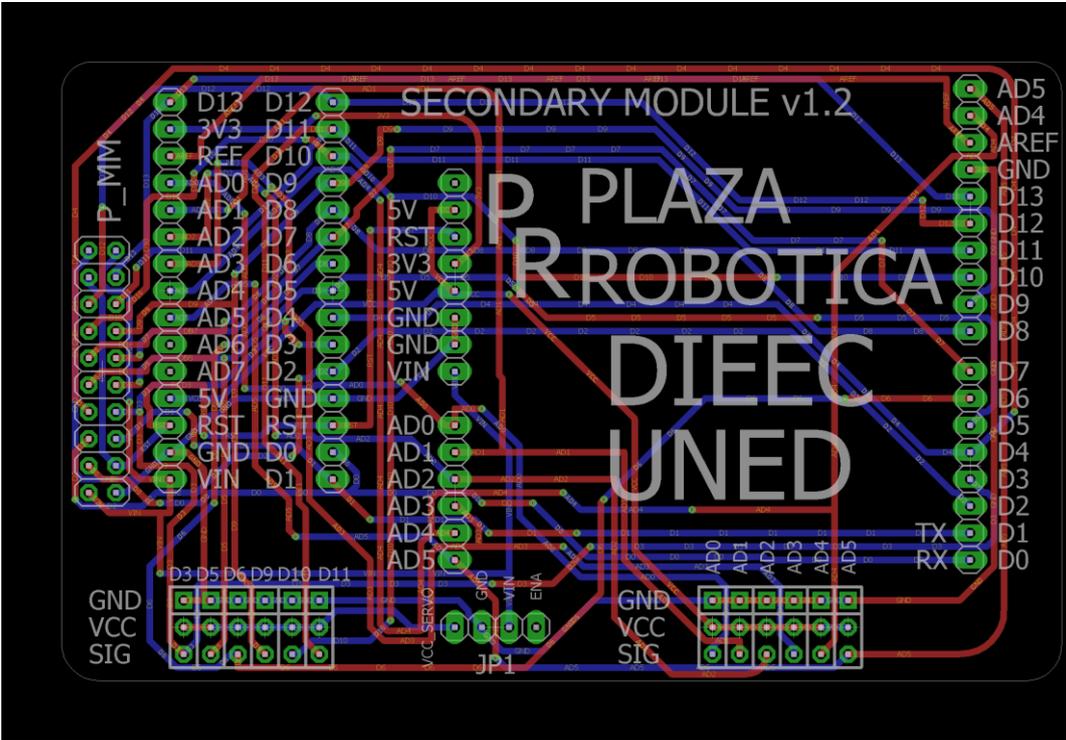


Figura 65. Layout de la versión 1.2 del módulo secundario.

A lo largo del paso 5 se generan los archivos de fabricación. Para la tercera versión del módulo principal se han obtenido los siguientes archivos:

- MAIN_MODULE.drd
- MAIN_MODULE.dri
- MAIN_MODULE.cmp
- MAIN_MODULE.sol
- MAIN_MODULE.plc
- MAIN_MODULE.stc
- MAIN_MODULE.sts
- MAIN_MODULE.gpi

Para la tercera versión del módulo secundario se han obtenido los siguientes archivos:

- SECONDARY_MODULE.drd
- SECONDARY_MODULE.dri
- SECONDARY_MODULE.cmp
- SECONDARY_MODULE.sol
- SECONDARY_MODULE.plc
- SECONDARY_MODULE.stc
- SECONDARY_MODULE.sts
- SECONDARY_MODULE.gpi

La Figura 66 muestra la versión 1.2 del PCB del módulo principal. En la parte superior de la figura se puede ver la cara superior de la tarjeta y en la parte inferior de la figura se puede ver la cara inferior de la tarjeta.



Figura 66. Versión 1.2 del PCB del módulo principal.

La Figura 67 muestra una comparativa de la cara superior de la versión 1.1 del PCB del módulo principal y de la nueva versión 1.2. En la parte inferior de la figura se puede ver la cara inferior de la tarjeta.

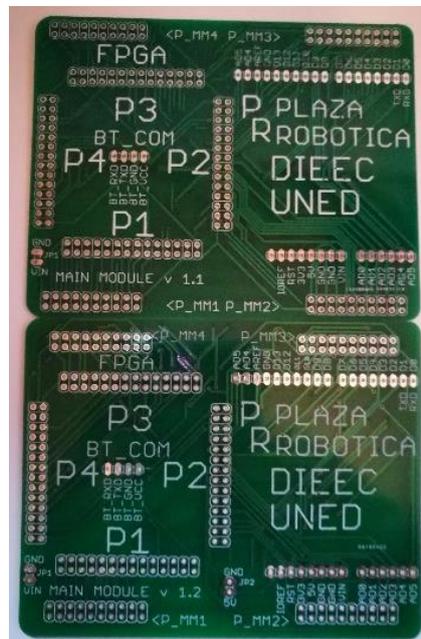


Figura 67. Versiones 1.1 (arriba) y 1.2 (abajo) del PCB del módulo principal.

La Figura 68 muestra la versión 1.2 del PCB del módulo secundario. En la parte superior de la figura se puede ver la cara superior de la tarjeta y en la parte inferior de la figura se puede ver la cara inferior de la tarjeta.

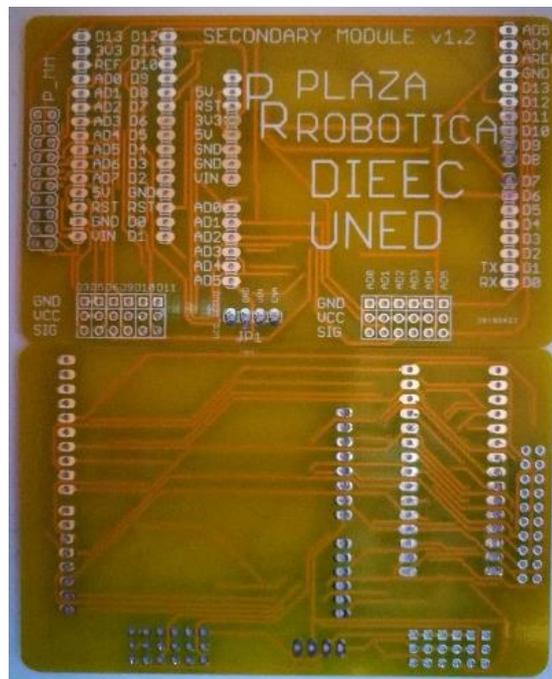


Figura 68. Versión 1.2 del PCB del módulo secundario.

La Figura 69 muestra una comparativa de la cara superior de la versión 1.1 del PCB del módulo secundario y de la nueva versión 1.2.

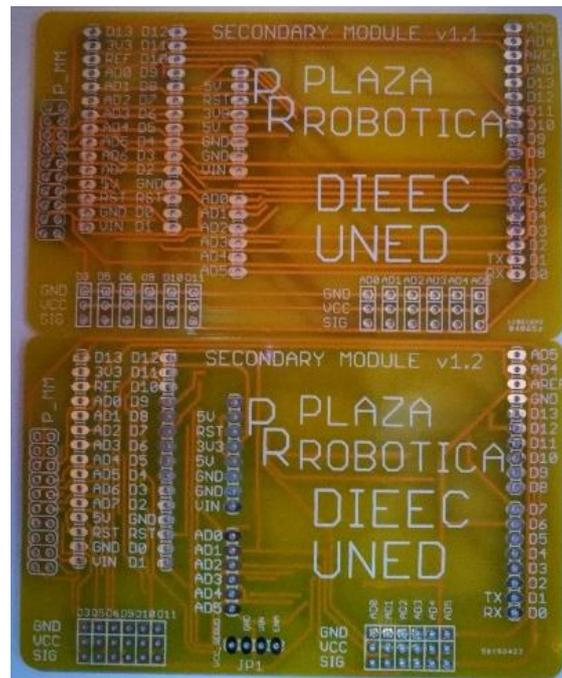


Figura 69. Versiones 1.1 (arriba) y 1.2 (abajo) del PCB del módulo secundario.

La Figura 70 muestra la versión 1.2 del módulo principal con todos los conectores y tarjeta montados. En la parte inferior de la figura se puede ver la cara inferior de la tarjeta.



Figura 70. Versión 1.2 del módulo principal.

La Figura 71 muestra una comparativa de la cara superior de la versión 1.1 de la PBA del módulo principal y de la nueva versión 1.2.



Figura 71. Versiones 1.1 (arriba) y 1.2 (abajo) de la PBA del módulo principal.

La Figura 72 muestra la versión 1.2 del módulo secundario con todos los conectores ya montados.

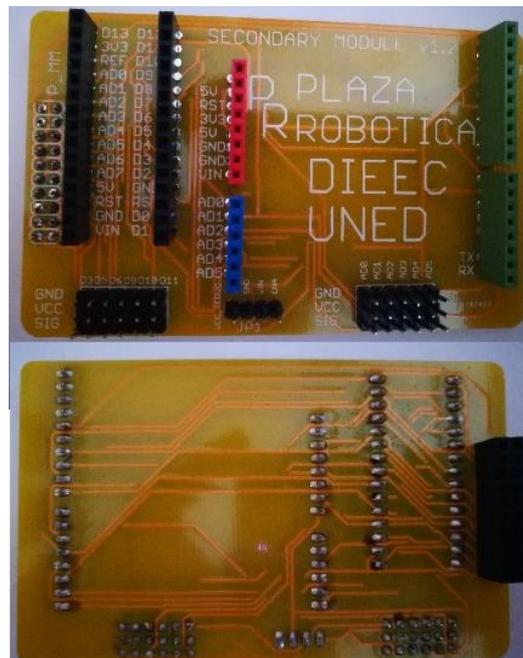


Figura 72. Versión 1.2 del módulo secundario.

La Figura 73 muestra una comparativa de la cara superior de la versión 1.1 de la PBA del módulo secundario y de la nueva versión 1.2.

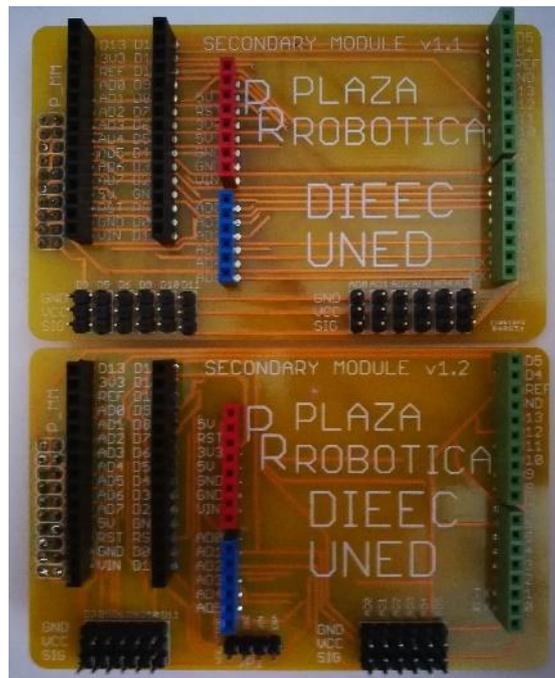


Figura 73. Versiones 1.1 (arriba) y 1.2 (abajo) de la PBA del módulo secundario.

Esta versión ya permite montar todos los elementos. La Figura 74 muestra la versión 1.2 del montaje completo que incluye:

- Sistema básico de desarrollo FPGA basado en Altera Cyclone II EP2C5T144
- Módulo principal versión 1.2
- Módulo Bluetooth
- Shield Wifi
- Shield TFT LCD Táctil
- 4 módulos secundarios versión 1.2
- 4 Arduino Nano
- 4 Shield de prototipado

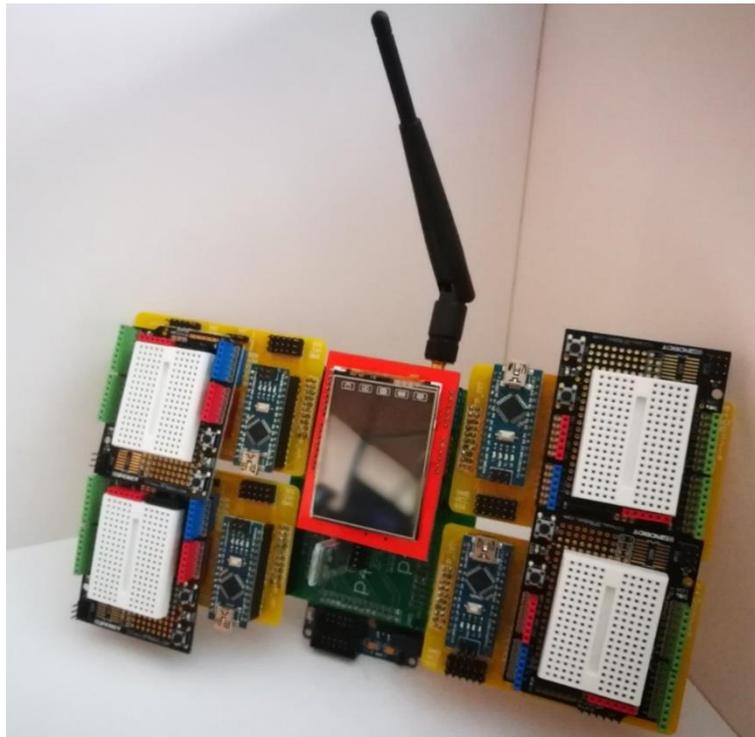


Figura 74. Versión 1.2 de la Plataforma.

Tras obtener la versión 1.2 del módulo principal y del módulo secundario se procede a realizar las pruebas de integridad del hardware descritas en la sección 5.1.7:

1. Se comprueba la continuidad o resistencia eléctrica entre puntos de la tarjeta que estén unidos por una pista. También se mide la no continuidad entre puntos del PCB contiguos.
2. En un segundo paso se aplica alimentación y se mide tensión en los puntos donde debe haber tensión de alimentación.

Una vez concluidas las pruebas se comprueba que la versión 1.2 del módulo principal y del módulo secundario está lista para ser usada en las pruebas de integración hardware-firmware – ver sección 6.3.

6.5. Cuarto prototipo, versión 1.3

Esta versión contiene los cambios mencionados en la sección anterior, además de las correcciones propuestas como resultado de la integración hardware-firmware-software detectadas en la versión 1.0 del software –sección 7.2. De acuerdo con la metodología descrita en el capítulo 5, los cambios aplicados para la cuarta versión del prototipo, versión 1.3 son motivados por las pruebas de integración de hardware-firmware-software. La Figura 75 muestra la transición que proviene de la integración hardware-firmware-software y afecta al desarrollo de hardware.

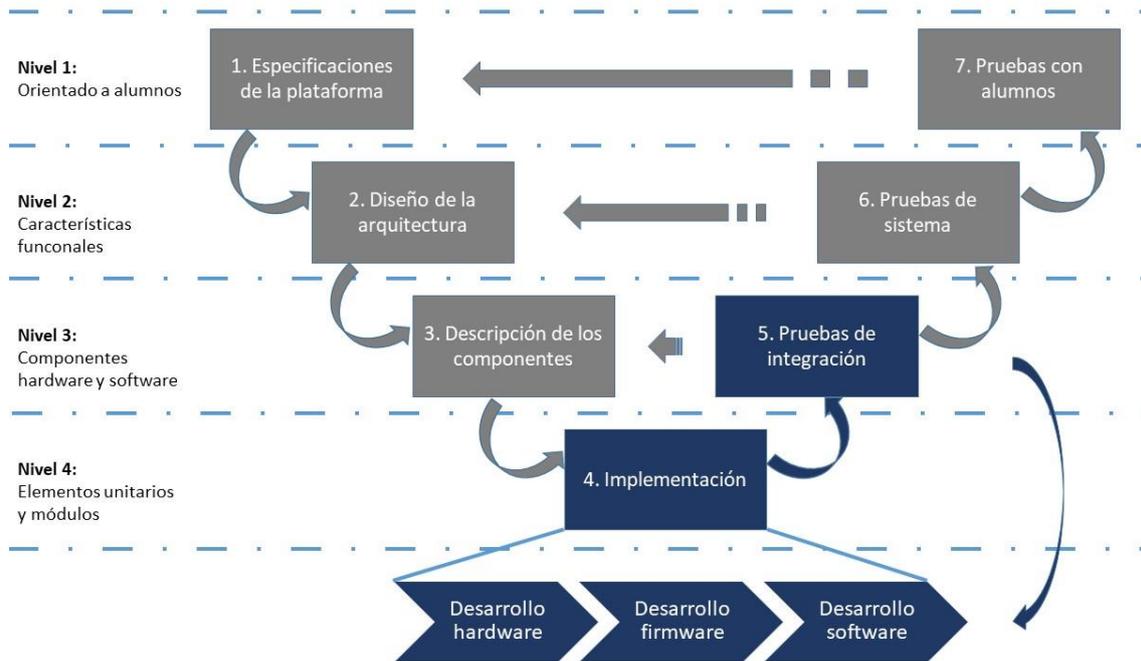


Figura 75. Diagrama de bloques del tercer cambio hardware.

Los cambios son los siguientes:

- El diseño del PCB del módulo principal se mejora realizando las siguientes modificaciones:
 - Los conectores P_MM se recolocan a una posición más externa del PCB.
 - Se añade el componente ST2378ETTR del fabricante STMicroelectronics. Se trata de un traductor de niveles de tensión bidireccional. Usando este componente se convierten los niveles de 3,3 Vdc de la FPGA a niveles de 5 Vdc de las Arduino Nano. Se coloca un convertidor en las proximidades de cada uno de los conectores que hay en el módulo principal para conectar con cada uno de los módulos secundarios. La Figura 76 muestra el aspecto del componente ST2378ETTR que tienen un encapsulado TSSOP.
 - Los conectores entre el módulo principal y los módulos secundarios pasan de tener 20 pines y 14 líneas de comunicación a tener 10 pines y ocho líneas de comunicación. Las señales que se eliminan son las señales conectadas a los puertos PWM de las Arduino Nano. La Tabla 76 detalla la correspondencia entre las entradas/salidas de la tarjeta Arduino Nano y el conector P_MM.
- El diseño del PCB del módulo secundario se puede cambia el conector entre el módulo principal y el módulo secundarios pasan de tener 20 pines y 14 líneas de

comunicación a tener 10 pines y ocho líneas de comunicación. Las señales que se eliminan son las señales conectadas a los puertos PWM de las Arduino Nano.



Figura 76. Esquemáticos de la versión 1.3 del módulo principal.

Tabla 76. Relación entre las señales del módulo principal y el conector P_MM.

	Conector P_MM				
Pin Conector	2	4	6	8	10
Señal módulo principal	VIN	D1	D4	D8	D13
Pin Conector	1	3	5	7	9
Señal módulo principal	GND	D0	D2	D7	D12

Para aplicar los cambios que contiene esta segunda versión de tarjetas, se realizan las modificaciones pertinentes utilizando la metodología de desarrollo hardware a partir del paso 3, diseño de los esquemáticos. Como resultado de la ejecución de este paso, se han obtenido los esquemáticos para la versión 1.3 del módulo principal. La Figura 77 y la Figura 78 muestran los esquemáticos de esta primera versión del módulo principal y del módulo secundario respectivamente. Como se puede ver en dichas figuras, quedan representados los conectores para cada uno de los elementos de dichas tarjetas y las conexiones que hay entre los diferentes conectores.

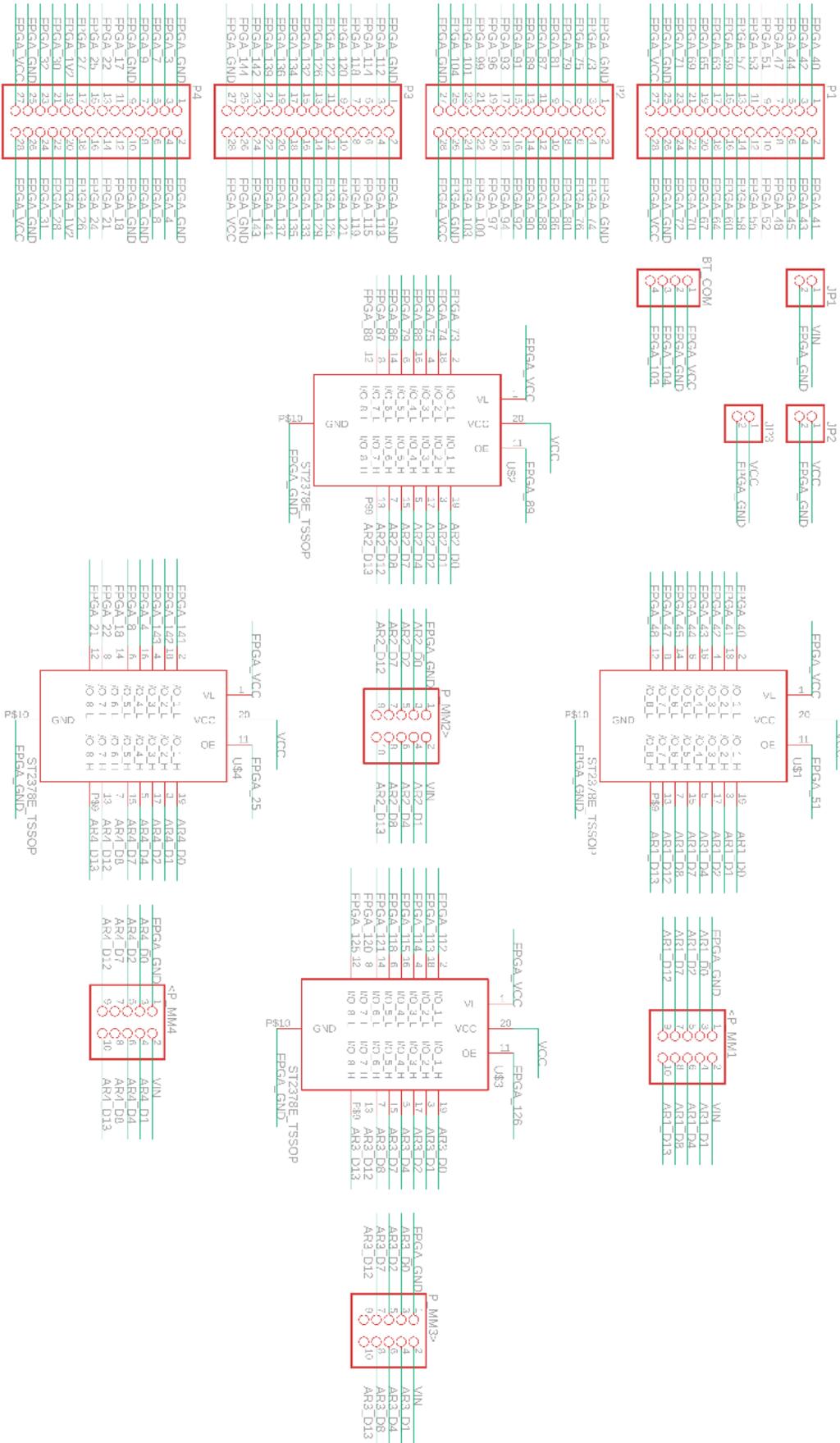


Figura 77. Esquemáticos de la versión 1.3 del módulo principal.

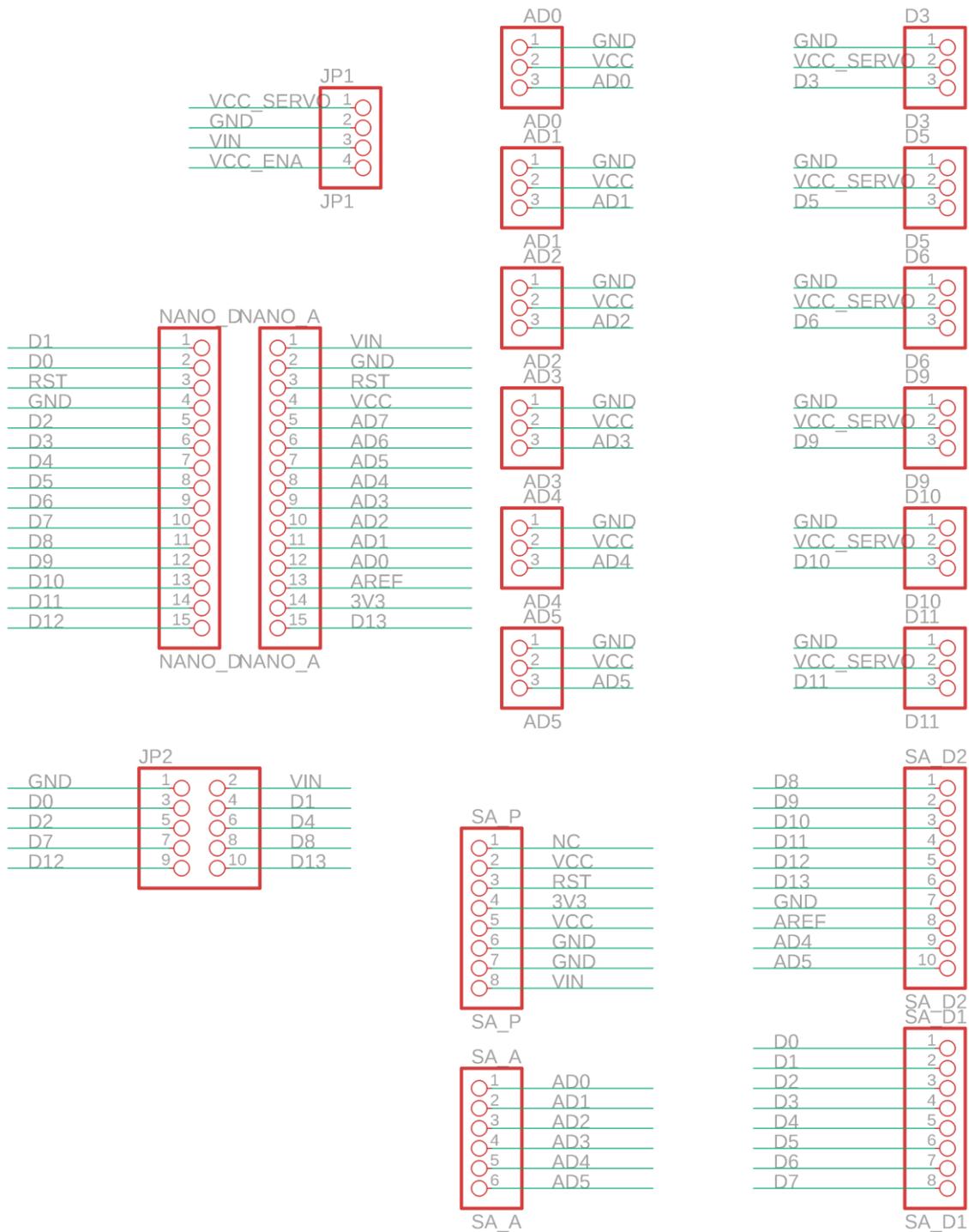


Figura 78. Esquemáticos de la versión 1.3 del módulo secundario.

Posteriormente se procede al paso 4, diseño físico del PCB. En este caso, se han recolocado todos los elementos de la tarjeta en la localización correcta y se han posicionado todas las conexiones entre los diferentes elementos de manera que no quede ninguna conexión sin trazar. El resultado de este cuarto paso lo muestra la Figura 79 y la Figura 80 para el módulo principal y para el módulo secundario.

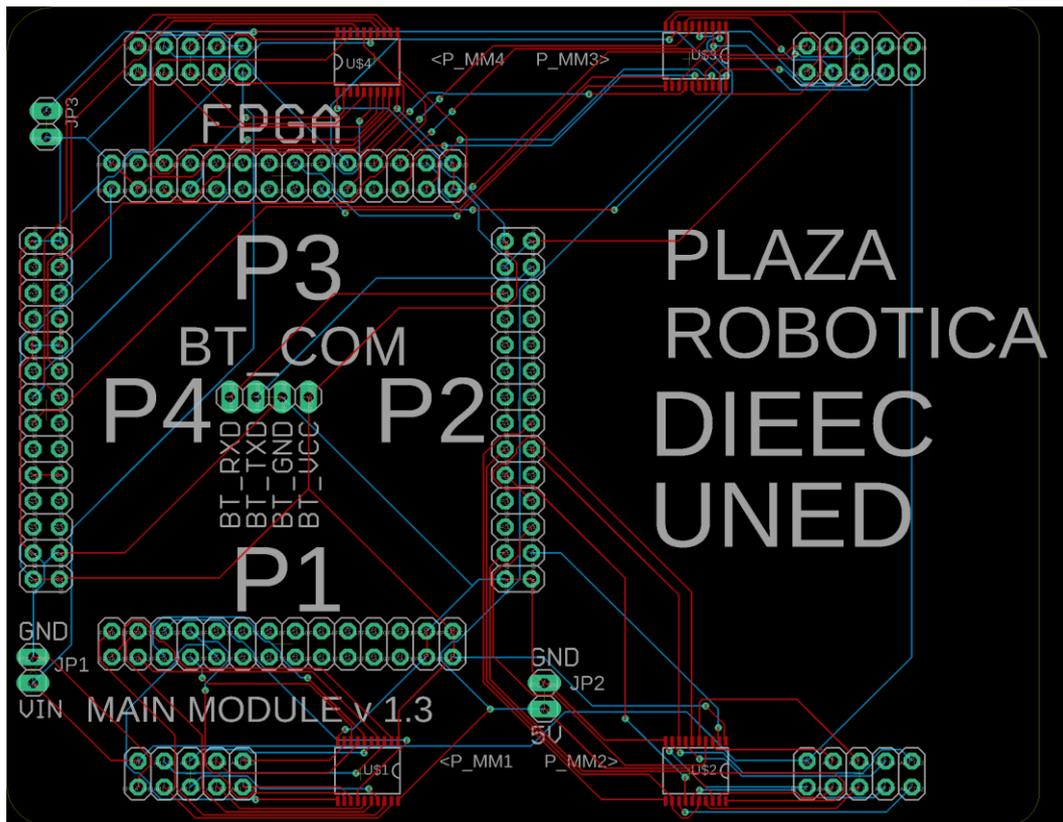


Figura 79. Layout de la versión 1.3 del módulo principal.

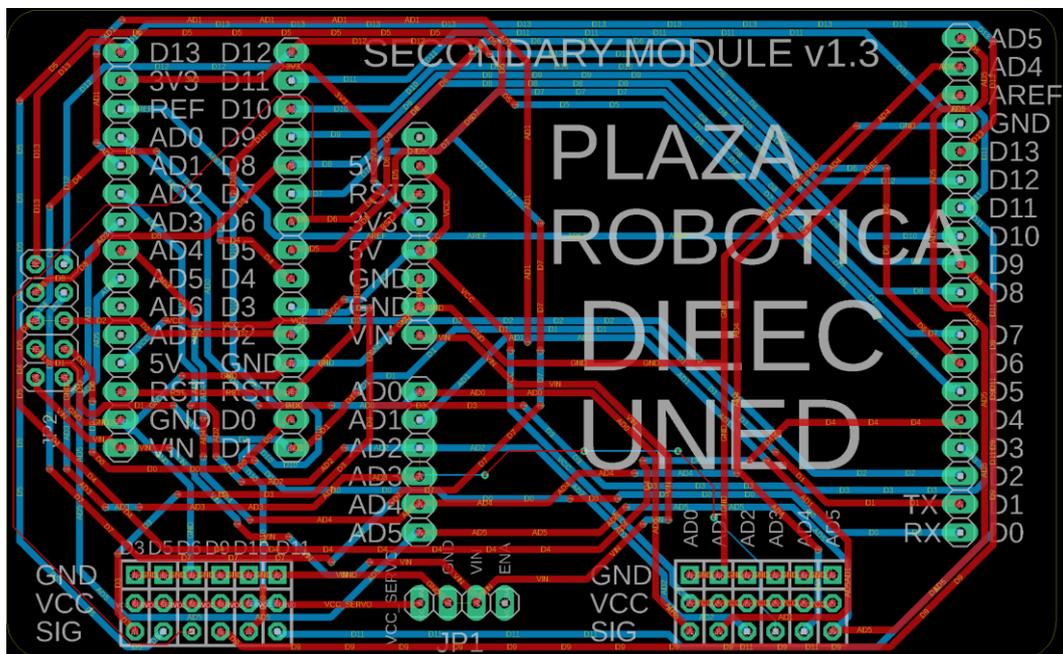


Figura 80. Layout de la versión 1.3 del módulo secundario.

A lo largo del paso 5 se generan los archivos de fabricación. Para la tercera versión del módulo principal se han obtenido los siguientes archivos:

- MAIN_MODULE.drd

- MAIN_MODULE.dri
- MAIN_MODULE.cmp
- MAIN_MODULE.sol
- MAIN_MODULE.plc
- MAIN_MODULE.stc
- MAIN_MODULE.sts
- MAIN_MODULE.gpi

Para la tercera versión del módulo secundario se han obtenido los siguientes archivos:

- SECONDARY_MODULE.drd
- SECONDARY_MODULE.dri
- SECONDARY_MODULE.cmp
- SECONDARY_MODULE.sol
- SECONDARY_MODULE.plc
- SECONDARY_MODULE.stc
- SECONDARY_MODULE.sts
- SECONDARY_MODULE.gpi

La Figura 81 muestra la versión 1.3 del PCB del módulo principal. En la parte superior de la figura se puede ver la cara superior de la tarjeta y en la parte inferior de la figura se puede ver la cara inferior de la tarjeta.

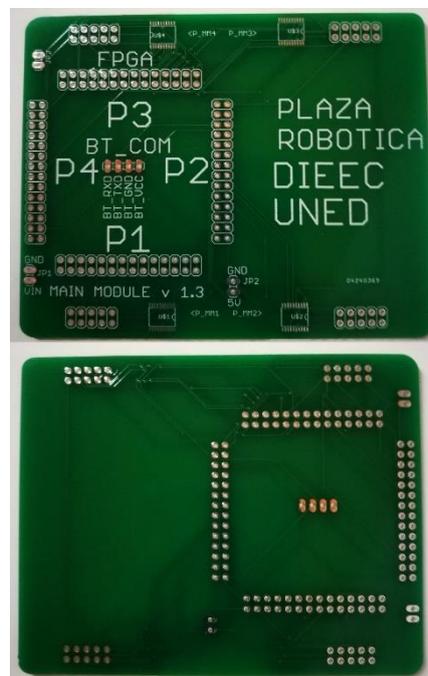


Figura 81. Versión 1.3 del PCB del módulo principal.

La Figura 82 muestra una comparativa de la cara superior de la versión 1.2 del PCB del módulo principal y de la nueva versión 1.3.

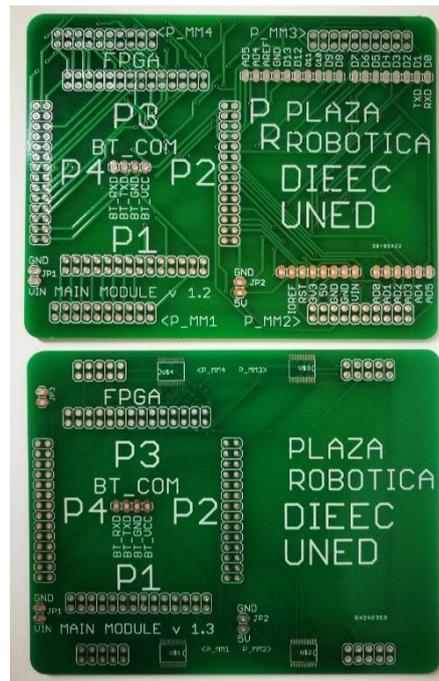


Figura 82. Versiones 1.2 (arriba) y 1.3 (abajo) del PCB del módulo principal.

La Figura 83 muestra la versión 1.3 del PCB del módulo secundario. En la parte superior de la figura se puede ver la cara superior de la tarjeta y en la parte inferior de la figura se puede ver la cara inferior de la tarjeta.



Figura 83. Versión 1.3 del PCB del módulo secundario.

La Figura 84 muestra una comparativa de la cara superior de la versión 1.2 del PCB del módulo secundario y de la nueva versión 1.3.

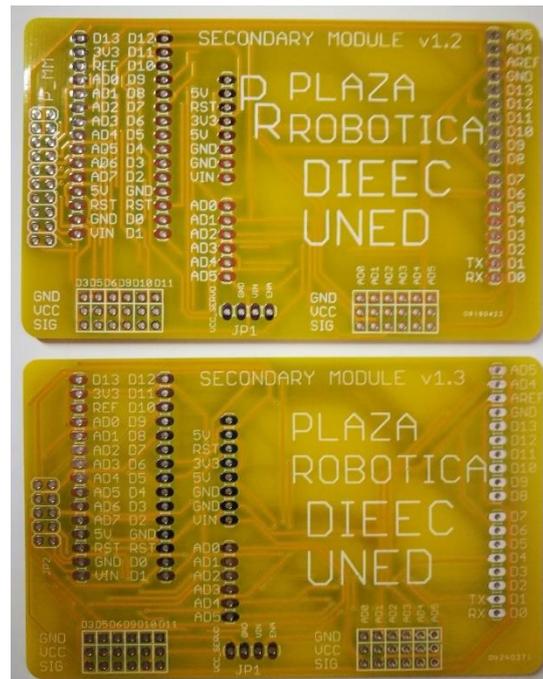


Figura 84. Versiones 1.2 (arriba) y 1.3 (abajo) del PCB del módulo secundario.

Los componentes ST2378ETTR son de tipo SMD. Para su montaje se utiliza una estación de soldadura como el que la Figura 85 muestra. Esta herramienta tiene las siguientes características principales:

- Contiene pistola de aire caliente y fuente de alimentación DC. Es un equipo para soldar y desoldar SOIC, QFP, PLCC, BGA, SMD y demás componentes electrónicos.
- Tiene una pantalla digital LED para mostrar la temperatura y se puede calentar rápidamente.
- Es válida para una amplia gama de aplicaciones: La estación soldadura aire caliente apto para la reparación de teléfono móvil, laboratorios, investigación científica, mantenimiento de Laptop, las líneas de producción, etc.



Figura 85. Estación de soldadura.

Además, para poder comprobar que las soldaduras se han realizado correctamente, ha sido necesario el uso de un microscopio electrónico como el que la Figura 86 muestra.



Figura 86. Microscopio electrónico.

La Figura 87 muestra el resultado tras el proceso de soldadura por chorro de aire caliente.

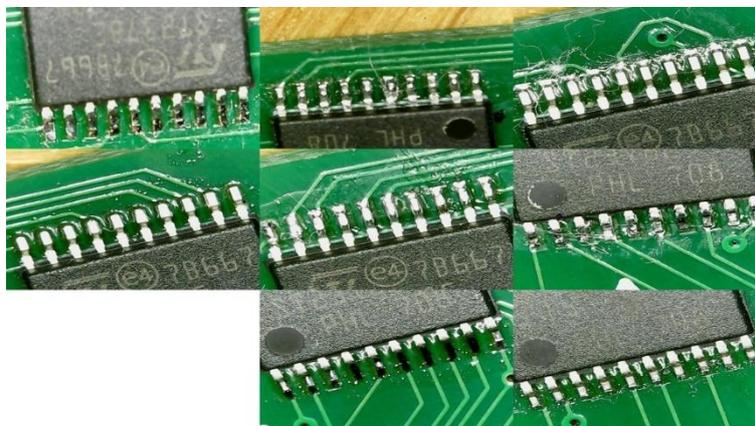


Figura 87. Resultado de la soldadura de los componentes ST2378ETTR.

La Figura 88 muestra la versión 1.3 del módulo principal con todos los conectores ya montados. En la parte inferior de la figura se puede ver la cara inferior de la tarjeta.



Figura 88. Versión 1.3 del módulo principal.

La Figura 89 muestra una comparativa de la cara superior de la versión 1.2 de la PBA del módulo principal y de la nueva versión 1.3.

10

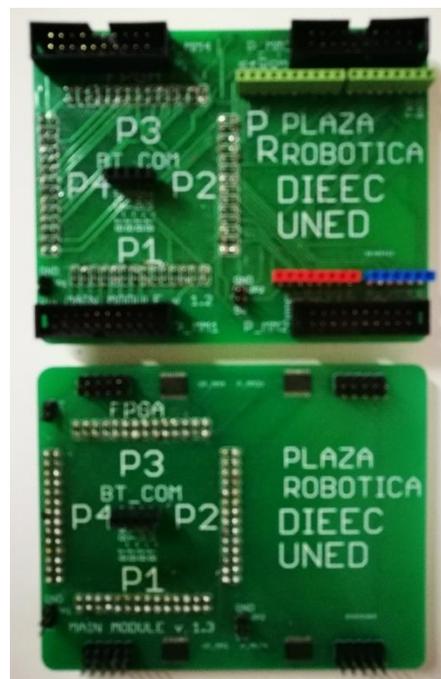


Figura 89. Versiones 1.2 (arriba) y 1.3 (abajo) de la PBA del módulo principal.

La Figura 90 muestra la versión 1.3 del módulo secundario con todos los conectores ya montados.

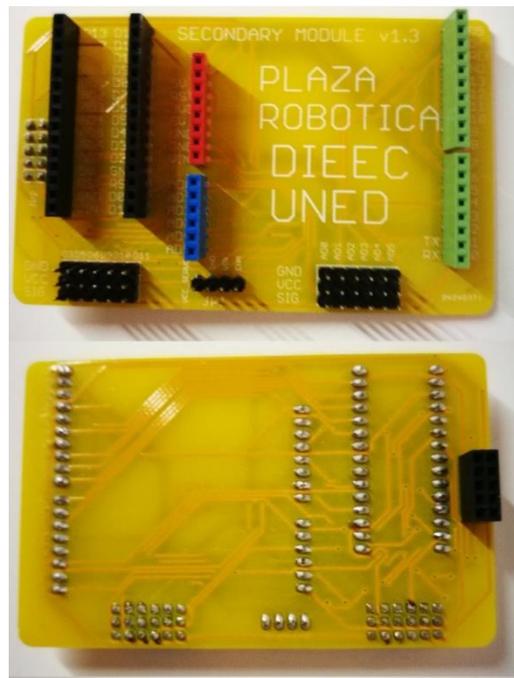


Figura 90. Versión 1.3 del módulo secundario.

La Figura 91 muestra una comparativa de la cara superior de la versión 1.2 de la PBA del módulo secundario y de la nueva versión 1.3.

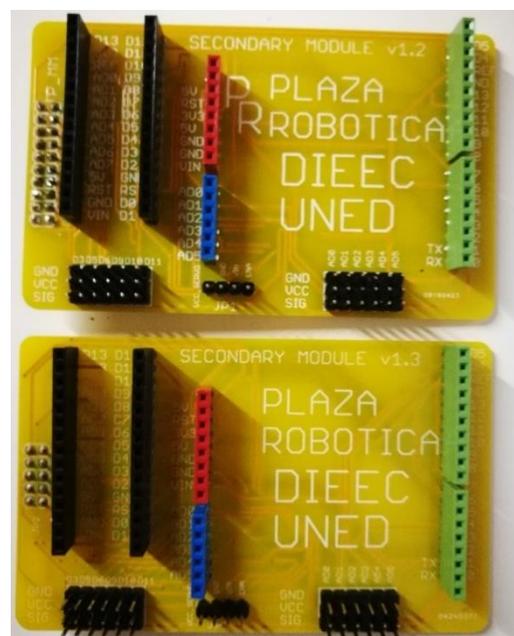


Figura 91. Versiones 1.2 (arriba) y 1.3 (abajo) de la PBA del módulo secundario.

Esta versión ya permite montar todos los elementos. La Figura 92 muestra la versión 1.3 del montaje completo que incluye:

- Sistema básico de desarrollo FPGA basado en Altera Cyclone II EP2C5T144.
- Módulo principal versión 1.3.
- Módulo Bluetooth.
- Shield TFT LCD Táctil.
- 4 módulos secundarios versión 1.3.
- 4 Arduino Nano.
- 4 Shield de prototipado.
- Componentes para actividades con LED

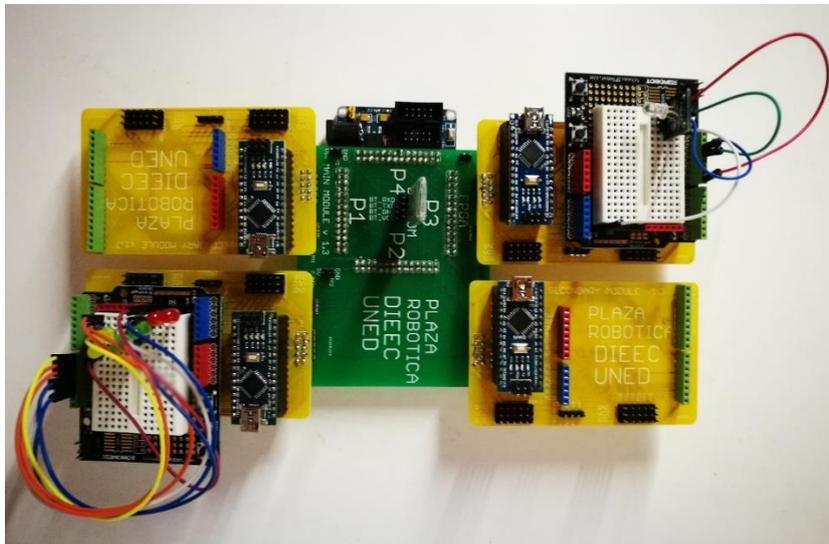


Figura 92. Versión 1.3 de la Plataforma.

Tras obtener la versión 1.3 del módulo principal y del módulo secundario se procede a realizar las pruebas de integridad del hardware descritas en la sección 5.1.7:

1. Se comprueba la continuidad o resistencia eléctrica entre puntos de la tarjeta que estén unidos por una pista. También se mide la no continuidad entre puntos del PCB contiguos.
2. En un segundo paso se aplica alimentación y se mide tensión en los puntos donde debe haber tensión de alimentación.

Una vez concluidas las pruebas se comprueba que la versión 1.3 del módulo principal y del módulo secundario está lista para ser usada en las pruebas de integración hardware-firmware – ver sección 6.4.

CAPÍTULO 7

7. DESARROLLO FIRMWARE

En este capítulo se recoge toda la información recopilada durante diseño firmware del desarrollo de la arquitectura de la plataforma de robótica educativa. De acuerdo con la arquitectura definida en el capítulo 5, es necesario el desarrollo de firmware para que la FPGA que forma parte del sistema gestione las funcionalidades requeridas por parte del firmware.

7.1. Metodología de desarrollo firmware

El desarrollo firmware de la Plataforma se encuentra ubicado en la etapa 5, referente a la implementación de la Metodología de Modelo en V usado a lo largo de la elaboración de la arquitectura de la Plataforma, tal y como se describe en el capítulo 5 a lo largo de la sección 3.

Para el desarrollo firmware de la Plataforma Robótica Educativa se ha elaborado una metodología de desarrollo de firmware. Esta metodología consiste en varios pasos, como son el diseño, la implementación y la verificación del firmware de la FPGA del módulo principal:

1. Se indican las especificaciones identificadas que tienen relación con el firmware.
2. Posteriormente se identifican los bloques funcionales que van a ser implementados en la FPGA y que serán fundamentales a la hora de implementar las funcionalidades especificadas.
3. Se procede a la elaboración de los archivos fuente que describen mediante un lenguaje de descripción de hardware, HDL (Hardware Description Language) dichos bloques funcionales.
4. Tras la elaboración de los archivos fuente, en un siguiente paso, se procede a la verificación de estos mediante simulación, es decir, elaboración de bancos de pruebas y análisis de las formas de onda de las salidas de los bloques funcionales. De esta manera es posible comprobar el comportamiento de los bloques funcionales frente a las especificaciones de estos.
5. Una vez concluidos los pasos anteriormente mencionados, se genera el proyecto en la herramienta de diseño de FPGA para obtener los archivos de programación, tanto de la FPGA, como de la memoria de configuración.

6. Tras la comprobación del cumplimiento de las restricciones de la FPGA (uso de recursos y retardo de las señales), se realizan pruebas de verificación para comprobar que el diseño firmware se ajusta a las especificaciones. Para ello, será necesario contar con diferentes elementos hardware, y en este paso se realizará una integración parcial de la Plataforma, integración hardware-firmware.

Para acometer el paso 4, pruebas de verificación de los bloques funcionales implementados, se ha escogido la herramienta Altera Modelsim [Modelsim, 2018]. Se trata de una versión de la herramienta Modelsim, cuyo uso es gratuito cuando se usa juntamente con la herramienta Quartus II del fabricante de FPGAs Altera [Quartus II, 2018a]. Modelsim es un software que administra un entorno que permite editar, compilar, simular y depurar diseños de sistemas digitales descritos en VHDL, Verilog y SystemC. La Figura 93 representa el flujo de diseño con Modelsim [Modelsim, 2018]:

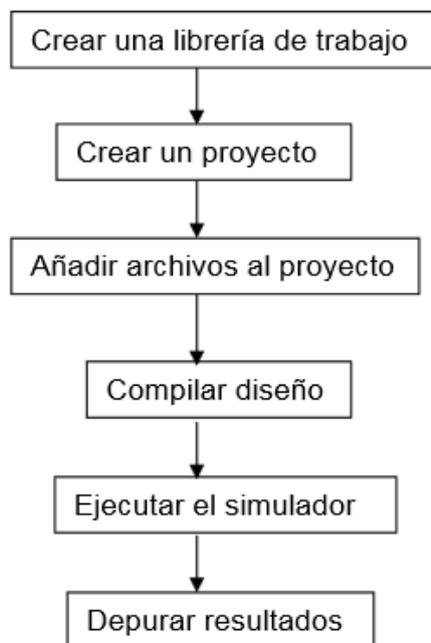


Figura 93. Flujo de verificación con Modelsim.

Cuando se usa la herramienta Modelsim, existen tres modos de operación:

- Mediante procesos por lotes.
- Línea de comandos (se trata de un entorno interactivo en modo texto, accesible desde el 'prompt' del sistema operativo con el comando vsim -c).
- Ejecución basada en GUI (Graphic User Interface). Es el modo por defecto y el más habitual pues utiliza una interfaz amigable con el usuario. Es interactiva, maneja menús, ventanas, botones y líneas de comando.

Para acometer el paso 5, se hace uso de la herramienta Quartus II [Quartus II, 2018a]. El software de diseño de Altera, Quartus II, ofrece un completo entorno de diseño multi-plataforma que se adapta fácilmente a las necesidades específicas de diseño. Incluye soluciones para todas las fases de diseño de FPGA y el diseño CPLD [Quartus II, 2018b], [Quartus II, 2018c] y [Quartus II, 2018d].

La Figura 93 muestra las fases de diseño. Estas fases se resumen en los siguientes puntos:

- **Design Entry:** un proyecto Quartus II incluye todos los archivos de diseño, de código fuente y resto de archivos necesarios para la implementación de un diseño de un dispositivo lógico programable. Se puede utilizar el Quartus II Block Editor, Editor de Texto, MegaWizard Plugin Manager y herramientas EDA de entrada de diseño para crear archivos de diseño que incluyen mega-funciones de Altera, módulos de parámetros de biblioteca (LPM), y funciones de propiedad intelectual (IP).
- **Synthesis:** se puede utilizar el módulo de Análisis y Síntesis del compilador para analizar sus archivos de diseño y creación de la base de datos del proyecto. se utiliza el análisis y síntesis Quartus II para sintetizar los archivos de diseño Verilog (. V) o archivos de diseño VHDL (. vhd).
- **Place & Route:** el Fitter de Quartus II, también se conoce como "fitting", realiza el rutado y localización de la lógica descrita en el diseño. Usando la base de datos que ha sido creado por Análisis y Síntesis, el Fitter trata de cumplir los requisitos de lógica y temporales con los recursos disponibles en el dispositivo de destino. Se asigna a cada función lógica la mejor ubicación de la celda lógica de enrutamiento y localización, y se seleccionan los caminos de interconexión apropiados y las asignaciones de pin. Esta fase está compuesta por las siguientes sub-fases:
 - Power Analysis
 - Debugging
 - Engineering Change Management
 - Timing Closure
- **Timing Analysis:** el analizador de tiempos Timequest permite analizar las características de sincronización del diseño. El analizador utiliza la metodología industry-standard Synopsys Design Constraint (SDC) para limitar los diseños e informar de los resultados. Se puede utilizar la información generada por el analizador de tiempos para analizar, depurar y validar el rendimiento de la sincronización del diseño.
- **Simulation:** en este punto se podrá realizar un análisis del comportamiento del diseño completo para poder verificar que se cumplen los requisitos funcionales.

- Programming & Configuration: una vez que haya compilado correctamente un proyecto con el software Quartus II, se puede programar o configurar un dispositivo Altera. El módulo ensamblador del compilador Quartus II genera archivos de programación que utilizar para configurar el dispositivo de lógica programable. Para grabar los archivos de programación, también puede utilizar una versión independiente del dispositivo programador de Quartus II para configurar dispositivos.

El software Quartus II permite el uso de la interfaz gráfica de usuario y el interfaz de línea de comandos para cada fase del flujo de diseño.

La Figura 94 muestra un diagrama con el flujo de diseño guiado cuando se emplea la herramienta de software Quartus II y su metodología de uso asociada. Como muestra la Figura 94 se define una secuenciación de pasos definidos en la metodología de diseño de firmware de la Plataforma.

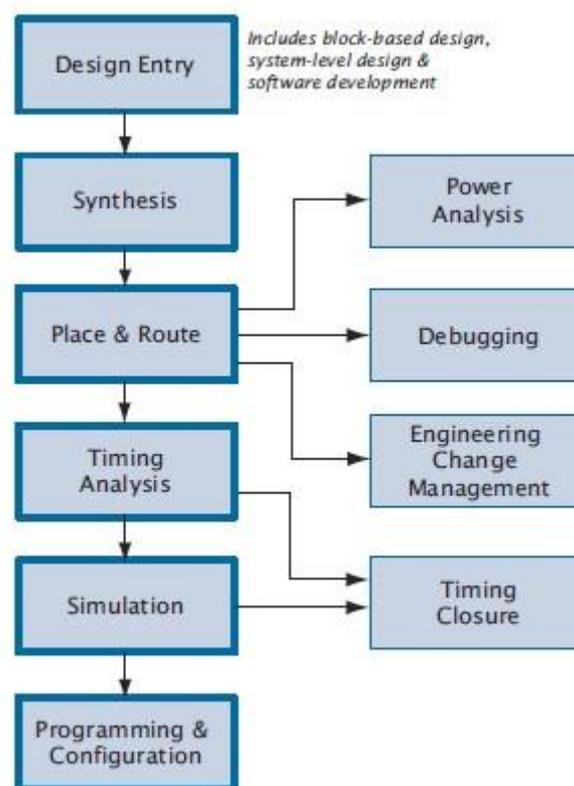


Figura 94. Flujo de diseño usando QUARTUS II.

El software Quartus II incluye un compilador modular. El compilador incluye los siguientes módulos:

- Analysis & Synthesis: se analiza y sintetiza el diseño descrito
- Partition Merge: se unifican las particiones generadas

- Fitter: se adapta el diseño inferido al hardware específico que se va a usar para implementar el diseño
- Assembler: se genera la netlist con los diferentes elementos del dispositivo programable
- TimeQuest Timing Analyzer: se analiza la respuesta de las diferentes funciones lógicas para detectar si las señales llegarán con retardo o cumplen las restricciones temporales configuradas
- Design Assistant: se muestra los resultados de implementación
- EDA Netlist Writer: se graba el dispositivo de lógica programable con el diseño
- HardCopy® Netlist Writer: se graba la memoria de configuración del dispositivo programable

La Figura 95 muestra el aspecto que tiene el Quartus II. Gracias a la distribución modular de esta herramienta se puede tener una visión general del diseño, el estado de implementación y un fácil acceso a los resultados de cada etapa del proceso de diseño de firmware.

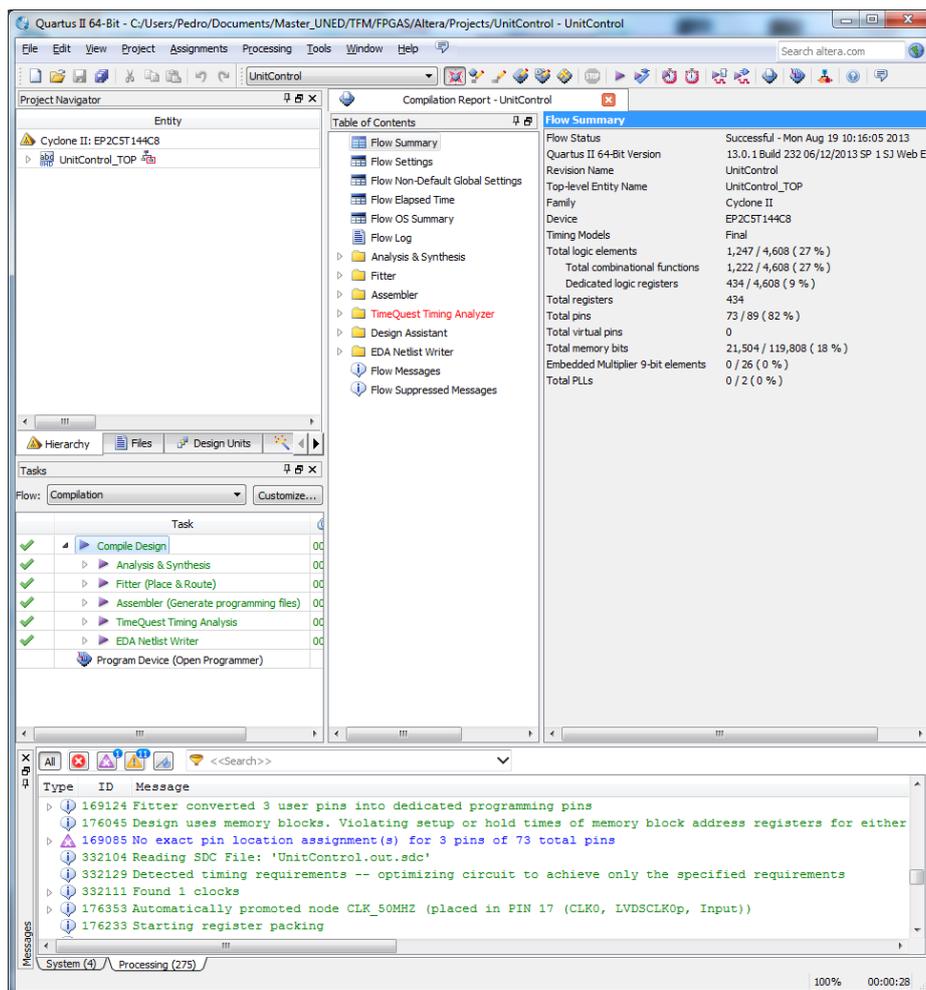


Figura 95. QUARTUS II web edition 13.0SP1.

Si se detectase algún error de diseño o incumplimiento de cualquier especificación, se evaluará si es posible continuar con las siguientes etapas, y si fuese necesario, se realizarán tareas de rediseño hasta que se obtenga una versión de firmware que cumpla con las especificaciones fijadas.

Las sucesivas secciones muestran toda la información de diseño obtenida a lo largo del desarrollo de las versiones firmware, incluyendo errores de diseño o funcionalidades que no se cumplen, paso en el que se detectó y soluciones propuestas para que sean subsanados.

7.1.1. PASO 1: ESPECIFICACIONES FIRMWARE

Las especificaciones firmware consisten en una serie de restricciones que se usan para la toma de decisiones de diseño en los siguientes pasos del diseño firmware de la plataforma. Estas decisiones de diseño se orientan, por un lado, en la elección de los puertos de entrada/salida de la FPGA, y, por otro lado, las funcionalidades asociadas a dichos puertos.

Puertos de entrada/salida de la FPGA

El módulo principal es un PCB que hace de conexión entre el EP2C5/EP2C8 MINI BOARD y los módulos secundarios. Contiene los conectores, el rutado de las líneas y contiene conexiones con módulo de comunicaciones Bluetooth y Shield de Arduino.

Los puertos de la FPGA están conectados con el módulo Bluetooth. La Tabla 77 detalla la funcionalidad de cada uno de ellos. Son necesarios dos puertos para la comunicación. En primer lugar, un puerto de transmisión de datos para enviar información desde la FPGA al módulo Bluetooth. Por otro lado, un puerto de recepción de datos para recibir información desde el módulo Bluetooth en la FPGA.

Tabla 77. Puertos de la FPGA conectados al módulo Bluetooth.

Conexiones con módulo Bluetooth	
Pin FPGA	Funcionalidad
103	Puerto de transmisión de datos de la FPGA al módulo Bluetooth.
104	Puerto de recepción de datos de la FPGA del módulo Bluetooth.

Los puertos de la FPGA están conectados con el Shield Arduino del módulo principal. La Tabla 78 detalla la funcionalidad de cada uno de ellos. Hay tres grupos de puertos de conexión. En primer lugar, los puertos de intercambio de información entre la FPGA y el Shield Arduino. En segundo lugar, puertos digitales de entrada y de salida entre la FPGA

y el Shield de Arduino. También se conectan los puertos PWM del Shield Arduino con la FPGA.

Tabla 78. Puertos de la FPGA conectados al Shield de Arduino.

Conexiones con Shield Arduino	
Pin FPGA	Funcionalidad
63	Puerto de transmisión de datos de la FPGA al Shield Arduino.
64	Puerto de recepción de datos de la FPGA del Shield Arduino.
65	Puerto digital D2 del Shield Arduino. Entrada discreta 1 del Shield Arduino.
67	Puerto digital D3 del Shield Arduino. Puerto PWM.
69	Puerto digital D4 del Shield Arduino. Entrada discreta 2 del Shield Arduino.
70	Puerto digital D5 del Shield Arduino. Puerto PWM.
71	Puerto digital D6 del Shield Arduino. Puerto PWM.
72	Puerto digital D7 del Shield Arduino. Entrada discreta 3 del Shield Arduino.
96	Puerto digital D8 del Shield Arduino. Salida discreta 1 del Shield Arduino.
97	Puerto digital D9 del Shield Arduino. Puerto PWM.
99	Puerto digital D10 del Shield Arduino. Puerto PWM.
100	Puerto digital D11 del Shield Arduino. Puerto PWM.
101	Puerto digital D12 del Shield Arduino. Salida discreta 2 del Shield Arduino.
136	Puerto digital D13 del Shield Arduino. Salida discreta 3 del Shield Arduino.

Los puertos de la FPGA están conectados con el módulo secundario 1 mediante el conector P_MM_1, con el módulo secundario 2 mediante el conector P_MM_2, con el módulo secundario 3 mediante el conector P_MM_3 y con el módulo secundario 4 mediante el conector P_MM_4. Las tablas Tabla 79, Tabla 80, Tabla 81 y Tabla 82 detallan la funcionalidad de cada una de las conexiones. La FPGA tiene la capacidad de aplicar reset al módulo secundario. También se añaden puertos de comunicación serie entre la FPGA y el módulo secundario.

Tabla 79. Puertos de la FPGA conectados al módulo secundario 1.

Conector P_MM_1	
Pin FPGA	Funcionalidad
40	Reset de la Arduino Nano 1.
41	Puerto de transmisión de datos de la FPGA a la Arduino Nano 1.
42	Puerto de recepción de datos de la FPGA de la Arduino Nano 1.
43	Puerto digital D2 de la Arduino Nano 1. Entrada discreta 1 de la Arduino Nano 1.
44	Puerto digital D3 de la Arduino Nano 1. Puerto PWM.
45	Puerto digital D4 de la Arduino Nano 1. Entrada discreta 2 de la Arduino Nano 1.
47	Puerto digital D5 de la Arduino Nano 1. Puerto PWM.
48	Puerto digital D6 de la Arduino Nano 1. Puerto PWM.
51	Puerto digital D7 de la Arduino Nano 1. Entrada discreta 3 de la Arduino Nano 1.
52	Puerto digital D8 de la Arduino Nano 1. Salida discreta 1 de la Arduino Nano 1.

Conector P_MM_1	
Pin FPGA	Funcionalidad
53	Puerto digital D9 de la Arduino Nano 1. Puerto PWM.
55	Puerto digital D10 de la Arduino Nano 1. Puerto PWM.
57	Puerto digital D11 de la Arduino Nano 1. Puerto PWM.
58	Puerto digital D12 de la Arduino Nano 1. Salida discreta 2 de la Arduino Nano 1.
59	Puerto digital D13 de la Arduino Nano 1. Salida discreta 3 de la Arduino Nano 1.

Tabla 80. Puertos de la FPGA conectados al módulo secundario 2.

Conector P_MM_2	
Pin FPGA	Funcionalidad
73	Reset de la Arduino Nano 2.
74	Puerto de transmisión de datos de la FPGA a la Arduino Nano 2.
75	Puerto de recepción de datos de la FPGA de la Arduino Nano 2.
76	Puerto digital D2 de la Arduino Nano 2. Entrada discreta 1 de la Arduino Nano 2.
79	Puerto digital D3 de la Arduino Nano 2. Puerto PWM.
80	Puerto digital D4 de la Arduino Nano 2. Entrada discreta 2 de la Arduino Nano 2.
81	Puerto digital D5 de la Arduino Nano 2. Puerto PWM.
86	Puerto digital D6 de la Arduino Nano 2. Puerto PWM.
87	Puerto digital D7 de la Arduino Nano 2. Entrada discreta 3 de la Arduino Nano 2.
88	Puerto digital D8 de la Arduino Nano 2. Salida discreta 1 de la Arduino Nano 2.
89	Puerto digital D9 de la Arduino Nano 2. Puerto PWM.
90	Puerto digital D10 de la Arduino Nano 2. Puerto PWM.
91	Puerto digital D11 de la Arduino Nano 2. Puerto PWM.
92	Puerto digital D12 de la Arduino Nano 2. Salida discreta 2 de la Arduino Nano 2.
93	Puerto digital D13 de la Arduino Nano 2. Salida discreta 3 de la Arduino Nano 2.
94	Reset de la Arduino Nano 2.

Tabla 81. Puertos de la FPGA conectados al módulo secundario 3.

Conector P_MM_3	
Pin FPGA	Funcionalidad
112	Reset de la Arduino Nano 3.
113	Puerto de transmisión de datos de la FPGA a la Arduino Nano 3.
114	Puerto de recepción de datos de la FPGA de la Arduino Nano 3.
115	Puerto digital D2 de la Arduino Nano 3. Entrada discreta 1 de la Arduino Nano 3.
118	Puerto digital D3 de la Arduino Nano 3. Puerto PWM.
119	Puerto digital D4 de la Arduino Nano 3. Entrada discreta 2 de la Arduino Nano 3.
120	Puerto digital D5 de la Arduino Nano 3. Puerto PWM.
121	Puerto digital D6 de la Arduino Nano 3. Puerto PWM.
122	Puerto digital D7 de la Arduino Nano 3. Entrada discreta 3 de la Arduino Nano 3.
125	Puerto digital D8 de la Arduino Nano 3. Salida discreta 1 de la Arduino Nano 3.
126	Puerto digital D9 de la Arduino Nano 3. Puerto PWM.
129	Puerto digital D10 de la Arduino Nano 3. Puerto PWM.

Conector P_MM_3	
Pin FPGA	Funcionalidad
132	Puerto digital D11 de la Arduino Nano 3. Puerto PWM.
133	Puerto digital D12 de la Arduino Nano 3. Salida discreta 2 de la Arduino Nano 3.
134	Puerto digital D13 de la Arduino Nano 3. Salida discreta 3 de la Arduino Nano 3.
135	Reset de la Arduino Nano 3.

Tabla 82. Puertos de la FPGA conectados al módulo secundario 4.

Conector P_MM_4	
Pin FPGA	Funcionalidad
141	Reset de la Arduino Nano 4.
142	Puerto de transmisión de datos de la FPGA a la Arduino Nano 4.
143	Puerto de recepción de datos de la FPGA de la Arduino Nano 4.
4	Puerto digital D2 de la Arduino Nano 4. Entrada discreta 1 de la Arduino Nano 4.
8	Puerto digital D3 de la Arduino Nano 4. Puerto PWM.
18	Puerto digital D4 de la Arduino Nano 4. Entrada discreta 2 de la Arduino Nano 4.
22	Puerto digital D5 de la Arduino Nano 4. Puerto PWM.
21	Puerto digital D6 de la Arduino Nano 4. Puerto PWM.
25	Puerto digital D7 de la Arduino Nano 4. Entrada discreta 3 de la Arduino Nano 4.
24	Puerto digital D8 de la Arduino Nano 4. Salida discreta 1 de la Arduino Nano 4.
27	Puerto digital D9 de la Arduino Nano 4. Puerto PWM.
26	Puerto digital D10 de la Arduino Nano 4. Puerto PWM.
30	Puerto digital D11 de la Arduino Nano 4. Puerto PWM.
28	Puerto digital D12 de la Arduino Nano 4. Salida discreta 2 de la Arduino Nano 4.
32	Puerto digital D13 de la Arduino Nano 4. Salida discreta 3 de la Arduino Nano 4.
31	Reset de la Arduino Nano 4.

Funcionalidades de los puertos de la FPGA

El módulo secundario es un PCB que hace de conexión entre el módulo principal, la Arduino Nano y los Shield de Arduino. Contiene los conectores y el rutado de las líneas. También se pasan señales de reset del módulo principal a los módulos secundarios para poder hacer un reset a las Arduino Nano desde la FPGA.

En el módulo principal se pueden incluir hasta cuatro módulos secundarios, en función de las necesidades de uso relativas a sistemas robóticos que se quieran manejar, o grupos de alumnos que se creen para realizar diferentes actividades educativas.

La Figura 96 muestra las conexiones existentes entre los diferentes componentes hardware que componen la herramienta colaborativa de robótica educativa.

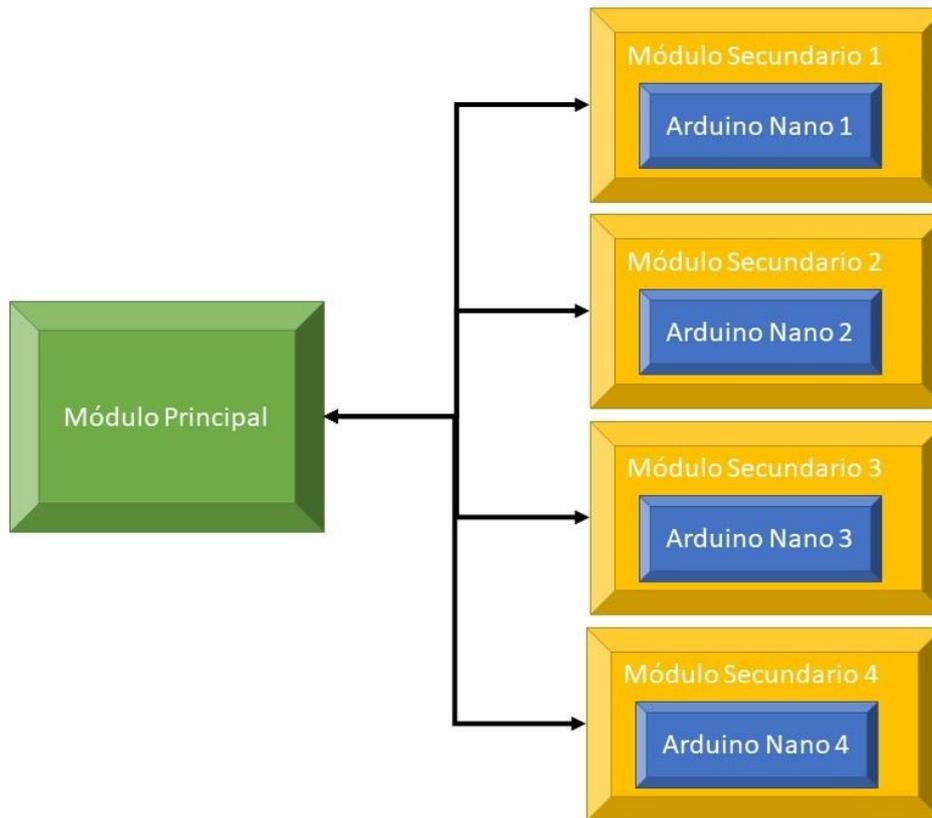


Figura 96. Conexiones entre los diferentes componentes hardware.

7.1.2. PASO 2: BLOQUES FUNCIONALES

Los bloques funcionales son enumerados, añadiendo en cada uno de ellos las funcionalidades que implementan en relación con las especificaciones fijadas para la FPGA. Adicionalmente, se incluye una enumeración de los puertos de entrada/salida con su correspondiente descripción.

Los bloques funcionales son UART Bluetooth, UART Shield, Comunicación discreta Shield, UART Arduino, Monitorización PWM y Comunicación discreta Arduino. La Figura 97 muestra un diagrama de comunicaciones entre cada uno de los bloques funcionales mencionados. También se muestran las conexiones entre dichos bloques funcionales y los componentes hardware con los que interactúan.

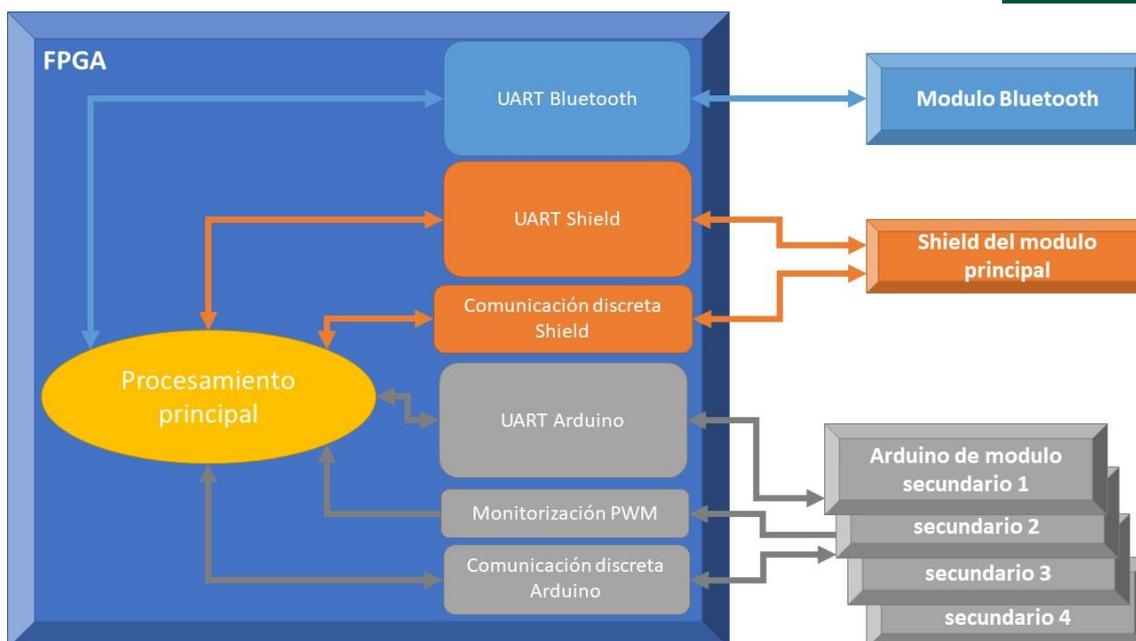


Figura 97. Diagrama de bloques de los bloques funcionales.

UART Bluetooth

Este bloque es el encargado de comunicarse con el módulo Bluetooth Arduino localizado en el módulo principal. Debe implementar un protocolo de comunicaciones RS232 y obtener los comandos recibidos y enviar los valores de todas las variables internas de estado. La Tabla 83 contiene los pines de la FPGA, el nombre del puerto y la descripción de la funcionalidad que desempeña.

Tabla 83. Puertos de la funcionalidad UART Bluetooth.

Puertos del UART Bluetooth		
Pin FPGA	Nombre del puerto	Funcionalidad
103	UART_BT_TX	Puerto de transmisión de datos de la FPGA al módulo Bluetooth.
104	UART_BT_RX	Puerto de recepción de datos de la FPGA del módulo Bluetooth.

UART Shield

Este bloque es el encargado de comunicarse con el Shield de Arduino localizado en el módulo principal. Debe implementar un protocolo de comunicaciones RS232 y obtener los comandos recibidos y enviar los valores de todas las variables internas de estado. La Tabla 84 contiene los pines de la FPGA, el nombre del puerto y la descripción de la funcionalidad que desempeña.

Tabla 84. Puertos de la funcionalidad UART Shield.

Puertos del UART Shield		
Pin FPGA	Nombre del puerto	Funcionalidad
63	UART_SH_TX	Puerto de transmisión de datos de la FPGA al Shield Arduino.
64	UART_SH_RX	Puerto de recepción de datos de la FPGA del Shield Arduino.

Comunicación discreta Shield

Este bloque es el encargado de comunicarse con el Shield de Arduino localizado en el módulo principal mediante señales discretas. Hay tres entradas digitales y tres salidas digitales de propósito general. Además, este bloque podrá manejar en modo PWM los puertos del Shield de Arduino con el identificador numérico 3, 5, 6, 9, 10, y 11. La Tabla 85 contiene los pines de la FPGA, el nombre del puerto y la descripción de la funcionalidad que desempeña.

Tabla 85. Puertos de la comunicación discreta Shield.

Puertos de la comunicación discreta Shield		
Pin FPGA	Nombre del puerto	Funcionalidad
65	DISC_IN_1_SH_D2	Puerto digital D2 del Shield Arduino. Entrada discreta 1 del Shield Arduino.
67	PWM_1_SH_D3	Puerto digital D3 del Shield Arduino. Puerto PWM.
69	DISC_IN_2_SH_D4	Puerto digital D4 del Shield Arduino. Entrada discreta 2 del Shield Arduino.
70	PWM_2_SH_D5	Puerto digital D5 del Shield Arduino. Puerto PWM.
71	PWM_3_SH_D6	Puerto digital D6 del Shield Arduino. Puerto PWM.
72	DISC_IN_3_SH_D7	Puerto digital D7 del Shield Arduino. Entrada discreta 3 del Shield Arduino.
96	DISC_OUT_1_SH_D8	Puerto digital D8 del Shield Arduino. Salida discreta 1 del Shield Arduino.
97	PWM_4_SH_D9	Puerto digital D9 del Shield Arduino. Puerto PWM.
99	PWM_5_SH_D10	Puerto digital D10 del Shield Arduino. Puerto PWM.
100	PWM_6_SH_D11	Puerto digital D11 del Shield Arduino. Puerto PWM.
101	DISC_OUT_2_SH_D12	Puerto digital D12 del Shield Arduino. Salida discreta 2 del Shield Arduino.
136	DISC_OUT_3_SH_D13	Puerto digital D13 del Shield Arduino. Salida discreta 3 del Shield Arduino.

UART Arduino

Este bloque es el encargado de comunicarse con la Arduino localizado en el módulo secundario. Debe implementar un protocolo de comunicaciones RS232 y obtener los comandos recibidos y enviar los valores de todas las variables internas de estado. Este bloque se debe replicar una vez por cada módulo secundario, es decir, en total se deben

instanciar cuatro de estos módulos. La Tabla 86 contiene los pines de la FPGA, el nombre del puerto y la descripción de la funcionalidad que desempeña.

Tabla 86. Puertos de la funcionalidad UART Arduino.

Puertos del UART Arduino		
Pin FPGA	Nombre del puerto	Funcionalidad
41	UART_AR1_TX	Puerto de transmisión de datos de la FPGA a la Arduino Nano 1.
42	UART_AR1_RX	Puerto de recepción de datos de la FPGA de la Arduino Nano 1.
74	UART_AR2_TX	Puerto de transmisión de datos de la FPGA a la Arduino Nano 2.
75	UART_AR2_RX	Puerto de recepción de datos de la FPGA de la Arduino Nano 2.
113	UART_AR3_TX	Puerto de transmisión de datos de la FPGA a la Arduino Nano 3.
114	UART_AR3_RX	Puerto de recepción de datos de la FPGA de la Arduino Nano 3.
142	UART_AR4_TX	Puerto de transmisión de datos de la FPGA a la Arduino Nano 4.
143	UART_AR4_RX	Puerto de recepción de datos de la FPGA de la Arduino Nano 4.

Monitorización PWM

Este bloque es el encargado de monitorizar las salidas PWM de la Arduino, localizado en el módulo secundario. Para ello recibirá como entradas los puertos digitales de la Arduino Nano con el identificador numérico 3, 5, 6, 9, 10, y 11. Este bloque se debe replicar una vez por cada módulo secundario, es decir, en total se deben instanciar cuatro de estos módulos. La Tabla 87 contiene los pines de la FPGA, el nombre del puerto y la descripción de la funcionalidad que desempeña.

Tabla 87. Puertos de la monitorización PWM.

Puertos de la monitorización PWM		
Pin FPGA	Nombre del puerto	Funcionalidad
44	PWM_1_AR1_D3	Puerto digital D3 de la Arduino Nano 1. Puerto PWM.
47	PWM_2_AR1_D5	Puerto digital D5 de la Arduino Nano 1. Puerto PWM.
48	PWM_3_AR1_D6	Puerto digital D6 de la Arduino Nano 1. Puerto PWM.
53	PWM_4_AR1_D9	Puerto digital D9 de la Arduino Nano 1. Puerto PWM.
55	PWM_5_AR1_D10	Puerto digital D10 de la Arduino Nano 1. Puerto PWM.
57	PWM_6_AR1_D11	Puerto digital D11 de la Arduino Nano 1. Puerto PWM.
79	PWM_1_AR2_D3	Puerto digital D3 de la Arduino Nano 2. Puerto PWM.
81	PWM_2_AR2_D5	Puerto digital D5 de la Arduino Nano 2. Puerto PWM.

Puertos de la monitorización PWM		
Pin FPGA	Nombre del puerto	Funcionalidad
86	PWM_3_AR2_D6	Puerto digital D6 de la Arduino Nano 2. Puerto PWM.
89	PWM_4_AR2_D9	Puerto digital D9 de la Arduino Nano 2. Puerto PWM.
90	PWM_5_AR2_D10	Puerto digital D10 de la Arduino Nano 2. Puerto PWM.
91	PWM_6_AR2_D11	Puerto digital D11 de la Arduino Nano 2. Puerto PWM.
118	PWM_1_AR3_D3	Puerto digital D3 de la Arduino Nano 3. Puerto PWM.
120	PWM_2_AR3_D5	Puerto digital D5 de la Arduino Nano 3. Puerto PWM.
121	PWM_3_AR3_D6	Puerto digital D6 de la Arduino Nano 3. Puerto PWM.
126	PWM_4_AR3_D9	Puerto digital D9 de la Arduino Nano 3. Puerto PWM.
129	PWM_5_AR3_D10	Puerto digital D10 de la Arduino Nano 3. Puerto PWM.
132	PWM_6_AR3_D11	Puerto digital D11 de la Arduino Nano 3. Puerto PWM.
8	PWM_1_AR4_D3	Puerto digital D3 de la Arduino Nano 4. Puerto PWM.
22	PWM_2_AR4_D5	Puerto digital D5 de la Arduino Nano 4. Puerto PWM.
21	PWM_3_AR4_D6	Puerto digital D6 de la Arduino Nano 4. Puerto PWM.
27	PWM_4_AR4_D9	Puerto digital D9 de la Arduino Nano 4. Puerto PWM.
26	PWM_5_AR4_D10	Puerto digital D10 de la Arduino Nano 4. Puerto PWM.
30	PWM_6_AR4_D11	Puerto digital D11 de la Arduino Nano 4. Puerto PWM.

Comunicación discreta Arduino

Este bloque es el encargado de comunicarse con la Arduino, localizado en el módulo secundario mediante señales discretas. Hay tres entradas digitales y tres salidas digitales de propósito general. Este bloque se debe replicar una vez por cada módulo secundario, es decir, en total se deben instanciar cuatro de estos módulos. La Tabla 88 contiene los pines de la FPGA, el nombre del puerto y la descripción de la funcionalidad que desempeña.

Tabla 88. Puertos de la comunicación discreta Arduino.

Puertos de la comunicación discreta Arduino		
Pin FPGA	Nombre del puerto	Funcionalidad
43	DISC_IN_1_AR1_D2	Puerto digital D2 de la Arduino Nano 1. Entrada discreta 1 de la Arduino Nano 1.
45	DISC_IN_2_AR1_D4	Puerto digital D4 de la Arduino Nano 1. Entrada discreta 2 de la Arduino Nano 1.
51	DISC_IN_3_AR1_D7	Puerto digital D7 de la Arduino Nano 1. Entrada discreta 3 de la Arduino Nano 1.
52	DISC_OUT_1_AR1_D8	Puerto digital D8 de la Arduino Nano 1. Salida discreta 1 de la Arduino Nano 1.
58	DISC_OUT_2_AR1_D12	Puerto digital D12 de la Arduino Nano 1. Salida discreta 2 de la Arduino Nano 1.
59	DISC_OUT_3_AR1_D13	Puerto digital D13 de la Arduino Nano 1. Salida discreta 3 de la Arduino Nano 1.
76	DISC_IN_1_AR2_D2	Puerto digital D2 de la Arduino Nano 2. Entrada discreta 1 de la Arduino Nano 2.

Puertos de la comunicación discreta Arduino		
Pin FPGA	Nombre del puerto	Funcionalidad
80	DISC_IN_2_AR2_D4	Puerto digital D4 de la Arduino Nano 2. Entrada discreta 2 de la Arduino Nano 2.
87	DISC_IN_3_AR2_D7	Puerto digital D7 de la Arduino Nano 2. Entrada discreta 3 de la Arduino Nano 2.
88	DISC_OUT_1_AR2_D8	Puerto digital D8 de la Arduino Nano 2. Salida discreta 1 de la Arduino Nano 2.
92	DISC_OUT_2_AR2_D12	Puerto digital D12 de la Arduino Nano 2. Salida discreta 2 de la Arduino Nano 2.
93	DISC_OUT_3_AR2_D13	Puerto digital D13 de la Arduino Nano 2. Salida discreta 3 de la Arduino Nano 2.
115	DISC_IN_1_AR3_D2	Puerto digital D2 de la Arduino Nano 3. Entrada discreta 1 de la Arduino Nano 3.
119	DISC_IN_2_AR3_D4	Puerto digital D4 de la Arduino Nano 3. Entrada discreta 2 de la Arduino Nano 3.
122	DISC_IN_3_AR3_D7	Puerto digital D7 de la Arduino Nano 3. Entrada discreta 3 de la Arduino Nano 3.
125	DISC_OUT_1_AR3_D8	Puerto digital D8 de la Arduino Nano 3. Salida discreta 1 de la Arduino Nano 3.
133	DISC_OUT_2_AR3_D12	Puerto digital D12 de la Arduino Nano 3. Salida discreta 2 de la Arduino Nano 3.
134	DISC_OUT_3_AR3_D13	Puerto digital D13 de la Arduino Nano 3. Salida discreta 3 de la Arduino Nano 3.
4	DISC_IN_1_AR4_D2	Puerto digital D2 de la Arduino Nano 4. Entrada discreta 1 de la Arduino Nano 4.
18	DISC_IN_2_AR4_D4	Puerto digital D4 de la Arduino Nano 4. Entrada discreta 2 de la Arduino Nano 4.
25	DISC_IN_3_AR4_D7	Puerto digital D7 de la Arduino Nano 4. Entrada discreta 3 de la Arduino Nano 4.
24	DISC_OUT_1_AR4_D8	Puerto digital D8 de la Arduino Nano 4. Salida discreta 1 de la Arduino Nano 4.
28	DISC_OUT_2_AR4_D12	Puerto digital D12 de la Arduino Nano 4. Salida discreta 2 de la Arduino Nano 4.
32	DISC_OUT_3_AR4_D13	Puerto digital D13 de la Arduino Nano 4. Salida discreta 3 de la Arduino Nano 4.

7.1.3. PASO 3: DESCRIPCIÓN FIRMWARE

Tras la elaboración de las especificaciones del firmware y el detalle de los bloques funcionales necesarios, se avanza hasta el tercer paso. En este paso se realiza la descripción hardware. Para ello, se realiza una metodología de descripción Top-Down [Volnei, 2014]. Es decir, se realiza la descripción de alto nivel, y a continuación se va describiendo el firmware del siguiente nivel jerárquico inferior. Para cumplir con las especificaciones del firmware, se van a usar sólo tres niveles jerárquicos. El primer nivel jerárquico es el de más alto nivel, el segundo nivel jerárquico se usa para los bloques funcionales. Por último, el más bajo nivel jerárquico es el utilizado para las funcionalidades más sencillas,

y que podrían usarse como referencia o base para otros bloques funcionales o para otros diseños firmware ajenos al ámbito de aplicación de esta tesis doctoral.

La Figura 98 muestra el diseño jerárquico Top-Down indicando cada uno de los componentes y el nivel jerárquico en el que están.

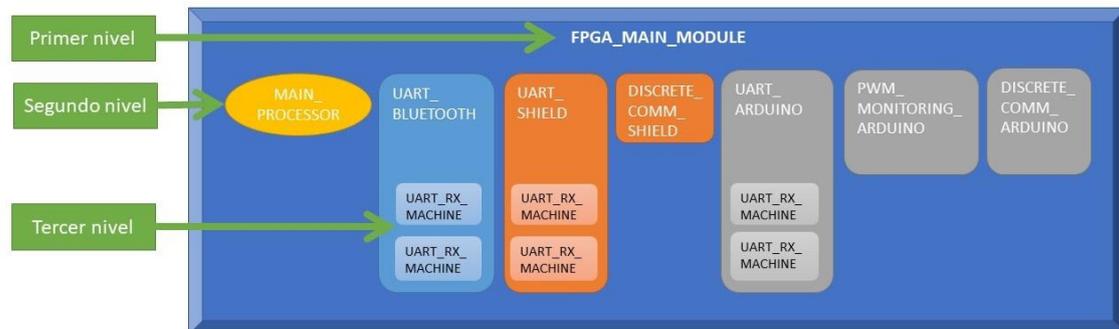


Figura 98. Diseño jerárquico Top-Down.

FPGA_MAIN_MODULE

Este bloque será el encargado de gestionar los puertos de la FPGA y conectarlos con los diferentes bloques funcionales descritos a más bajo nivel. Además, en este nivel se definen las instanciaciones de los diferentes bloques funcionales y las conexiones entre ellos. Para poder gestionar los datos que se usan en los diferentes bloques funcionales y el control de las señales se usa un bloque principal de procesamiento, el bloque MAIN_PROCESSOR. De manera que, los bloques a instanciar son los siguientes:

- MAIN_PROCESSOR.
- UART_BLUETOOTH:
 - UART_RX_MACHINE.
 - UART_TX_MACHINE.
- UART_SHIELD:
 - UART_RX_MACHINE.
 - UART_TX_MACHINE.
- DISCRETE_COMM_SHIELD.
- UART_ARDUINO:
 - UART_RX_MACHINE.
 - UART_TX_MACHINE.
- PWM_MONITORING_ARDUINO.
- DISCRETE_COMM_ARDUINO.

La Figura 99 representa el diagrama de bloques del sistema completo. Asimismo, se han incluido las líneas de comunicaciones entre los bloques internos de la FPGA. Además, este diagrama contiene las comunicaciones entre el hardware externo a la FPGA y los bloques internos de la FPGA.

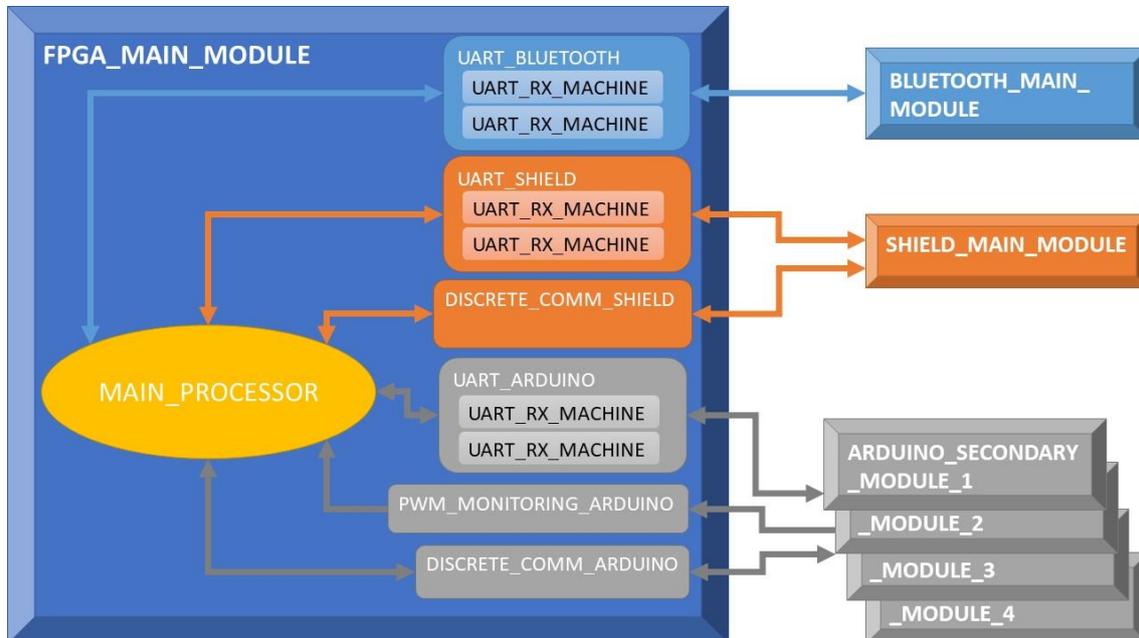


Figura 99. Diagrama de bloques del sistema completo.

El bloque FPGA_MAIN_MODULE usa las señales definidas en la Tabla 77, Tabla 78, Tabla 79, Tabla 80, Tabla 81 y Tabla 82 para las comunicaciones entre el firmware y los pines de la FPGA. También usa las señales definidas en las Tablas Tabla 83, Tabla 84, Tabla 85, Tabla 86, Tabla 87 y Tabla 88 para implementar las conexiones entre los diferentes bloques funcionales.

MAIN_PROCESSOR

Este bloque se encarga del procesamiento general dentro de la FPGA. Esto implica gestionar las comunicaciones UART de los diferentes bloques que hay en la FPGA, y de recopilar los datos procesados por los otros bloques. El bloque MAIN_PROCESSOR tiene las siguientes funcionalidades:

- Comunicaciones serie: procesa todas las comunicaciones entrantes. Es decir, los bloques UART_BLUETOOTH, UART_SHIELD y UART_ARDUINO le transmitirán los mensajes recibidos. Para ello, se definen las correspondientes señales de datos recibidos para cada uno de los bloques. Además, el bloque MAIN_PROCESSOR les indicará qué mensaje mandar y cuándo deben hacerlo cada uno de los bloques de comunicación UART. Por tanto, se definen las correspondientes señales

de datos enviados para cada uno de los bloques. También se tienen en cuenta las señales de datos recibidos y de habilitación de envío de datos.

- Comunicación con UART_BLUETOOTH: El bloque MAIN_PROCESSOR espera a recibir una trama del bloque UART_BLUETOOTH, a la cual contestará con la trama “HeCRE ready”. Después enviará el estado de las comunicaciones serie de las Arduino Nano (UART_ARDUINO) y el estado de las señales discretas (DISCRETE_COMM_ARDUINO) y señales PWM de las Arduino Nano (PWM_MONITORING_ARDUINO). La Figura 100 muestra un ejemplo de comunicación UART_BLUETOOTH donde se detalla un diagrama de secuencia de mensajes.
- Comunicación con UART_SHIELD y UART_ARDUINO: El bloque MAIN_PROCESSOR espera a recibir una trama de los bloques UART_SHIELD y UART_ARDUINO, a la cual contestará con la trama “Message received”. Después almacenará el carácter recibido.
- Señales discretas: procesa todas las señales discretas recibidas y gestiona las señales discretas enviadas. Es decir, los bloques DISCRETE_COMM_SHIELD y DISCRETE_COMM_ARDUINO le indicarán cuando ha habido un cambio en el nivel de las señales discretas que manejan. El bloque MAIN_PROCESSOR les indicará cuándo cambiar de nivel la salida correspondiente.
- Señales PWM: procesa todos los pulsos PWM recibidos por bloques PWM_MONITORING_ARDUINO instanciados para cada uno de los módulos secundarios. Los bloques PWM_MONITORING_ARDUINO le indicarán los milisegundos que han transcurrido en cada nivel (alto y bajo) para cada periodo registrado.

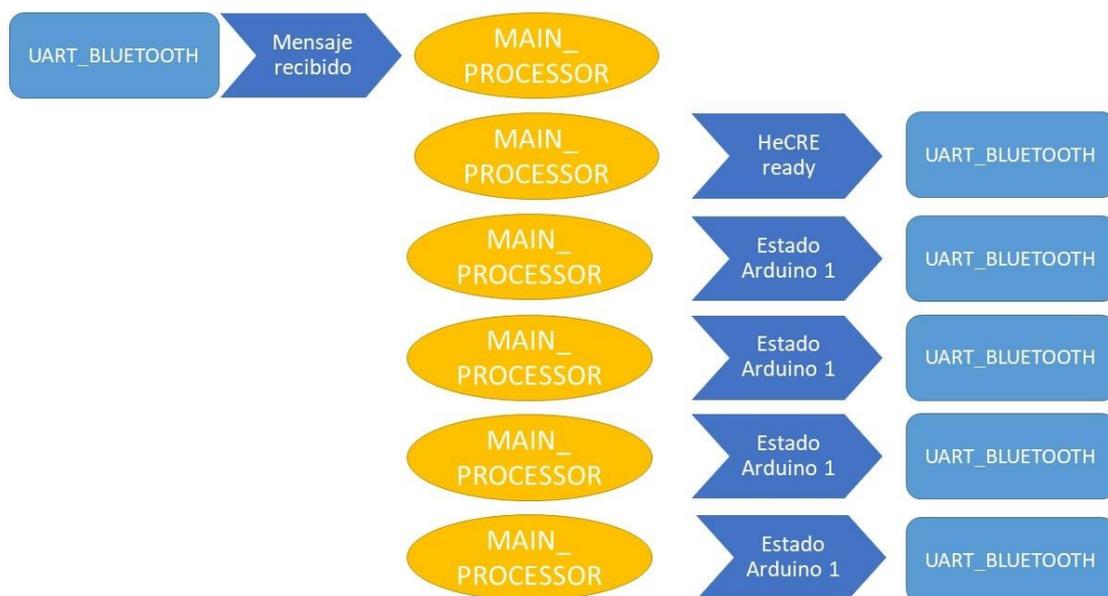


Figura 100. Diagrama de secuencia para la comunicación con UART_BLUETOOTH.

La Tabla 89 contiene las señales de entrada/salida del bloque MAIN_PROCESSOR, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 89. Señales de entrada/salida del bloque MAIN_PROCESSOR.

Señales de entrada/salida del bloque MAIN_PROCESSOR			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_BT_IN_FRAME	Entrada 8 bits	UART_BLUETOOTH	Datos recibidos por el bloque UART Bluetooth.
UART_BT_IN_RECEIVED	Entrada 1 bit	UART_BLUETOOTH	Nuevo dato recibido por el bloque UART Bluetooth.
UART_BT_OUT_FRAME	Salida 8 bits	UART_BLUETOOTH	Datos que enviar por el bloque UART Bluetooth.
UART_BT_OUT_SEND	Salida 1 bit	UART_BLUETOOTH	Orden para enviar datos por el bloque UART Bluetooth.
UART_SH_IN_FRAME	Entrada 8 bits	UART_SHIELD	Datos recibidos por el bloque UART Shield.
UART_SH_IN_RECEIVED	Entrada 1 bit	UART_SHIELD	Nuevo dato recibido por el bloque UART Shield.
UART_SH_OUT_FRAME	Salida 8 bits	UART_SHIELD	Datos que enviar por el bloque UART Shield.
UART_SH_OUT_SEND	Salida 1 bit	UART_SHIELD	Orden para enviar datos por el bloque UART Shield.
UART_AR_IN_FRAME(i)	Entrada 8 bits 4 señales	UART_ARDUINO	Datos recibidos por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.
UART_AR_IN_RECEIVED(i)	Entrada 1 bit 4 señales	UART_ARDUINO	Nuevo dato recibido por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.
UART_AR_OUT_FRAME(i)	Salida 8 bits 4 señales	UART_ARDUINO	Datos que enviar por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.
UART_AR_OUT_SEND(i)	Salida 1 bit 4 señales	UART_ARDUINO	Orden para enviar datos por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.
DISC_IN_SH_EDGE	Entrada 3 bits	DISCRETE_COMM_SHIELD	Cambio de nivel detectado en la señal correspondiente de las señales discretas del Shield de Arduino.
DISC_OUT_SH_LEVEL	Salida 3 bits	DISCRETE_COMM_SHIELD	Nivel ordenado para la señal correspondiente de las señales discretas del Shield de Arduino.
PWM_AR_PULSE_HI	Entrada 8 bits 24 señales	PWM_MONITORING_ARDUINO	Milisegundos que está la señal PWM a nivel alto. Hay seis señales de este tipo para cada módulo secundario.
PWM_AR_PULSE_LO	Salida 8 bits 24 señales	PWM_MONITORING_ARDUINO	Milisegundos que está la señal PWM a nivel bajo. Hay seis señales de este tipo para cada módulo secundario.

Señales de entrada/salida del bloque MAIN_PROCESSOR			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
DISC_IN_AR_EDGE(i)	Entrada 3 bits 4 señales	DISCRETE_COMM_AR DUINO	Cambio de nivel detectado en la señal correspondiente de las señales discretas de módulo secundario.
DISC_OUT_AR_LEVEL(i)	Salida 3 bits 4 señales	DISCRETE_COMM_AR DUINO	Nivel ordenado para la señal correspondiente de las señales discretas de módulo secundario.

El archivo fuente que implementa las mencionadas funcionalidades se llama MAIN_PROCESSOR.vhd.

UART_BLUETOOTH

Este bloque implementa las comunicaciones UART entre la FPGA y el módulo Bluetooth que hay en el módulo principal. El bloque UART_BLUETOOTH tiene las siguientes funcionalidades:

- Comunicaciones serie: implementa las comunicaciones con el módulo Bluetooth del módulo principal. En este bloque se conectan las señales del bloque UART_BLUETOOTH y se instancian los dos sub-bloques: UART_RX_MACHINE y UART_TX_MACHINE. UART_BLUETOOTH se encarga de gestionar la transmisión y recepción en función de los datos que sea necesario transmitir y de los datos que se hayan recibido. El sub-bloque UART_TX_MACHINE maneja las señales que se conectan directamente a los pines de la FPGA para la transmisión de datos. El sub-bloque UART_RX_MACHINE maneja las señales que se conectan directamente a los pines de la FPGA para la recepción de datos. El bloque UART_BLUETOOTH informará al bloque MAIN_PROCESSOR cuando haya llegado una nueva trama. También le pasará dicha trama. Además, el bloque UART_BLUETOOTH enviará la trama que le pase el bloque MAIN_PROCESSOR cuando este bloque se lo indique.

La Tabla 90 contiene las señales de entrada/salida del bloque UART_BLUETOOTH, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 90. Señales de entrada/salida del bloque UART_BLUEETOOTH.

Señales de entrada/salida del bloque UART_BLUEETOOTH			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_BT_TX	Salida 1 bit	FPGA_MAIN_MODULE	Puerto de transmisión de datos de la FPGA al módulo Bluetooth.
UART_BT_RX	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto de recepción de datos de la FPGA del módulo Bluetooth.
UART_BT_IN_FRAME	Salida 8 bits	MAIN_PROCESSOR	Datos recibidos por el bloque UART Bluetooth.
UART_BT_IN_RECEIVED	Salida 1 bit	MAIN_PROCESSOR	Nuevo dato recibido por el bloque UART Bluetooth.
UART_BT_OUT_FRAME	Entrada 8 bits	MAIN_PROCESSOR	Datos que enviar por el bloque UART Bluetooth.
UART_BT_OUT_SEND	Entrada 1 bit	MAIN_PROCESSOR	Orden para enviar datos por el bloque UART Bluetooth.

El archivo fuente que implementa las mencionadas funcionalidades se llama UART_BLUEETOOTH.vhd.

UART_SHIELD

Este bloque implementa las comunicaciones UART entre la FPGA y el Shield Arduino que hay en el módulo principal. El bloque UART_SHIELD tiene las siguientes funcionalidades:

- Comunicaciones serie: implementa las comunicaciones con el Shield Arduino del módulo principal. En este bloque se conectan las señales del bloque UART_SHIELD y se instancian los dos sub-bloques: UART_RX_MACHINE y UART_TX_MACHINE. UART_SHIELD se encarga de gestionar la transmisión y recepción en función de los datos que sea necesario transmitir y de los datos que se hayan recibido. El sub-bloque UART_TX_MACHINE maneja las señales que se conectan directamente a los pines de la FPGA para la transmisión de datos. El sub-bloque UART_RX_MACHINE maneja las señales que se conectan directamente a los pines de la FPGA para la recepción de datos. El bloque UART_SHIELD informará al bloque MAIN_PROCESSOR cuando haya llegado una nueva trama. También le pasará dicha trama. Además, el bloque UART_SHIELD enviará la trama que le pase el bloque MAIN_PROCESSOR cuando este bloque se lo indique.

La Tabla 91 contiene las señales de entrada/salida del bloque UART_SHIELD, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 91. Señales de entrada/salida del bloque UART_SHIELD.

Señales de entrada/salida del bloque UART_SHIELD			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_SH_TX	Salida 1 bit	FPGA_MAIN_MODULE	Puerto de transmisión de datos de la FPGA al Shield Arduino.
UART_SH_RX	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto de recepción de datos de la FPGA del Shield Arduino.
UART_SH_IN_FRAME	Salida 8 bits	MAIN_PROCESSOR	Datos recibidos por el bloque UART Shield.
UART_SH_IN_RECEIVED	Salida 1 bit	MAIN_PROCESSOR	Nuevo dato recibido por el bloque UART Shield.
UART_SH_OUT_FRAME	Entrada 8 bits	MAIN_PROCESSOR	Datos que enviar por el bloque UART Shield.
UART_SH_OUT_SEND	Entrada 1 bit	MAIN_PROCESSOR	Orden para enviar datos por el bloque UART Shield.

El archivo fuente que implementa las mencionadas funcionalidades se llama UART_SHIELD.vhd.

DISCRETE_COMM_SHIELD

Este bloque se encarga de monitorizar las señales discretas del Shield conectado al módulo principal para posteriormente enviar la información procesada al bloque MAIN_PROCESSOR. El bloque DISCRETE_COMM_SHIELD tiene las siguientes funcionalidades:

- Señales discretas: indica al bloque MAIN_PROCESSOR cuando ha habido cambios en las señales discretas recibidas. Además, el bloque MAIN_PROCESSOR indicará cuándo cambiar de nivel la salida correspondiente al bloque DISCRETE_COMM_SHIELD.

La Tabla 92 contiene las señales de entrada/salida del bloque DISCRETE_COMM_SHIELD, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 92. Señales de entrada/salida del bloque DISCRETE_COMM_SHIELD.

Señales de entrada/salida del bloque DISCRETE_COMM_SHIELD			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
DISC_IN_1_SH_D2	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D2 del Shield Arduino. Entrada discreta 1 del Shield Arduino.
PWM_1_SH_D3	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D3 del Shield Arduino. Puerto PWM.

Señales de entrada/salida del bloque DISCRETE_COMM_SHIELD			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
DISC_IN_2_SH_D4	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D4 del Shield Arduino. Entrada discreta 2 del Shield Arduino.
PWM_2_SH_D5	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D5 del Shield Arduino. Puerto PWM.
PWM_3_SH_D6	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D6 del Shield Arduino. Puerto PWM.
DISC_IN_3_SH_D7	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D7 del Shield Arduino. Entrada discreta 3 del Shield Arduino.
DISC_OUT_1_SH_D8	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D8 del Shield Arduino. Salida discreta 1 del Shield Arduino.
PWM_4_SH_D9	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D9 del Shield Arduino. Puerto PWM.
PWM_5_SH_D10	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D10 del Shield Arduino. Puerto PWM.
PWM_6_SH_D11	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D11 del Shield Arduino. Puerto PWM.
DISC_OUT_2_SH_D12	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D12 del Shield Arduino. Salida discreta 2 del Shield Arduino.
DISC_OUT_3_SH_D13	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D13 del Shield Arduino. Salida discreta 3 del Shield Arduino.
DISC_IN_SH_EDGE	Salida 1 bit	MAIN_PROCESSOR	Cambio de nivel detectado en la señal correspondiente de las señales discretas del Shield de Arduino.
DISC_OUT_SH_LEVEL	Entrada 9 bit	MAIN_PROCESSOR	Nivel ordenado para la señal correspondiente de las señales discretas del Shield de Arduino.

El archivo fuente que implementa las mencionadas funcionalidades se llama DISCRETE_COMM_SHIELD.vhd.

UART_ARDUINO

Este bloque implementa las comunicaciones UART entre la FPGA y la Arduino que hay en el módulo secundario. El bloque UART_ARDUINO tiene las siguientes funcionalidades:

- Comunicaciones serie: implementa las comunicaciones con las Arduino de los módulos secundarios. En este bloque se conectan las señales del bloque UART_SHIELD y se instancian los dos sub-bloques: UART_RX_MACHINE y UART_TX_MACHINE. UART_MAIN_CONTROL se encarga de gestionar la transmisión y recepción en función de los datos que sea necesario transmitir y de los datos que se hayan recibido. El sub-bloque UART_TX_MACHINE maneja las señales que se conectan directamente a los pines de la FPGA para la transmisión de datos. El

sub-bloque UART_RX_MACHINE maneja las señales que se conectan directamente a los pines de la FPGA para la recepción de datos. El bloque UART_ARDUINO informará al bloque MAIN_PROCESSOR cuando haya llegado una nueva trama. También le pasará dicha trama. Además, el bloque UART_ARDUINO enviará la trama que le pase el bloque MAIN_PROCESSOR cuando este bloque se lo indique.

La Tabla 93 contiene las señales de entrada/salida del bloque UART_ARDUINO, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 93. Señales de entrada/salida del bloque UART_ARDUINO.

Señales de entrada/salida del bloque UART_ARDUINO			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_AR_TX	Salida 1 bit	FPGA_MAIN_MODULE	Puerto de transmisión de datos de la FPGA a la Arduino Nano.
UART_AR_RX	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto de recepción de datos de la FPGA de la Arduino Nano.
UART_AR_IN_FRAME	Salida 8 bits	MAIN_PROCESSOR	Datos recibidos por el bloque UART Arduino.
UART_AR_IN_RECEIVED	Salida 1 bit	MAIN_PROCESSOR	Nuevo dato recibido por el bloque UART Arduino.
UART_AR_OUT_FRAME	Entrada 8 bits	MAIN_PROCESSOR	Datos que enviar por el bloque UART Arduino.
UART_AR_OUT_SEND	Entrada 1 bit	MAIN_PROCESSOR	Orden para enviar datos por el bloque UART Arduino.

El archivo fuente que implementa las mencionadas funcionalidades se llama UART_ARDUINO.vhd.

PWM_MONITORING_ARDUINO

Este bloque se encarga de monitorizar las señales PWN de la Arduino conectado al módulo secundario para posteriormente enviar la información procesada al bloque MAIN_PROCESSOR. El bloque PWM_MONITORING_ARDUINO tiene las siguientes funcionalidades:

- Señales PWM: indica el ancho de los pulsos PWM recibidos al bloque MAIN_PROCESSOR, los milisegundos que han transcurrido en cada nivel (alto y bajo) para cada periodo registrado.

La Tabla 94 contiene las señales de entrada/salida del bloque PWM_MONITORING_ARDUINO, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 94. Señales de entrada/salida del bloque PWM_MONITORING_ARDUINO.

Señales de entrada/salida del bloque PWM_MONITORING_ARDUINO			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
PWM_1_AR_D3	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D3 de la Arduino Nano. Puerto PWM.
PWM_2_AR_D5	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D5 de la Arduino Nano. Puerto PWM.
PWM_3_AR_D6	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D6 de la Arduino Nano. Puerto PWM.
PWM_4_AR_D9	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D9 de la Arduino Nano. Puerto PWM.
PWM_5_AR_D10	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D10 de la Arduino Nano. Puerto PWM.
PWM_6_AR_D11	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D11 de la Arduino Nano. Puerto PWM.
PWM_AR_PULSE_HI	Salida 8 bit 6 señales	MAIN_PROCESSOR	Milisegundos que está la señal PWM a nivel alto. Hay seis señales de este tipo para cada módulo secundario.
PWM_AR_PULSE_LO	Salida 8 bit 6 señales	MAIN_PROCESSOR	Milisegundos que está la señal PWM a nivel bajo. Hay seis señales de este tipo para cada módulo secundario.

El archivo fuente que implementa las mencionadas funcionalidades se llama PWM_MONITORING_ARDUINO.vhd.

DISCRETE_COMM_ARDUINO

Este bloque se encarga de monitorizar las señales discretas de la Arduino conectado al módulo secundario para posteriormente enviar la información procesada al bloque MAIN_PROCESSOR. El bloque DISCRETE_COMM_ARDUINO tiene las siguientes funcionalidades:

- Señales discretas: indica al bloque MAIN_PROCESSOR cuando ha habido cambios en las señales discretas recibidas. Además, el bloque MAIN_PROCESSOR indicará cuándo cambiar de nivel la salida correspondiente al bloque DISCRETE_COMM_ARDUINO.

La Tabla 95 contiene las señales de entrada/salida del bloque DISCRETE_COMM_ARDUINO, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 95. Señales de entrada/salida del bloque DISCRETE_COMM_ARDUINO.

Señales de entrada/salida del bloque DISCRETE_COMM_ARDUINO			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
DISC_IN_1_AR_D2	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D2 de la Arduino Nano. Entrada discreta 1 de la Arduino Nano.
DISC_IN_2_AR_D4	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D4 de la Arduino Nano. Entrada discreta 2 de la Arduino Nano.
DISC_IN_3_AR_D7	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto digital D7 de la Arduino Nano. Entrada discreta 3 de la Arduino Nano.
DISC_OUT_AR_LEVEL	Entrada 3 bits	MAIN_PROCESSOR	Nivel ordenado para la señal correspondiente de las señales discretas de módulo secundario.
DISC_OUT_1_AR_D8	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D8 de la Arduino Nano. Salida discreta 1 de la Arduino Nano.
DISC_OUT_2_AR_D12	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D12 de la Arduino Nano. Salida discreta 2 de la Arduino Nano.
DISC_OUT_3_AR_D13	Salida 1 bit	FPGA_MAIN_MODULE	Puerto digital D13 de la Arduino Nano. Salida discreta 3 de la Arduino Nano.
DISC_IN_AR_EDGE	Salida 3 bits	MAIN_PROCESSOR	Cambio de nivel detectado en la señal correspondiente de las señales discretas de módulo secundario.

El archivo fuente que implementa las mencionadas funcionalidades se llama DISCRETE_COMM_ARDUINO.vhd.

UART_RX_MACHINE

Este bloque se encarga de recibir mensajes a través del UART. El bloque UART_RX_MACHINE tiene las siguientes funcionalidades:

- Comunicaciones serie: implementa las comunicaciones con el puerto serie. El sub-bloque UART_RX_MACHINE maneja la señal que se conecta directamente al pin de la FPGA para la recepción de datos.

La Tabla 96 contiene las señales de entrada/salida del bloque UART_RX_MACHINE, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 96. Señales de entrada/salida del bloque UART_RX_MACHINE.

Señales de entrada/salida del bloque UART_RX_MACHINE			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_RX	Entrada 1 bit	FPGA_MAIN_MODULE	Puerto de recepción de datos de la FPGA al Puerto Serie.
UART_IN_FRAME	Salida 8 bits	UART_BLUETOOTH, UART_SHIELD, UART_ARDUINO	Datos recibidos por el bloque UART.
UART_IN_RECEIVED	Salida 1 bit	UART_BLUETOOTH, UART_SHIELD, UART_ARDUINO	Nuevo dato recibido por el bloque UART.

El archivo fuente que implementa las mencionadas funcionalidades se llama UART_RX_MACHINE.vhd.

UART_TX_MACHINE

Este bloque se encarga de enviar mensajes a través del UART. El bloque UART_TX_MACHINE tiene las siguientes funcionalidades:

- Comunicaciones serie: implementa las comunicaciones con el puerto serie. El sub-bloque UART_TX_MACHINE maneja la señal que se conecta directamente al pin de la FPGA para la transmisión de datos.

La Tabla 97 contiene las señales de entrada/salida del bloque UART_TX_MACHINE, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 97. Señales de entrada/salida del bloque UART_TX_MACHINE.

Señales de entrada/salida del bloque UART_TX_MACHINE			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_OUT_FRAME	Entrada 8 bits	UART_BLUETOOTH, UART_SHIELD, UART_ARDUINO	Datos que enviar por el bloque UART.
UART_OUT_SEND	Entrada 1 bit	UART_BLUETOOTH, UART_SHIELD, UART_ARDUINO	Orden para enviar datos por el bloque UART.

Señales de entrada/salida del bloque UART_TX_MACHINE			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_TX	Salida 1 bits	FPGA_MAIN_MODULE	Puerto de transmisión de datos de la FPGA al Puerto Serie.

El archivo fuente que implementa las mencionadas funcionalidades se llama UART_TX_MACHINE.vhd.

7.1.4. PASO 4: SIMULACIÓN

Una vez completado el paso de implementación, se realiza la verificación por simulación. En este paso se usa una metodología de verificación down-to-top [Volnei, 2014]. Es decir, se realiza la verificación de bajo nivel, y a continuación se va verificando el firmware del siguiente nivel jerárquico superior. Para comprobar que se cumplen las especificaciones del firmware, se van a realizar simulaciones de los tres niveles jerárquicos. El primer nivel jerárquico es el de más bajo nivel, el segundo nivel jerárquico se usa para los bloques funcionales. Por último, el más alto nivel jerárquico es el que incluye todas las funcionalidades. Las simulaciones sólo cubren el ámbito de aplicación de esta tesis doctoral. La Figura 101 muestra verificación jerárquica Down-Top indicando cada uno de los componentes y el nivel jerárquico en el que están.

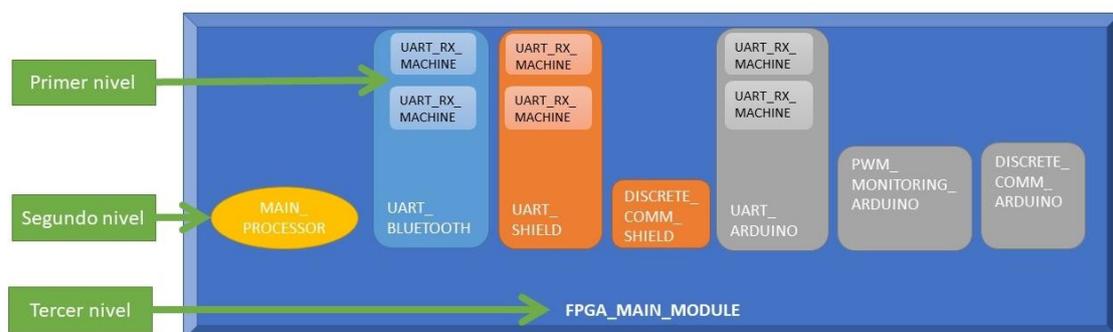


Figura 101. Verificación jerárquica Down-Top.

Las simulaciones usarán un archivo identificado como Test Bench y un script. El Test Bench incluirá todas las pruebas necesarias para verificar que se cumplen las especificaciones para cada uno de los bloques. Además, se incluye una prueba del valor de reset de cada una de las salidas del bloque. También se incluye en el Test Bench la generación de un informe de resultados para las pruebas ejecutadas por el Test Bench. El script de simulación incluye los comandos de compilación, iniciar la simulación, incluir las señales que se visualizan en las formas de onda, las instrucciones para adquirir la hora del sis-

tema y la detección del fin de la simulación. La Figura 102 muestra el proceso de verificación, los tipos de elementos que se usan en cada paso y los resultados que se obtienen cada vez que se verifica cada uno de los bloques obtenidos durante la etapa de diseño de firmware.



Figura 102. Diagrama de flujo del proceso de verificación.

UART_RX_MACHINE

En el bloque UART_RX_MACHINE se prueban la siguiente funcionalidad:

- Se mandan tramas con valores desde x“00” hasta x“FF” a través del puerto de entrada UART_RX. Posteriormente se espera a que el puerto UART_IN_RECEIVED tome un valor de ‘1’, el bloque indica que ha recibido una trama, y se comprueba que el dato que ha recibido (UART_IN_FRAME) coincide con la trama que se le ha enviado.

Se usa un UART_RX_MACHINE_TB.do para ejecutar la simulación. El archivo de simulación que implementa las mencionadas funcionalidades se llama UART_RX_MACHINE_TB.vhd. El informe de resultados es el archivo UART_RX_MACHINE_RESULTS.txt.

UART_TX_MACHINE

En el bloque UART_TX_MACHINE se prueban la siguiente funcionalidad:

- Se mandan tramas con valores desde x“00” hasta x“FF” a través del puerto de entrada UART_OUT_FRAME y se indica que mande una trama poniendo activo el puerto de entrada UART_OUT_SEND. Posteriormente se leen los datos que envía a través del puerto de salida UART_TX, y se comprueba que el dato que ha enviado coincide con la trama que se le ha indicado que envíe. También se comprueba que fije el valor de inicio y de fin de transmisión correcta.

Se usa un UART_TX_MACHINE_TB.do para ejecutar la simulación. El archivo de simulación que implementa las mencionadas funcionalidades se llama UART_TX_MACHINE_TB.vhd. El informe de resultados es el archivo UART_TX_MACHINE_RESULTS.txt.

UART_BLUETOOTH

En el bloque UART_BLUETOOTH se prueban las siguientes funcionalidades:

- Se conecta el puerto de transmisión UART_BT_TX con el puerto de recepción UART_BT_RX. De esta manera los datos que envía por el puerto de transmisión los recibe por el puerto de recepción.
- Se mandan tramas con valores desde x"00" hasta x"FF" a través del puerto de entrada UART_BT_OUT_FRAME y se indica que mande una trama poniendo activo el puerto de entrada UART_BT_OUT_SEND. Posteriormente se espera a que el puerto UART_BT_IN_RECEIVED tome un valor de '1', el bloque indica que ha recibido una trama, y se comprueba que el dato que ha recibido (UART_BT_IN_FRAME) coincide con la trama que se le ha enviado.

Se usa un UART_BLUETOOTH_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama UART_BLUETOOTH_TB.vhd. El informe de resultados es el archivo UART_BLUETOOTH_RESULTS.txt.

UART_SHIELD

En el bloque UART_SHIELD se prueban las siguientes funcionalidades:

- Se conecta el puerto de transmisión UART_SH_TX con el puerto de recepción UART_SH_RX. De esta manera los datos que envía por el puerto de transmisión los recibe por el puerto de recepción.
- Se mandan tramas con valores desde x"00" hasta x"FF" a través del puerto de entrada UART_SH_OUT_FRAME y se indica que mande una trama poniendo activo el puerto de entrada UART_SH_OUT_SEND. Posteriormente se espera a que el puerto UART_SH_IN_RECEIVED tome un valor de '1', el bloque indica que ha recibido una trama, y se comprueba que el dato que ha recibido (UART_SH_IN_FRAME) coincide con la trama que se le ha enviado.

Se usa un UART_SHIELD_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama UART_SHIELD_TB.vhd. El informe de resultados es el archivo UART_SHIELD_RESULTS.txt.

UART_ARDUINO

En el bloque UART_ARDUINO se prueban las siguientes funcionalidades:

- Se conecta el puerto de transmisión UART_AR_TX con el puerto de recepción UART_AR_RX. De esta manera los datos que envía por el puerto de transmisión los recibe por el puerto de recepción.

- Se mandan tramas con valores desde x"00" hasta x"FF" a través del puerto de entrada UART_AR_OUT_FRAME y se indica que mande una trama poniendo activo el puerto de entrada UART_AR_OUT_SEND. Posteriormente se espera a que el puerto UART_AR_IN_RECEIVED tome un valor '1', el bloque indica que ha recibido una trama, y se comprueba que el dato que ha recibido (UART_AR_IN_FRAME) coincide con la trama que se le ha enviado.

Se usa un UART_ARDUINO_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama UART_ARDUINO_TB.vhd. El informe de resultados es el archivo UART_ARDUINO_RESULTS.txt.

DISCRETE_COMM_SHIELD

En el bloque DISCRETE_COMM_SHIELD se prueban las siguientes funcionalidades:

- Se prueban las entradas discretas (DISC_IN_1_SH_D2, DISC_IN_2_SH_D4 y DISC_IN_3_SH_D7) generando un pulso a nivel activo durante un ciclo de reloj y se espera a que se ponga a '1' el puerto de salida DISC_IN_SH_EDGE correspondiente.
- Se prueban los comandos de las salidas discretas DISC_OUT_SH_LEVEL fijando su valor a nivel activo y se comprueba a que se ponga a '1' el puerto de salida correspondiente (PWM_1_SH_D3, PWM_2_SH_D5, PWM_3_SH_D6, DISC_OUT_1_SH_D8, DISC_OUT_2_SH_D12, DISC_OUT_3_SH_D13, PWM_4_SH_D9, PWM_5_SH_D10 y PWM_6_SH_D11).
- Se prueban los comandos de las salidas discretas DISC_OUT_AR_LEVEL fijando su valor a nivel inactivo y se comprueba a que se ponga a '0' el puerto de salida correspondiente (PWM_1_SH_D3, PWM_2_SH_D5, PWM_3_SH_D6, DISC_OUT_1_SH_D8, DISC_OUT_2_SH_D12, DISC_OUT_3_SH_D13, PWM_4_SH_D9, PWM_5_SH_D10 y PWM_6_SH_D11).

Se usa un DISCRETE_COMM_SHIELD_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama DISCRETE_COMM_SHIELD_TB.vhd. El informe de resultados es el archivo DISCRETE_COMM_SHIELD_RESULTS.txt.

PWM_MONITORING_ARDUINO

En el bloque PWM_MONITORING_ARDUINO se prueban las siguientes funcionalidades:

- Se genera un pulso de 5 ms a nivel alto y un pulso de 5 ms a nivel bajo en el puerto de entrada PWM_1_AR_D3. Se comprueban que los valores obtenidos en los puertos de salida PWM_AR_PULSE_HI(0) y PWM_AR_PULSE_LO(0) coinciden con los pulsos introducidos.

- Se genera un pulso de 6 ms a nivel alto y un pulso de 6 ms a nivel bajo en el puerto de entrada PWM_2_AR_D5. Se comprueban que los valores obtenidos en los puertos de salida PWM_AR_PULSE_HI(1) y PWM_AR_PULSE_LO(1) coinciden con los pulsos introducidos.
- Se genera un pulso de 7 ms a nivel alto y un pulso de 7 ms a nivel bajo en el puerto de entrada PWM_3_AR_D6. Se comprueban que los valores obtenidos en los puertos de salida PWM_AR_PULSE_HI(2) y PWM_AR_PULSE_LO(2) coinciden con los pulsos introducidos.
- Se genera un pulso de 8 ms a nivel alto y un pulso de 8 ms a nivel bajo en el puerto de entrada PWM_4_AR_D9. Se comprueban que los valores obtenidos en los puertos de salida PWM_AR_PULSE_HI(3) y PWM_AR_PULSE_LO(3) coinciden con los pulsos introducidos.
- Se genera un pulso de 9 ms a nivel alto y un pulso de 9 ms a nivel bajo en el puerto de entrada PWM_5_AR_D10. Se comprueban que los valores obtenidos en los puertos de salida PWM_AR_PULSE_HI(4) y PWM_AR_PULSE_LO(4) coinciden con los pulsos introducidos.
- Se genera un pulso de 10 ms a nivel alto y un pulso de 10 ms a nivel bajo en el puerto de entrada PWM_6_AR_D11. Se comprueban que los valores obtenidos en los puertos de salida PWM_AR_PULSE_HI(5) y PWM_AR_PULSE_LO(5) coinciden con los pulsos introducidos.

Se usa un UART_RX_MACHINE_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama PWM_MONITORING_ARDUINO_TB.vhd. El informe de resultados es el archivo PWM_MONITORING_ARDUINO_RESULTS.txt.

DISCRETE_COMM_ARDUINO

En el bloque DISCRETE_COMM_ARDUINO se prueban las siguientes funcionalidades:

- Se prueban las entradas discretas (DISC_IN_1_AR_D2, DISC_IN_2_AR_D4 y DISC_IN_3_AR_D7) generando un pulso a nivel activo durante un ciclo de reloj y se espera a que se ponga a '1' el puerto de salida DISC_IN_AR_EDGE correspondiente.
- Se prueban los comandos de las salidas discretas DISC_OUT_AR_LEVEL fijando su valor a nivel activo y se comprueba a que se ponga a '1' el puerto de salida correspondiente (DISC_OUT_1_AR_D8, DISC_OUT_2_AR_D12 y DISC_OUT_3_AR_D13).
- Se prueban los comandos de las salidas discretas DISC_OUT_AR_LEVEL fijando su valor a nivel inactivo y se comprueba a que se ponga a '0' el puerto de salida correspondiente (DISC_OUT_1_AR_D8, DISC_OUT_2_AR_D12 y DISC_OUT_3_AR_D13).

Se usa un DISCRETE_COMM_ARDUINO_TB.do para ejecutar la simulación. El archivo de simulación que implementa las mencionadas funcionalidades se llama DISCRETE_COMM_ARDUINO_TB.vhd. El informe de resultados es el archivo DISCRETE_COMM_ARDUINO_RESULTS.txt.

MAIN_PROCESSOR

En el bloque MAIN_PROCESSOR se prueban las siguientes funcionalidades:

- Para estas pruebas se implementa un conversor de binario de 8 bits a código ASCII para poder añadir en el archivo de informe de resultados la información enviada por el bloque MAIN_PROCESSOR en formato ASCII, tanto para el canal de comunicaciones UART_BLUETOOTH, como para los canales de comunicaciones UART_SHIELD y UART_ARDUINO.
- Se implementan relojes de transmisión para los puertos de comunicaciones de UART_BLUETOOTH, UART_SHIELD y UART_ARDUINO. De esta manera, en las formas de onda se puede ver cuándo está transmitiendo cada uno de los canales de transmisión.
- Se implementan canales de transmisión para los puertos de comunicaciones de UART_BLUETOOTH, UART_SHIELD y UART_ARDUINO. Almacenando los datos recibidos en mensajes de tipo binario de 8 bits con el tamaño correspondiente para cada uno de ellos.
- Se prueba la recepción del mensaje UART_BT_START_RESPONSE activando la entrada UART_BT_IN_RECEIVED.
- Se prueba la recepción del mensaje SHIELD_STATUS_CHAR activando la entrada UART_SH_IN_RECEIVED.
- Se prueba la recepción del mensaje ARDUINO_STATUS_CHAR activando las entradas UART_AR_IN_RECEIVED(0), UART_AR_IN_RECEIVED(1), UART_AR_IN_RECEIVED(2) y UART_AR_IN_RECEIVED(3).
- Se prueban las entradas discretas (DISC_IN_SH_EDGE(0), DISC_IN_SH_EDGE(1) y DISC_IN_SH_EDGE(2)) generando un pulso a nivel activo durante un ciclo de reloj y se espera a que se ponga a '1' el puerto de salida correspondiente (DISC_OUT_SH_LEVEL(0), ..., DISC_OUT_SH_LEVEL(8)).
- Se prueban las entradas discretas (DISC_IN_SH_EDGE(0), DISC_IN_SH_EDGE(1) y DISC_IN_SH_EDGE(2)) generando un pulso a nivel activo durante un ciclo de reloj y se espera a que se ponga a '0' el puerto de salida correspondiente (DISC_OUT_SH_LEVEL(0), ..., DISC_OUT_SH_LEVEL(8)).
- Se fija un pulso igual a "10101010" para los pulsos de entrada (PWM_AR_PULSE_HI(0), ..., PWM_AR_PULSE_HI(8), PWM_AR_PULSE_LO(0), ...,

PWM_AR_PULSE_LO(8)) y se comprueba el valor recibido en los puertos de comunicaciones de UART_BLUETOOTH cuando se envían datos a los puertos de comunicaciones UART_ARDUINO.

- Se prueban las entradas discretas (DISC_IN_AR_EDGE(0), ..., DISC_IN_AR_EDGE(11)) generando un pulso a nivel activo durante un ciclo de reloj y se espera a que se ponga a '1' el puerto de salida correspondiente (DISC_OUT_AR_LEVEL(0), ..., DISC_OUT_AR_LEVEL(11)).

Hay que indicar que se usa un MAIN_PROCESSOR_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama MAIN_PROCESSOR_TB.vhd. El informe de resultados es el archivo MAIN_PROCESSOR_RESULTS.txt.

FPGA_MAIN_MODULE

En el bloque MAIN_PROCESSOR se prueban las siguientes funcionalidades:

- Para estas pruebas se implementa un conversor de binario de 8 bits a código ASCII para poder añadir en el archivo de informe de resultados la información enviada por el bloque MAIN_PROCESSOR en formato ASCII, tanto para el canal de comunicaciones UART_BLUETOOTH, como para los canales de comunicaciones UART_SHIELD y UART_ARDUINO.
- Se implementan relojes de transmisión para los puertos de comunicaciones de UART_BLUETOOTH, UART_SHIELD y UART_ARDUINO. De esta manera, en las formas de onda se puede ver cuándo está transmitiendo cada uno de los canales de transmisión.
- Se implementan canales de transmisión para los puertos de comunicaciones de UART_BLUETOOTH, UART_SHIELD y UART_ARDUINO, almacenando los datos recibidos en mensajes de tipo binario de 8 bits con el tamaño correspondiente para cada uno de ellos.
- Se prueba la recepción del mensaje UART_BT_START_RESPONSE enviando la trama "10101010" a través de la entrada UART_BT_RX.
- Se prueba la recepción del mensaje SHIELD_UPDATE enviando la trama "10101010" a través de la entrada UART_SH_RX.
- Se prueban las entradas discretas (DISC_IN_1_SH_D2, DISC_IN_2_SH_D4 y DISC_IN_3_SH_D7) poniendo a nivel activo y se comprueba que el puerto de salida correspondiente esté a nivel activo (PWM_1_SH_D3, PWM_2_SH_D5, PWM_3_SH_D6, DISC_OUT_1_SH_D8, PWM_4_SH_D9, PWM_5_SH_D10, PWM_6_SH_D11, DISC_OUT_2_SH_D12 y DISC_OUT_3_SH_D13).
- Se prueban las entradas discretas (DISC_IN_1_SH_D2, DISC_IN_2_SH_D4 y DISC_IN_3_SH_D7) poniendo a nivel inactivo y se comprueba que el puerto de salida correspondiente esté a nivel inactivo (PWM_1_SH_D3, PWM_2_SH_D5,

PWM_3_SH_D6, DISC_OUT_1_SH_D8, PWM_4_SH_D9, PWM_5_SH_D10, PWM_6_SH_D11, DISC_OUT_2_SH_D12 y DISC_OUT_3_SH_D13).

- Se prueba la recepción del mensaje ARDUINO_UPDATE enviando la trama "10101010" a través de las entradas UART_AR1_RX, UART_AR2_RX, UART_AR3_RX y UART_AR4_RX.
- Se genera un pulso de 5 ms a nivel alto y un pulso de 10 ms a nivel bajo en los puertos de entrada PWM_1_AR1_D3, PWM_2_AR1_D5, PWM_3_AR1_D6, PWM_4_AR1_D9, PWM_5_AR1_D10 y PWM_6_AR1_D11. Se comprueba el valor recibido en el puerto de comunicaciones de UART_BLUETOOTH cuando se envían datos al puerto de comunicaciones UART_ARDUINO 1.
- Se genera un pulso de 6 ms a nivel alto y un pulso de 11 ms a nivel bajo en los puertos de entrada PWM_1_AR2_D3, PWM_2_AR2_D5, PWM_3_AR2_D6, PWM_4_AR2_D9, PWM_5_AR2_D10 y PWM_6_AR2_D11. Se comprueba el valor recibido en el puerto de comunicaciones de UART_BLUETOOTH cuando se envían datos al puerto de comunicaciones UART_ARDUINO 2.
- Se genera un pulso de 7 ms a nivel alto y un pulso de 12 ms a nivel bajo en los puertos de entrada PWM_1_AR3_D3, PWM_2_AR3_D5, PWM_3_AR3_D6, PWM_4_AR3_D9, PWM_5_AR3_D10 y PWM_6_AR3_D11. Se comprueba el valor recibido en el puerto de comunicaciones de UART_BLUETOOTH cuando se envían datos al puerto de comunicaciones UART_ARDUINO 3.
- Se genera un pulso de 8 ms a nivel alto y un pulso de 13 ms a nivel bajo en los puertos de entrada PWM_1_AR4_D3, PWM_2_AR4_D5, PWM_3_AR4_D6, PWM_4_AR4_D9, PWM_5_AR4_D10 y PWM_6_AR4_D11. Se comprueba el valor recibido en el puerto de comunicaciones de UART_BLUETOOTH cuando se envían datos al puerto de comunicaciones UART_ARDUINO 4.
- Se prueban las entradas discretas (DISC_IN_1_AR4_D2, DISC_IN_2_AR4_D4 y DISC_IN_3_AR4_D7) poniendo a nivel activo y se comprueba que el puerto de salida correspondiente esté a nivel activo (DISC_OUT_1_AR4_D8, DISC_OUT_2_AR4_D12 y DISC_OUT_3_AR4_D13).
- Se prueban las entradas discretas (DISC_IN_1_AR4_D2, DISC_IN_2_AR4_D4 y DISC_IN_3_AR4_D7) poniendo a nivel inactivo y se comprueba que el puerto de salida correspondiente esté a nivel inactivo (DISC_OUT_1_AR4_D8, DISC_OUT_2_AR4_D12 y DISC_OUT_3_AR4_D13).

Se usa un FPGA_MAIN_MODULE_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama FPGA_MAIN_MODULE_TB.vhd. El informe de resultados es el archivo FPGA_MAIN_MODULE_RESULTS.txt.

7.1.5. PASO 5: INTEGRACIÓN EN FPGA

Una vez completada la verificación, tanto de los bloques, como del módulo completo, se avanza al siguiente paso, la integración del firmware en la FPGA. Este paso se realiza mediante el uso de la herramienta Quartus II.

Síntesis y análisis

Se ha creado un proyecto en Quartus II. Se han añadido todos los archivos generados en el apartado 6.1.3 y se ha realizado el análisis y la síntesis del firmware.

- Version Quartus II 64-Bit: 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
- Nombre de la entidad de alto nivel: FPGA_MAIN_MODULE
- Familia: Cyclone II
- Elementos lógicos totales: 3.035
- Funciones combinatoriales totales: 2.801
- Registros dedicados a lógica: 2.026
- Registros totales: 2.026
- Pines totales: 86

La Tabla 98 muestra el resumen de uso de recursos de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 98. Resumen de uso de recursos de Análisis y Síntesis.

Resumen de uso de recursos de Análisis y Síntesis	
Recurso	Uso
Registros lógicos totales estimados	3,035
Funciones combinatoriales totales	2.801
Uso de elementos lógicos por número de entradas de LUT	
-- Funciones de 4 entradas	1.199
-- Funciones de 3 entradas	225
-- Funciones de menos de 2 entradas	1.377
Elementos lógicos por modo	
-- Modo normal	2.466
-- Modo aritmético	335
Registros totales	2.026
-- Registros lógicos dedicados	2.026
-- Registros de Entrada/Salida	0
Pines de Entrada/Salida	86
Nodo de máximo fan-out	CLK_50MHZ
Fan-out máximo	2.026
Fan-out total	13.828
Fan-out promedio	2,81

La Tabla 99 muestra el resumen de uso de recursos por entidad de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 99. Resumen de uso de recursos por entidad de Análisis y Síntesis.

Resumen de uso de recursos por entidad de Análisis y Síntesis		
Entidad jerárquica	Celdas Lógicas	Registros
FPGA_MAIN_MODULE	2801 (64)	2026 (29)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_1	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_2	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_3	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_4	9 (9)	9 (9)
DISCRETE_COMM_SHIELD:INST_DISCRETE_COMM_SHIELD	9 (9)	15 (15)
MAIN_PROCESSOR:INST_MAIN_PROCESSOR	1416 (1416)	971 (971)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_1	178 (178)	184 (184)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_2	156 (156)	168 (168)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_3	156 (156)	168 (168)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_4	156 (156)	168 (168)
UART_ARDUINO:INST_UART_ARDUINO_1	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_2	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_3	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_4	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_BLUETOOTH:INST_UART_BLUETOOTH	90 (0)	42 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	38 (38)	18 (18)
UART_TX_MACHINE:INST_UART_TX_MACHINE	52 (52)	24 (24)
UART_SHIELD:INST_UART_SHIELD	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)

Especificación de pines de la FPGA

Una vez obtenidos los resultados de síntesis se realiza la asignación de pines de la FPGA a los puertos de entrada y salida del firmware. Se han creado grupos de pines para tener más claridad a la hora de gestionar y mantener la asignación de pines.

La Tabla 100 muestra la relación entre los puertos de entrada y salida del firmware (FW) y el pin de la FPGA que se le ha asignado para el grupo hardware Externo (HWE):

Tabla 100. Relación entre los puertos E/S del FW y pines de la FPGA para el grupo de HWE.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo de hardware Externo					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
nRESET	Entrada	PIN_144	2	B2_N1	3.3-V LVTTTL (defecto)
CLK_50MHZ	Entrada	PIN_17	1	B1_N0	3.3-V LVTTTL (defecto)
LED_2	Salida	PIN_3	1	B1_N0	3.3-V LVTTTL (defecto)
LED_4	Salida	PIN_7	1	B1_N0	3.3-V LVTTTL (defecto)
LED_5	Salida	PIN_9	1	B1_N0	3.3-V LVTTTL (defecto)
UART_BT_RX	Entrada	PIN_104	3	B3_N0	3.3-V LVTTTL (defecto)
UART_BT_TX	Salida	PIN_103	3	B3_N0	3.3-V LVTTTL (defecto)

La Tabla 101 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo Module Shield:

Tabla 101. Relación entre los puertos E/S del FW y pines FPGA para el Shield.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Module Shield					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
UART_SH_RX	Entrada	PIN_64	4	B4_N0	3.3-V LVTTTL (defecto)
UART_SH_TX	Salida	PIN_63	4	B4_N0	3.3-V LVTTTL (defecto)
DISC_IN_1_SH_D2	Entrada	PIN_65	4	B4_N0	3.3-V LVTTTL (defecto)
PWM_1_SH_D3	Salida	PIN_67	4	B4_N0	3.3-V LVTTTL (defecto)
DISC_IN_2_SH_D4	Entrada	PIN_69	4	B4_N0	3.3-V LVTTTL (defecto)
PWM_3_SH_D6	Salida	PIN_71	4	B4_N0	3.3-V LVTTTL (defecto)
PWM_2_SH_D5	Salida	PIN_70	4	B4_N0	3.3-V LVTTTL (defecto)
DISC_IN_3_SH_D7	Entrada	PIN_72	4	B4_N0	3.3-V LVTTTL (defecto)
DISC_OUT_1_SH_D8	Salida	PIN_96	3	B3_N0	3.3-V LVTTTL (defecto)

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Module Shield					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
PWM_4_SH_D9	Salida	PIN_97	3	B3_N0	3.3-V LVTTTL (defecto)
PWM_5_SH_D10	Salida	PIN_99	3	B3_N0	3.3-V LVTTTL (defecto)
PWM_6_SH_D11	Salida	PIN_100	3	B3_N0	3.3-V LVTTTL (defecto)
DISC_OUT_2_SH_D12	Salida	PIN_101	3	B3_N0	3.3-V LVTTTL (defecto)
DISC_OUT_3_SH_D13	Salida	PIN_136	2	B2_N1	3.3-V LVTTTL (defecto)

La Tabla 102 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo Arduino 1:

Tabla 102. Relación entre los puertos E/S del FW y pines FPGA para Arduino 1.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 1					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
UART_AR1_RX	Entrada	PIN_42	4	B4_N1	3.3-V LVTTTL (defecto)
UART_AR1_TX	Salida	PIN_41	4	B4_N1	3.3-V LVTTTL (defecto)
DISC_IN_1_AR1_D2	Entrada	PIN_43	4	B4_N1	3.3-V LVTTTL (defecto)
DISC_IN_2_AR1_D4	Entrada	PIN_45	4	B4_N1	3.3-V LVTTTL (defecto)
DISC_IN_3_AR1_D7	Entrada	PIN_51	4	B4_N1	3.3-V LVTTTL (defecto)
DISC_OUT_1_AR1_D8	Salida	PIN_52	4	B4_N1	3.3-V LVTTTL (defecto)
DISC_OUT_2_AR1_D12	Salida	PIN_58	4	B4_N1	3.3-V LVTTTL (defecto)
DISC_OUT_3_AR1_D13	Salida	PIN_59	4	B4_N0	3.3-V LVTTTL (defecto)
PWM_1_AR1_D3	Entrada	PIN_44	4	B4_N1	3.3-V LVTTTL (defecto)
PWM_2_AR1_D5	Entrada	PIN_47	4	B4_N1	3.3-V LVTTTL (defecto)
PWM_3_AR1_D6	Entrada	PIN_48	4	B4_N1	3.3-V LVTTTL (defecto)
PWM_4_AR1_D9	Entrada	PIN_53	4	B4_N1	3.3-V LVTTTL (defecto)

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 1					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
PWM_5_AR1_D10	Entrada	PIN_55	4	B4_N1	3.3-V LVTTTL (defecto)
PWM_6_AR1_D11	Entrada	PIN_57	4	B4_N1	3.3-V LVTTTL (defecto)
nRESET_AR1	Salida	PIN_60	2	B2_N1	3.3-V LVTTTL (defecto)

La Tabla 103 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo Arduino 2:

Tabla 103. Relación entre los puertos E/S del FW y pines FPGA para Arduino 2.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 2					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
UART_AR2_RX	Entrada	PIN_75	3	B3_N1	3.3-V LVTTTL (defecto)
UART_AR2_TX	Salida	PIN_74	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_IN_1_AR2_D2	Entrada	PIN_76 (PIN_40) *	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_1_AR2_D3	Entrada	PIN_79	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_IN_2_AR2_D4	Entrada	PIN_80 **	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_2_AR2_D5	Entrada	PIN_81 ***	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_3_AR2_D6	Entrada	PIN_86	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_IN_3_AR2_D7	Entrada	PIN_87	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_OUT_1_AR2_D8	Salida	PIN_88 (PIN_73) ****	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_4_AR2_D9	Entrada	PIN_89	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_5_AR2_D10	Entrada	PIN_90	3	B3_N0	3.3-V LVTTTL (defecto)
PWM_6_AR2_D11	Entrada	PIN_91	3	B3_N0	3.3-V LVTTTL (defecto)
DISC_OUT_2_AR2_D12	Salida	PIN_92	3	B3_N0	3.3-V LVTTTL (defecto)

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 2					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
DISC_OUT_3_AR2_D13	Salida	PIN_93	3	B3_N0	3.3-V LVTTTL (defecto)
nRE-SET_AR2	Salida	PIN_94	2	B2_N1	3.3-V LVTTTL (defecto)

* El PIN_76 no se puede usar por restricciones de la FPGA. Para esta versión se ha usado el PIN_40. Para la siguiente versión del módulo principal es necesario sustituir el PIN_76 por el PIN_40. En la siguiente versión, el PIN_76 deberá ser clasificado como No Usado.

** El PIN_80 está conectado a GND en la tarjeta, con lo cual siempre estará a un valor de cero lógico. Para la siguiente versión del módulo principal es necesario sustituir el PIN_80 por el PIN_88. En la siguiente versión, el PIN_80 deberá ser clasificado como No Usado.

*** El PIN_81 está conectado a VCC de 1,2 V en la tarjeta, con lo cual siempre estará a un valor de cero lógico. Para la siguiente versión del módulo principal es necesario sustituir el PIN_80 por el PIN_112.

**** El PIN_88 es un pin dedicado a reloj de la FPGA, con lo cual sólo puede ser usado en modo entrada. Para esta versión se ha usado el PIN_73. Para la siguiente versión del módulo principal es necesario sustituir el PIN_88 por el PIN_73.

La Tabla 104 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo Arduino 3:

Tabla 104. Relación entre los puertos E/S del FW y pines FPGA para Arduino 3.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 3					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
UART_AR3_RX	Entrada	PIN_114	2	B2_N0	3.3-V LVTTTL (defecto)
UART_AR3_TX	Salida	PIN_113	2	B2_N0	3.3-V LVTTTL (defecto)
DISC_IN_1_AR3_D2	Entrada	PIN_115	2	B2_N0	3.3-V LVTTTL (defecto)
PWM_1_AR3_D3	Entrada	PIN_118	2	B2_N0	3.3-V LVTTTL (defecto)
DISC_IN_2_AR3_D4	Entrada	PIN_119	2	B2_N0	3.3-V LVTTTL (defecto)

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 3					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
PWM_2_AR3_D5	Entrada	PIN_120	2	B2_NO	3.3-V LVTTTL (defecto)
PWM_3_AR3_D6	Entrada	PIN_121	2	B2_NO	3.3-V LVTTTL (defecto)
DISC_IN_3_AR3_D7	Entrada	PIN_122	2	B2_NO	3.3-V LVTTTL (defecto)
DISC_OUT_1_AR3_D8	Salida	PIN_125	2	B2_NO	3.3-V LVTTTL (defecto)
PWM_4_AR3_D9	Entrada	PIN_126	2	B2_NO	3.3-V LVTTTL (defecto)
PWM_5_AR3_D10	Entrada	PIN_129	2	B2_N1	3.3-V LVTTTL (defecto)
PWM_6_AR3_D11	Entrada	PIN_132	2	B2_N1	3.3-V LVTTTL (defecto)
DISC_OUT_2_AR3_D12	Salida	PIN_133	2	B2_N1	3.3-V LVTTTL (defecto)
DISC_OUT_3_AR3_D13	Salida	PIN_134	2	B2_N1	3.3-V LVTTTL (defecto)
nRESET_AR3	Salida	PIN_135	2	B2_N1	3.3-V LVTTTL (defecto)

La Tabla 105 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo Arduino 4:

Tabla 105. Relación entre los puertos E/S del FW y pines FPGA para Arduino 4.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 4					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
UART_AR4_RX	Entrada	PIN_143	2	B2_N1	3.3-V LVTTTL (defecto)
UART_AR4_TX	Salida	PIN_142	2	B2_N1	3.3-V LVTTTL (defecto)
DISC_IN_1_AR4_D2	Entrada	PIN_4	1	B1_NO	3.3-V LVTTTL (defecto)
PWM_1_AR4_D3	Entrada	PIN_8	1	B1_NO	3.3-V LVTTTL (defecto)
DISC_IN_2_AR4_D4	Entrada	PIN_18	1	B1_NO	3.3-V LVTTTL (defecto)
PWM_2_AR4_D5	Entrada	PIN_22	1	B1_N1	3.3-V LVTTTL (defecto)
PWM_3_AR4_D6	Entrada	PIN_21	1	B1_N1	3.3-V LVTTTL (defecto)

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 4					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
DISC_IN_3_AR4_D7	Entrada	PIN_25	1	B1_N1	3.3-V LVTTTL (defecto)
DISC_OUT_1_AR4_D8	Salida	PIN_24	1	B1_N1	3.3-V LVTTTL (defecto)
PWM_4_AR4_D9	Entrada	PIN_27	1	B1_N1	3.3-V LVTTTL (defecto)
PWM_5_AR4_D10	Entrada	PIN_26	1	B1_N1	3.3-V LVTTTL (defecto)
PWM_6_AR4_D11	Entrada	PIN_30	1	B1_N1	3.3-V LVTTTL (defecto)
DISC_OUT_2_AR4_D12	Salida	PIN_28	1	B1_N1	3.3-V LVTTTL (defecto)
DISC_OUT_3_AR4_D13	Salida	PIN_32	1	B1_N1	3.3-V LVTTTL (defecto)
nRESET_AR4	Salida	PIN_31	2	B2_N1	3.3-V LVTTTL (defecto)

La Tabla 106 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo No Usado:

Tabla 106. Relación entre los puertos E/S del FW y pines FPGA para No Usado.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo No Usado					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
USER_IO_1	Entrada	PIN_1	1	B1_N0	3.3-V LVTTTL (defecto)
USER_IO_2	Entrada	PIN_2	1	B1_N0	3.3-V LVTTTL (defecto)
USER_IO_31	Entrada	PIN_31	1	B1_N1	3.3-V LVTTTL (defecto)
USER_IO_40	Entrada	PIN_40 *	4	B4_N1	3.3-V LVTTTL (defecto)
USER_IO_88	Entrada	PIN_88 **	3	B3_N1	3.3-V LVTTTL (defecto)
USER_IO_94	Entrada	PIN_94	3	B3_N0	3.3-V LVTTTL (defecto)
USER_IO_112	Entrada	PIN_135	2	B2_N1	3.3-V LVTTTL (defecto)
USER_IO_135	Entrada	PIN_135	2	B2_N1	3.3-V LVTTTL (defecto)
USER_IO_137	Entrada	PIN_137	2	B2_N1	3.3-V LVTTTL (defecto)

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo No Usado					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
USER_IO_139	Entrada	PIN_139	2	B2_N1	3.3-V LVTTTL (defecto)
USER_IO_141	Entrada	PIN_141	2	B2_N1	3.3-V LVTTTL (defecto)

* El PIN_40 se usará en la siguiente versión del módulo principal. Por lo tanto, desaparecerá de este grupo.

** El PIN_88 se usará en la siguiente versión del módulo principal. Por lo tanto, el PIN_80 pasará a ser un pin del grupo No Usado.

Resultados de la integración en FPGA

Una vez finalizada la implementación con la herramienta Quartus II se generan los informes de uso de recursos. La Tabla 107 muestra el resumen de uso de recursos de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 107. Resumen de uso de recursos de Análisis y Síntesis, versión 1 de firmware.

Resumen de uso de recursos de Análisis y Síntesis	
Recurso	Uso
Registros lógicos totales estimados	3040 / 4608 (66 %)
Combinacional sin registrar	1014
Sólo registros	239
Combinacional registrada	1787
Uso de elementos lógicos por número de entradas de LUT	
-- Funciones de 4 entradas	1199
-- Funciones de 3 entradas	225
-- Funciones de menos de 2 entradas	1377
-- Sólo registrados	239
Elementos lógicos por modo	
-- Modo normal	2466
-- Modo aritmético	335
Registros totales	2026 / 4851 (42 %)
-- Registros lógicos dedicados	2026 / 4608 (44 %)
-- Registros de Entrada/Salida	0 / 243 (0 %)
Pines de Entrada/Salida	86 / 89 (97 %)
Nodo de máximo fan-out	CLK_50MHZ
Fan-out máximo	2026
Fan-out total	14,009
Fan-out promedio	2,75

La Tabla 108 muestra el resumen de uso de recursos por entidad de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 108. Uso de recursos por entidad de Análisis y Síntesis, versión 1 de firmware.

Resumen de uso de recursos por entidad de Análisis y Síntesis		
Entidad jerárquica	Celdas Lógicas	Registros
FPGA_MAIN_MODULE	3040 (64)	2026 (29)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_1	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_2	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_3	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_4	9 (9)	9 (9)
DISCRETE_COMM_SHIELD:INST_DISCRETE_COMM_SHIELD	15 (15)	15 (15)
MAIN_PROCESSOR:INST_MAIN_PROCESSOR	1451 (1451)	971 (971)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_1	226 (226)	184 (184)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_2	204 (204)	168 (168)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_3	204 (204)	168 (168)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_4	204 (204)	168 (168)
UART_ARDUINO:INST_UART_ARDUINO_1	109 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	52 (52)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_2	109 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	52 (52)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_3	109 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	52 (52)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_4	109 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	52 (52)	23 (23)
UART_BLUETOOTH:INST_UART_BLUETOOTH	91 (0)	42 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	38 (38)	18 (18)
UART_TX_MACHINE:INST_UART_TX_MACHINE	53 (53)	24 (24)
UART_SHIELD:INST_UART_SHIELD	109 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	52 (52)	23 (23)

7.1.6. PASO 6: PRUEBAS DE INTEGRACIÓN HARDWARE-FIRMWARE

Las pruebas de integración hardware-firmware son pruebas que tratan de detectar la ausencia de errores de diseño firmware y en el comportamiento de la FPGA integrada en el hardware del módulo principal.

Las pruebas de integración hardware-firmware que se han de pasar son las siguientes:

1. Comportamiento de los LED de estado: se comprueba que los LED de estado de la FPGA se comportan correctamente.
2. Prueba de comunicación interna: se comprueba que la FPGA y el módulo bluetooth se comunican correctamente.
3. Prueba de comunicación externa: se comprueba que el módulo principal es capaz de comunicarse con un PC conectado por bluetooth.

7.2. Primera versión de firmware, versión 1

Se trata de la primera versión de firmware. Esta versión ya permite el uso de todas las funcionalidades:

- Comunicaciones serie con módulo Bluetooth.
- Comunicaciones serie con Shield del módulo principal.
- Comunicaciones serie con las Arduino Nano de los módulos secundarios.
- Detección de entradas discretas del Shield del módulo principal.
- Gestión de salidas discretas del Shield del módulo principal.
- Monitorización de los puertos PWM de las Arduino Nano de los módulos secundarios.
- Detección de entradas discretas de las Arduino Nano de los módulos secundarios.
- Gestión de salidas discretas de las Arduino Nano de los módulos secundarios.

Pruebas de integración hardware-firmware

En estas pruebas de integración hardware-firmware de la versión 1.1 del módulo principal, la versión 1.1 del módulo secundario (ver sección 6.3) y la versión 1 de firmware (descrita en esta sección).

En primer lugar, se carga el firmware en la FPGA. Para cargar el firmware en la memoria es necesario el programador USB Blaster II. La Figura 103 muestra un montaje con el programador USB Blaster II, el módulo principal y el sistema de alimentación.



Figura 103. Carga del firmware en la FPGA.

Los LED L2, L4 y L5 parpadean de acuerdo con el diseño. Al ensamblar la placa de la FPGA en el módulo principal, los LED L2, L4 y L5 lucen en aspecto fijo, lo cual da a entender que está activo el reset de la FPGA.

Tras una inspección del hardware, se detecta que el reset de la FPGA se ha fijado a GND en el módulo principal. Este error de hardware debe ser arreglado para la primera línea de base. Se aprovechará para aplicar las mejoras propuestas de la versión 1.1 de hardware, además de los errores detectados durante la fase de diseño firmware de la versión 1.

Para poder seguir avanzando en la integración se elimina el pin 25 del conector P3.

Después de aplicar el cambio se prueba la FPGA con el módulo principal, los LED L2, L4 y L5 parpadean de acuerdo con el diseño. La Figura 103 muestra como se ha cargado el firmware correctamente y los LED de estado lucen de acuerdo con el diseño implementado.

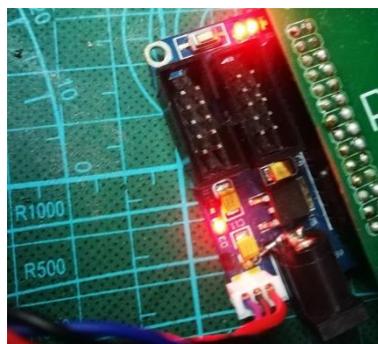


Figura 104. LED de estado de la FPGA.

Posteriormente se conecta el módulo Bluetooth. La Figura 104 muestra un montaje con la alimentación, el módulo principal y el módulo Bluetooth.

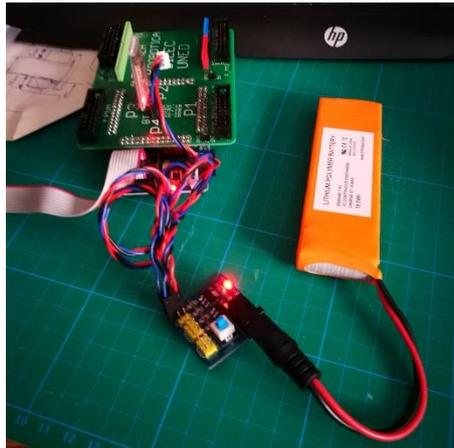


Figura 105. Integración del módulo Bluetooth.

La siguiente prueba de integración hardware-firmware consiste en identificar si los puertos serie Bluetooth y Shield funcionan correctamente. Para ello se va a usar un módulo Bluetooth HC-06. Este módulo debe ser configurado de acuerdo con el diseño firmware, es decir, con una velocidad de transmisión de 115200 baudios. Esto se realiza mediante los comandos AT. La velocidad por defecto es de 9600 baudios, para cambiarlo se hace uso del siguiente comando AT:

- Enviar: AT+BAUD<Numero>
- Respuesta: OK<baudrate>

Donde <Numero> equivale a una velocidad de <baudrate>, los valores pueden ser los que la Tabla 109 contiene:

Tabla 109. Valores de configuración de velocidad de transmisión del módulo HC-06.

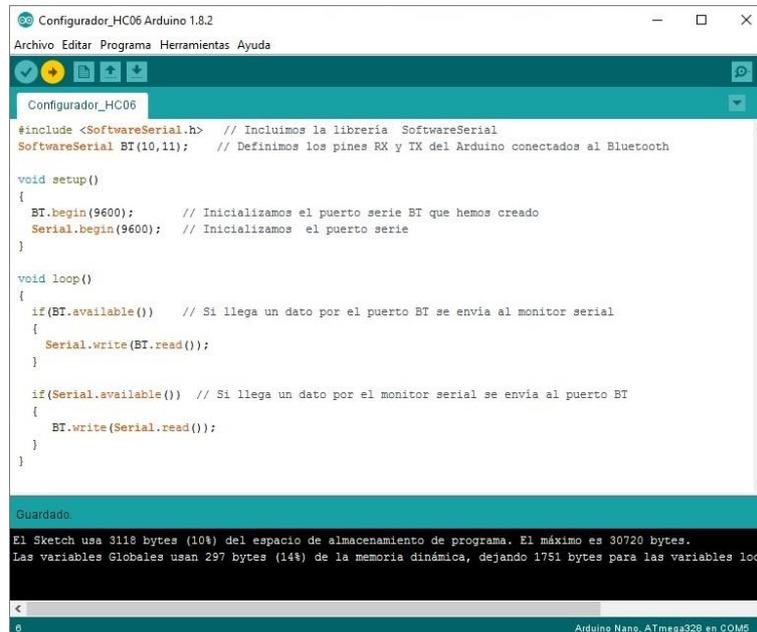
Numero	Baudrate
1	1200
2	2400
3	4800
4	9600
5	19200
6	38400
7	57600
8	115200

Adicionalmente se le modifica el nombre con el uso del siguiente comando:

- Enviar: AT+NAME<Nombre> Ejemplo: AT+NAMERobot.
- Respuesta: OKsetname.

El nombre puede ser de hasta 20 caracteres como máximo

Se usa una Arduino Nano para configurar el módulo Bluetooth. La Figura 106 muestra el código Arduino para la configuración del módulo Bluetooth.



```

Configurador_HC06 Arduino 1.8.2
Archivo Editar Programa Herramientas Ayuda

Configurador_HC06

#include <SoftwareSerial.h> // Incluimos la libreria SoftwareSerial
SoftwareSerial BT(10,11); // Definimos los pines RX y TX del Arduino conectados al Bluetooth

void setup()
{
  BT.begin(9600); // Inicializamos el puerto serie BT que hemos creado
  Serial.begin(9600); // Inicializamos el puerto serie
}

void loop()
{
  if(BT.available()) // Si llega un dato por el puerto BT se envía al monitor serial
  {
    Serial.write(BT.read());
  }

  if(Serial.available()) // Si llega un dato por el monitor serial se envía al puerto BT
  {
    BT.write(Serial.read());
  }
}

Guardado:
El Sketch usa 3118 bytes (10%) del espacio de almacenamiento de programa. El máximo es 30720 bytes.
Las variables Globales usan 297 bytes (14%) de la memoria dinámica, dejando 1751 bytes para las variables loca
Arduino Nano, ATmega328 en COM5
  
```

Figura 106. Código Arduino para la configuración del módulo Bluetooth.

La Figura 107 muestra captura de Windows de detección del módulo Bluetooth. En esta captura se puede ver el nombre del dispositivo y el estado de la conexión Bluetooth.

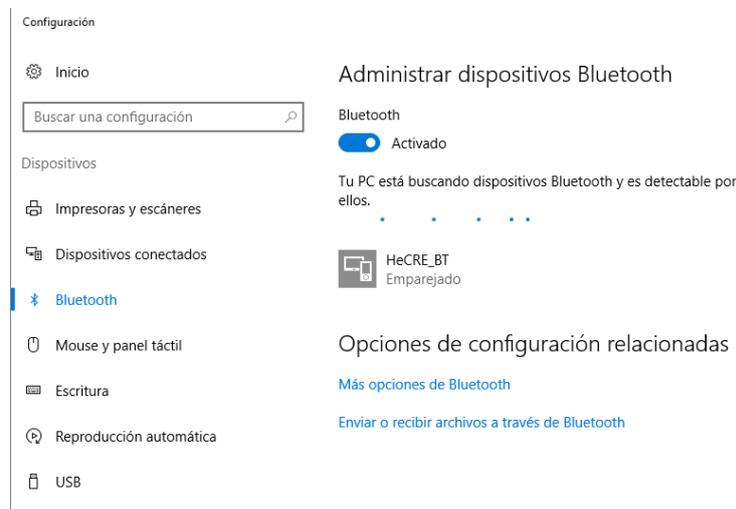


Figura 107. Captura de Windows de detección del módulo Bluetooth.

La Figura 108 muestra la respuesta del módulo Bluetooth cuando se le envía un carácter. Los caracteres mostrados son los que envía el HeCRE a través de Bluetooth al PC.

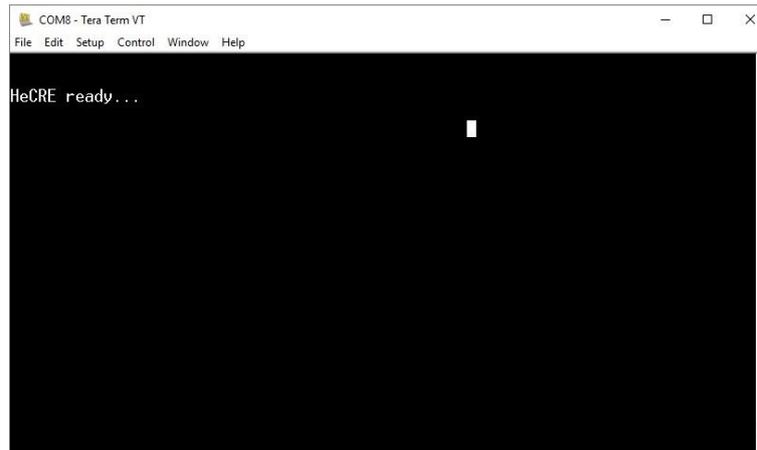


Figura 108. Respuesta del módulo Bluetooth cuando se le envía un caracter.

Debido a los errores detectados durante esta integración hardware-firmware de la versión 1.1 del módulo principal, la versión 1.1 del módulo secundario (ver sección 5.3) y la versión 1 de firmware (descrita en esta sección) es necesario aplicar cambios en el hardware, ver sección 5.4.

7.3. Segunda versión de firmware, versión 2

Esta versión contiene los cambios detectados como resultado de la integración hardware-firmware de la versión 1.1 del módulo principal, la versión 1.1 del módulo secundario con la versión 1 de firmware (ver sección 7.2). De acuerdo con la metodología descrita en el capítulo 5, los cambios aplicados para la segunda versión de firmware, versión 2, son motivados por las pruebas de integración de hardware-firmware (ver sección 7.2) y las modificaciones realizadas al hardware (sección 6.4). La Figura 109 muestra la transición que proviene del desarrollo de firmware y afecta al desarrollo de firmware.

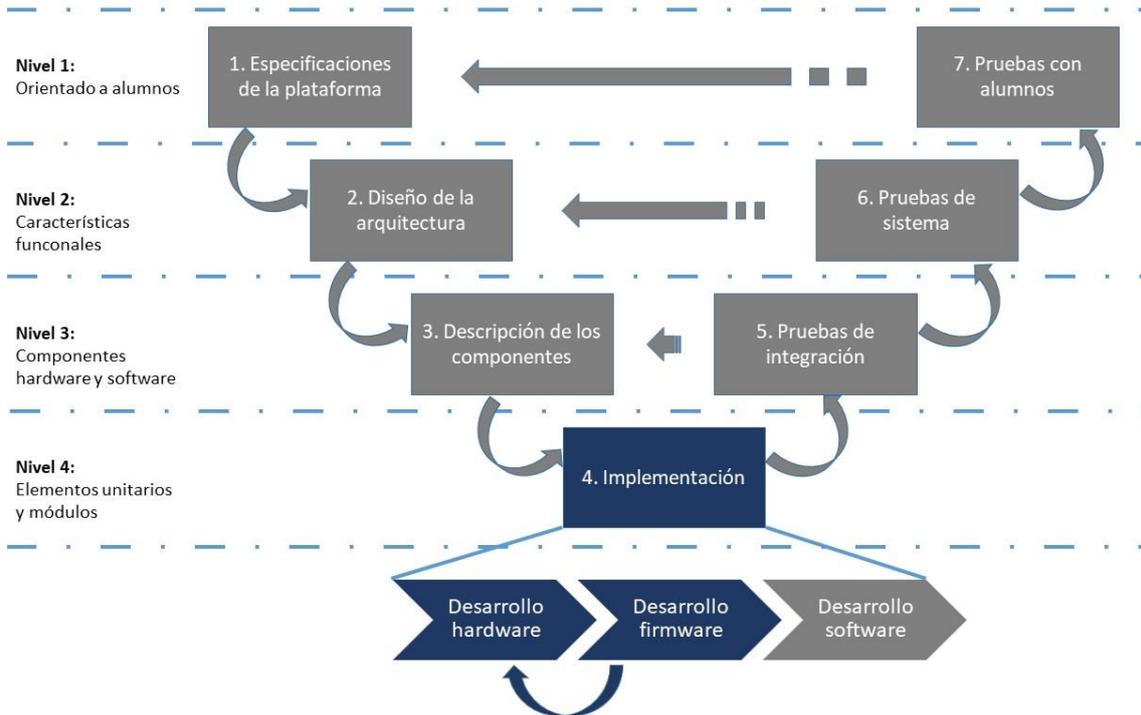


Figura 109. Diagrama de bloques del primer cambio firmware.

La Tabla 110 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo Arduino 2:

Tabla 110. Relación entre los puertos E/S del FW y pines FPGA para Arduino 2.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo Arduino 2					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
UART_AR2_RX	Entrada	PIN_75	3	B3_N1	3.3-V LVTTTL (defecto)
UART_AR2_TX	Salida	PIN_74	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_IN_1_AR2_D2	Entrada	PIN_40	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_1_AR2_D3	Entrada	PIN_79	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_IN_2_AR2_D4	Entrada	PIN_88	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_2_AR2_D5	Entrada	PIN_112	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_3_AR2_D6	Entrada	PIN_86	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_IN_3_AR2_D7	Entrada	PIN_87	3	B3_N1	3.3-V LVTTTL (defecto)
DISC_OUT_1_AR2_D8	Salida	PIN_73	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_4_AR2_D9	Entrada	PIN_89	3	B3_N1	3.3-V LVTTTL (defecto)
PWM_5_AR2_D10	Entrada	PIN_90	3	B3_N0	3.3-V LVTTTL (defecto)
PWM_6_AR2_D11	Entrada	PIN_91	3	B3_N0	3.3-V LVTTTL (defecto)
DISC_OUT_2_AR2_D12	Salida	PIN_92	3	B3_N0	3.3-V LVTTTL (defecto)
DISC_OUT_3_AR2_D13	Salida	PIN_93	3	B3_N0	3.3-V LVTTTL (defecto)
nRESET_AR2	Salida	PIN_94	2	B2_N1	3.3-V LVTTTL (defecto)

La Tabla 111 muestra la relación entre los puertos de entrada y salida del firmware y el pin de la FPGA que se le ha asignado para el grupo No Usado:

Tabla 111. Relación entre los puertos E/S del FW y pines FPGA para No Usado.

Relación entre los puertos E/S del firmware y pines de la FPGA para el grupo No Usado					
Nombre del nodo	Sentido	Posición	Banco de E/S	Grupo VREF	Estándar E/S
USER_IO_80	Entrada	PIN_80	4	B4_N1	3.3-V LVTTTL (defecto)
USER_IO_81	Entrada	PIN_80	4	B4_N1	3.3-V LVTTTL (defecto)
USER_IO_137	Entrada	PIN_137	2	B2_N1	3.3-V LVTTTL (defecto)
USER_IO_139	Entrada	PIN_139	2	B2_N1	3.3-V LVTTTL (defecto)
USER_IO_141	Entrada	PIN_141	2	B2_N1	3.3-V LVTTTL (defecto)

Resultados de la integración en FPGA

Una vez finalizada la implementación con la herramienta Quartus II se generan los informes de uso de recursos. La Tabla 112 muestra el resumen de uso de recursos de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 112. Resumen de uso de recursos de Análisis y Síntesis, versión 2 de firmware.

Resumen de uso de recursos de Análisis y Síntesis	
Recurso	Uso
Registros lógicos totales estimados	3044 / 4608 (66 %)
Combinacional sin registrar	1018
Sólo registros	239
Combinacional registrada	1787
Uso de elementos lógicos por número de entradas de LUT	
-- Funciones de 4 entradas	1198
-- Funciones de 3 entradas	224
-- Funciones de menos de 2 entradas	1383
-- Sólo registrados	239
Elementos lógicos por modo	
-- Modo normal	2470
-- Modo aritmético	335
Registros totales	2026 / 4851 (42 %)
-- Registros lógicos dedicados	2026 / 4608 (44 %)
-- Registros de Entrada/Salida	0 / 243 (0 %)
Pines de Entrada/Salida	86 / 89 (97 %)
Nodo de máximo fan-out	CLK_50MHZ
Fan-out máximo	2026
Fan-out total	14,001
Fan-out promedio	2,75

La Tabla 113 muestra el resumen de uso de recursos por entidad de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 113. Uso de recursos por entidad de Análisis y Síntesis, versión 2 de firmware.

Resumen de uso de recursos por entidad de Análisis y Síntesis		
Entidad jerárquica	Celdas Lógicas	Registros
FPGA_MAIN_MODULE	2801 (64)	2026 (29)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_1	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_2	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_3	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_4	9 (9)	9 (9)
DISCRETE_COMM_SHIELD:INST_DISCRETE_COMM_SHIELD	9 (9)	15 (15)
MAIN_PROCESSOR:INST_MAIN_PROCESSOR	1416 (1416)	971 (971)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_1	178 (178)	184 (184)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_2	156 (156)	168 (168)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_3	156 (156)	168 (168)
PWM_MONITORING_ARDUINO:INST_PWM_MONITORING_ARDUINO_4	156 (156)	168 (168)
UART_ARDUINO:INST_UART_ARDUINO_1	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_2	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_3	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_4	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)
UART_BLUETOOTH:INST_UART_BLUETOOTH	90 (0)	42 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	38 (38)	18 (18)
UART_TX_MACHINE:INST_UART_TX_MACHINE	52 (52)	24 (24)
UART_SHIELD:INST_UART_SHIELD	108 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	51 (51)	23 (23)

Pruebas de integración hardware-firmware

Tras obtener la versión 2 del firmware se procede a realizar las pruebas de integración hardware-firmware descritas en la sección 6.1.6:

1. Se comprueba que los LED de estado de la FPGA se comportan correctamente.
2. Se comprueba que la FPGA y el módulo bluetooth se comunican correctamente.

3. Se comprueba que el módulo principal es capaz de comunicarse con un PC conectado por bluetooth.

Una vez concluidas las pruebas se comprueba que la versión 2 del firmware está lista para ser usada en las pruebas de integración hardware-firmware-software – ver sección 7.2.

7.4. Tercera versión de firmware, versión 3

Esta versión contiene los cambios detectados como resultado de la integración hardware-firmware-software de la versión 2 del firmware con la versión 1.2 del hardware (sección 7.2). De acuerdo con la metodología descrita en el capítulo 5, los cambios aplicados para la tercera versión de firmware, versión 3 son motivados por las pruebas de integración de hardware-firmware-software (sección 7.3) y las modificaciones realizadas al hardware (sección 6.5). La Figura 110 muestra la transición que proviene del desarrollo de firmware y afecta al desarrollo de firmware.

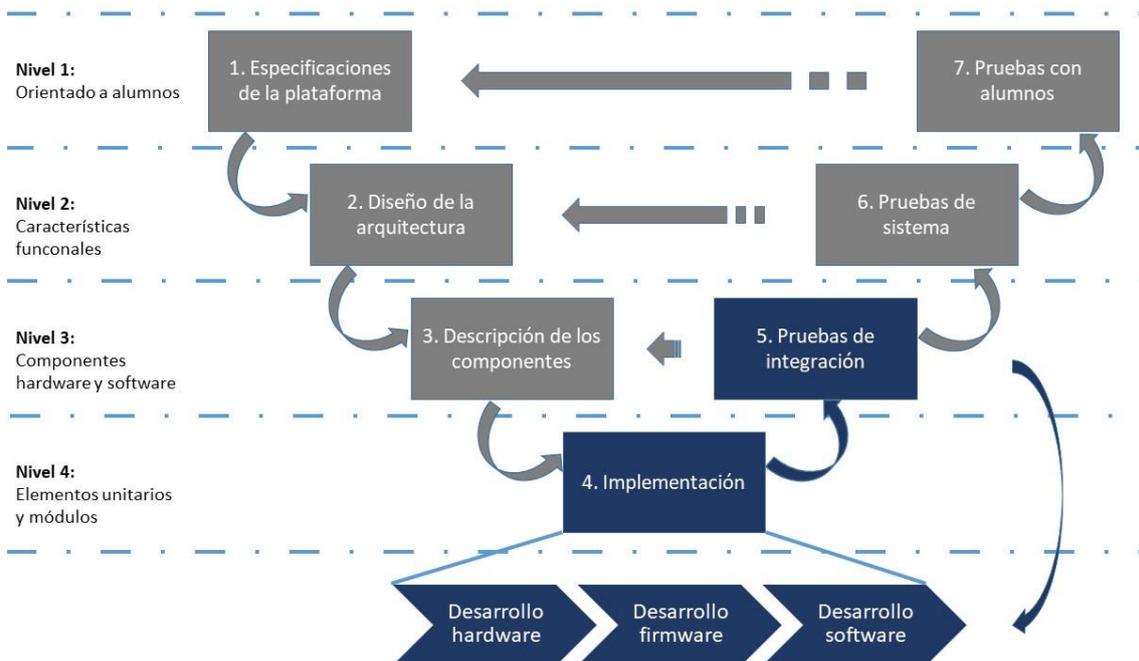


Figura 110. Diagrama de bloques del segundo cambio firmware.

Esta versión de firmware contiene los siguientes cambios que surgen de los cambios hardware aplicados:

- Se reduce el número de entradas/salidas entre el módulo principal y los módulos secundarios.
- El bloque UART_SHIELD se elimina.

- El bloque DISCRETE_COMM_SHIELD se elimina.
- El bloque PWM_MONITORING_ARDUINO se elimina.
- La funcionalidad de reset a las Arduino de los módulos secundarios se elimina.

Para la versión 3 de firmware ya no hay puertos conectados con el Shield Arduino. Las conexiones con los módulos secundarios se reducen. Los puertos de la FPGA están conectados con el módulo secundario 1 mediante el conector P_MM_1, con el módulo secundario 2 mediante el conector P_MM_2, con el módulo secundario 3 mediante el conector P_MM_3 y con el módulo secundario 4 mediante el conector P_MM_4. Las tablas Tabla 114, Tabla 115, Tabla 116 y Tabla 117 detallan la funcionalidad de cada uno de ellos.

Tabla 114. Puertos de la FPGA conectados al módulo secundario 1.

Conector P_MM_1	
Pin FPGA	Funcionalidad
41	Puerto de transmisión de datos de la FPGA a la Arduino Nano 1.
42	Puerto de recepción de datos de la FPGA de la Arduino Nano 1.
43	Puerto digital D2 de la Arduino Nano 1. Entrada discreta 1 de la Arduino Nano 1.
45	Puerto digital D4 de la Arduino Nano 1. Entrada discreta 2 de la Arduino Nano 1.
51	Puerto digital D7 de la Arduino Nano 1. Entrada discreta 3 de la Arduino Nano 1.
52	Puerto digital D8 de la Arduino Nano 1. Salida discreta 1 de la Arduino Nano 1.
58	Puerto digital D12 de la Arduino Nano 1. Salida discreta 2 de la Arduino Nano 1.
59	Puerto digital D13 de la Arduino Nano 1. Salida discreta 3 de la Arduino Nano 1.

Tabla 115. Puertos de la FPGA conectados al módulo secundario 2.

Conector P_MM_2	
Pin FPGA	Funcionalidad
74	Puerto de transmisión de datos de la FPGA a la Arduino Nano 2.
75	Puerto de recepción de datos de la FPGA de la Arduino Nano 2.
76	Puerto digital D2 de la Arduino Nano 2. Entrada discreta 1 de la Arduino Nano 2.
80	Puerto digital D4 de la Arduino Nano 2. Entrada discreta 2 de la Arduino Nano 2.
87	Puerto digital D7 de la Arduino Nano 2. Entrada discreta 3 de la Arduino Nano 2.
88	Puerto digital D8 de la Arduino Nano 2. Salida discreta 1 de la Arduino Nano 2.
92	Puerto digital D12 de la Arduino Nano 2. Salida discreta 2 de la Arduino Nano 2.
93	Puerto digital D13 de la Arduino Nano 2. Salida discreta 3 de la Arduino Nano 2.
94	Reset de la Arduino Nano 2.

Tabla 116. Puertos de la FPGA conectados al módulo secundario 3.

Conector P_MM_3	
Pin FPGA	Funcionalidad
113	Puerto de transmisión de datos de la FPGA a la Arduino Nano 3.

Conector P_MM_3	
Pin FPGA	Funcionalidad
114	Puerto de recepción de datos de la FPGA de la Arduino Nano 3.
115	Puerto digital D2 de la Arduino Nano 3. Entrada discreta 1 de la Arduino Nano 3.
119	Puerto digital D4 de la Arduino Nano 3. Entrada discreta 2 de la Arduino Nano 3.
122	Puerto digital D7 de la Arduino Nano 3. Entrada discreta 3 de la Arduino Nano 3.
125	Puerto digital D8 de la Arduino Nano 3. Salida discreta 1 de la Arduino Nano 3.
133	Puerto digital D12 de la Arduino Nano 3. Salida discreta 2 de la Arduino Nano 3.
134	Puerto digital D13 de la Arduino Nano 3. Salida discreta 3 de la Arduino Nano 3.
135	Reset de la Arduino Nano 3.

Tabla 117. Puertos de la FPGA conectados al módulo secundario 4.

Conector P_MM_4	
Pin FPGA	Funcionalidad
142	Puerto de transmisión de datos de la FPGA a la Arduino Nano 4.
143	Puerto de recepción de datos de la FPGA de la Arduino Nano 4.
4	Puerto digital D2 de la Arduino Nano 4. Entrada discreta 1 de la Arduino Nano 4.
18	Puerto digital D4 de la Arduino Nano 4. Entrada discreta 2 de la Arduino Nano 4.
25	Puerto digital D7 de la Arduino Nano 4. Entrada discreta 3 de la Arduino Nano 4.
24	Puerto digital D8 de la Arduino Nano 4. Salida discreta 1 de la Arduino Nano 4.
28	Puerto digital D12 de la Arduino Nano 4. Salida discreta 2 de la Arduino Nano 4.
32	Puerto digital D13 de la Arduino Nano 4. Salida discreta 3 de la Arduino Nano 4.
31	Reset de la Arduino Nano 4.

Para la versión 3 de firmware se eliminan los bloques funcionales UART Shield, Comunicación discreta Shield, y Monitorización PWM.

La Figura 111 muestra un diagrama de comunicaciones entre cada uno de los bloques funcionales mencionados. También se muestran las conexiones entre dichos bloques funcionales y los componentes hardware con los que interactúan.

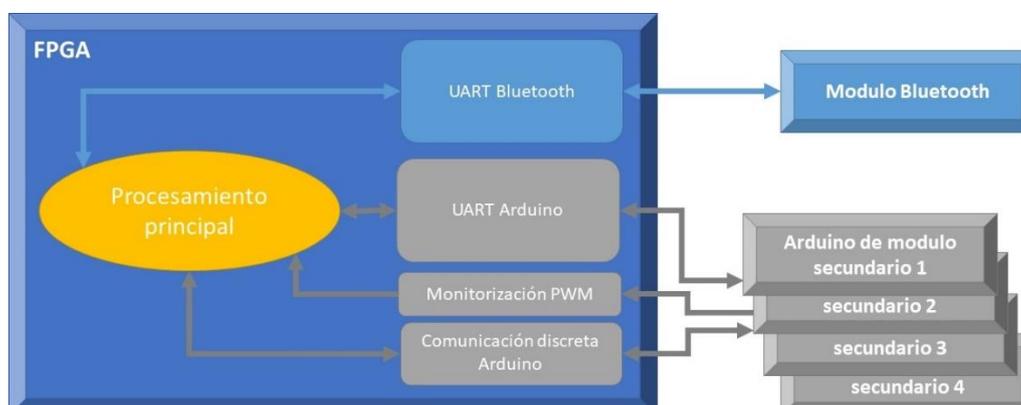


Figura 111. Bloques funcionales de la versión 3 de firmware.

El resto de los bloques funcionales permanecen tal y como se han descrito en para las versiones de firmware 1 y 2.

El diseño jerárquico de la versión de firmware 3 también se modifica de acuerdo con los cambios indicados

La Figura 112 muestra el diseño jerárquico Top-Down indicando cada uno de los componentes y el nivel jerárquico en el que están.

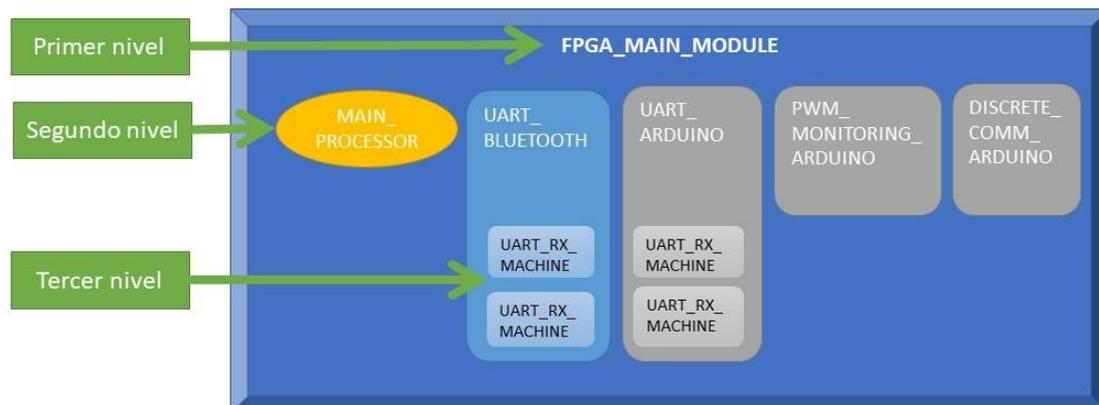


Figura 112. Diseño jerárquico Top-Down de la versión 3 de firmware.

Los bloques para instanciar son los siguientes en la versión 3 de firmware:

- MAIN_PROCESSOR.
- UART_BLUETOOTH:
 - UART_RX_MACHINE.
 - UART_TX_MACHINE.
- UART_ARDUINO:
 - UART_RX_MACHINE.
 - UART_TX_MACHINE.
- PWM_MONITORING_ARDUINO.
- DISCRETE_COMM_ARDUINO.

La Figura 113 representa el diagrama de bloques del sistema completo. Asimismo, hemos incluido las líneas de comunicaciones entre los bloques internos de la FPGA. Además, este diagrama contiene las comunicaciones entre el hardware externo a la FPGA y los bloques internos de la FPGA.

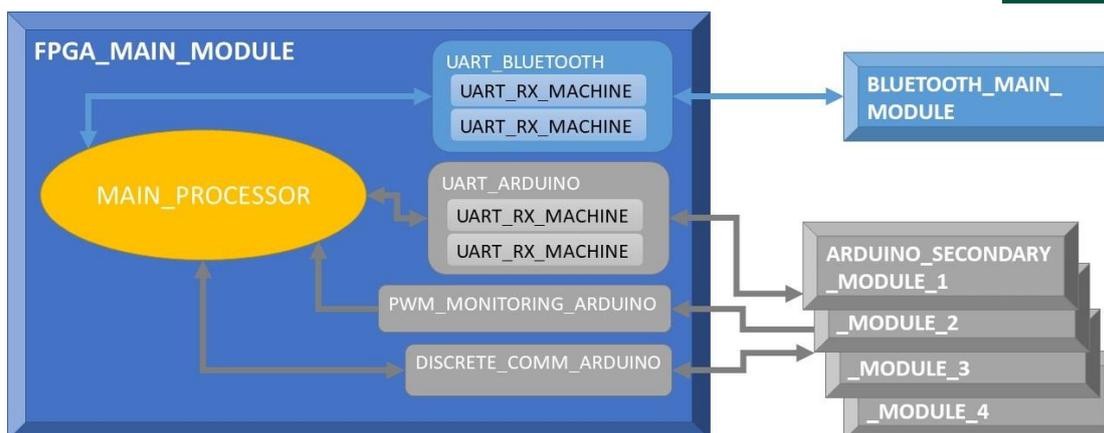


Figura 113. Diagrama de bloques del sistema completo de la versión 3 de firmware.

El bloque MAIN_PROCESSOR para la versión 3 de firmware ya no tiene en cuenta las funcionalidades asociadas a los bloques eliminados UART_SHIELD DISCRETE_COMM_SHIELD y PWM_MONITORING_ARDUINO.

La Tabla 118 contiene las señales de entrada/salida del bloque MAIN_PROCESSOR, el sentido de la señal (entrada o salida) y su tamaño, el bloque con el que se conecta y la descripción de la funcionalidad que desempeña.

Tabla 118. Señales de entrada/salida del bloque MAIN_PROCESSOR, versión 3 de firmware.

Señales de entrada/salida del bloque MAIN_PROCESSOR			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
UART_BT_IN_FRAME	Entrada 8 bits	UART_BLUETOOTH	Datos recibidos por el bloque UART Bluetooth.
UART_BT_IN_RECEIVED	Entrada 1 bit	UART_BLUETOOTH	Nuevo dato recibido por el bloque UART Bluetooth.
UART_BT_OUT_FRAME	Salida 8 bits	UART_BLUETOOTH	Datos que enviar por el bloque UART Bluetooth.
UART_BT_OUT_SEND	Salida 1 bit	UART_BLUETOOTH	Orden para enviar datos por el bloque UART Bluetooth.
UART_AR_IN_FRAME(i)	Entrada 8 bits 4 señales	UART_ARDUINO	Datos recibidos por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.
UART_AR_IN_RECEIVED(i)	Entrada 1 bit 4 señales	UART_ARDUINO	Nuevo dato recibido por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.
UART_AR_OUT_FRAME(i)	Salida 8 bits 4 señales	UART_ARDUINO	Datos que enviar por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.
UART_AR_OUT_SEND(i)	Salida 1 bit 4 señales	UART_ARDUINO	Orden para enviar datos por el bloque UART Arduino. Hay una señal de este tipo para cada módulo secundario.

Señales de entrada/salida del bloque MAIN_PROCESSOR			
Nombre de la señal	Sentido de la señal y tamaño	Bloque con el que comunica	Funcionalidad
DISC_IN_AR_EDGE(i)	Entrada 3 bits 4 señales	DISCRETE_COMM_AR DUINO	Cambio de nivel detectado en la señal correspondiente de las señales discretas de módulo secundario.
DISC_OUT_AR_LEVEL(i)	Salida 3 bits 4 señales	DISCRETE_COMM_AR DUINO	Nivel ordenado para la señal correspondiente de las señales discretas de módulo secundario.

El resto de los bloques en la versión 3 de firmware mantienen las mismas funcionalidades y los mismos puertos de entrada/salida.

La verificación por simulación para la versión 3 de firmware se repite para solamente para bloque MAIN_PROCESSOR. Pese a esto, verificación jerárquica Down-Top se modifica quitando los bloques eliminados.

La Figura 101 muestra verificación jerárquica Down-Top indicando cada uno de los componentes y el nivel jerárquico en el que están.

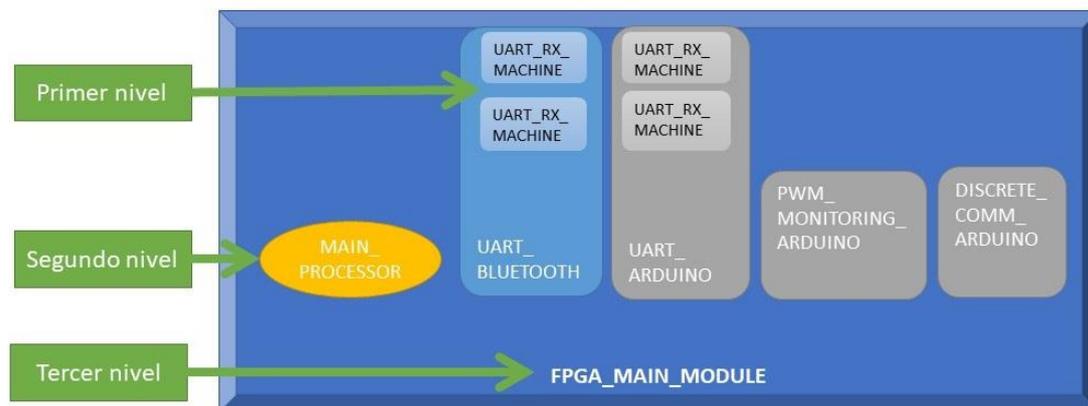


Figura 114. Verificación jerárquica Down-Top de la versión 3 de firmware.

MAIN_PROCESSOR

En el bloque MAIN_PROCESSOR se prueban las siguientes funcionalidades:

- Para estas pruebas se implementa un conversor de binario de 8 bits a código ASCII para poder añadir en el archivo de informe de resultados la información enviada por el bloque MAIN_PROCESSOR en formato ASCII, tanto para el canal de comunicaciones UART_BLUETOOTH, como para los canales de comunicaciones UART_ARDUINO.

- Se implementan relojes de transmisión para los puertos de comunicaciones de UART_BLUETOOTH y UART_ARDUINO. De esta manera, en las formas de onda se puede ver cuándo está transmitiendo cada uno de los canales de transmisión.
- Se implementan canales de transmisión para los puertos de comunicaciones de UART_BLUETOOTH y UART_ARDUINO. Almacenando los datos recibidos en mensajes de tipo binario de 8 bits con el tamaño correspondiente para cada uno de ellos.
- Se prueba la recepción del mensaje UART_BT_START_RESPONSE activando la entrada UART_BT_IN_RECEIVED.
- Se prueba la recepción del mensaje ARDUINO_STATUS_CHAR activando las entradas UART_AR_IN_RECEIVED(0), UART_AR_IN_RECEIVED(1), UART_AR_IN_RECEIVED(2) y UART_AR_IN_RECEIVED(3).
- Se prueban las entradas discretas (DISC_IN_AR_EDGE(0), ..., DISC_IN_AR_EDGE(11)) generando un pulso a nivel activo durante un ciclo de reloj y se espera a que se ponga a '1' el puerto de salida correspondiente (DISC_OUT_AR_LEVEL(0), ..., DISC_OUT_AR_LEVEL(11)).

Hay que indicar que se usa un MAIN_PROCESSOR_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama MAIN_PROCESSOR_TB.vhd. El informe de resultados es el archivo MAIN_PROCESSOR_RESULTS.txt.

FPGA_MAIN_MODULE

En el bloque MAIN_PROCESSOR se prueban las siguientes funcionalidades:

- Para estas pruebas se implementa un conversor de binario de 8 bits a código ASCII para poder añadir en el archivo de informe de resultados la información enviada por el bloque MAIN_PROCESSOR en formato ASCII, tanto para el canal de comunicaciones UART_BLUETOOTH, como para los canales de comunicaciones UART_ARDUINO.
- Se implementan relojes de transmisión para los puertos de comunicaciones de UART_BLUETOOTH y UART_ARDUINO. De esta manera, en las formas de onda se puede ver cuándo está transmitiendo cada uno de los canales de transmisión.
- Se implementan canales de transmisión para los puertos de comunicaciones de UART_BLUETOOTH y UART_ARDUINO. Almacenando los datos recibidos en mensajes de tipo binario de 8 bits con el tamaño correspondiente para cada uno de ellos.
- Se prueba la recepción del mensaje UART_BT_START_RESPONSE enviando la trama "10101010" a través de la entrada UART_BT_RX.

- Se prueba la recepción del mensaje ARDUINO_UPDATE enviando la trama "10101010" a través de las entradas UART_AR1_RX, UART_AR2_RX, UART_AR3_RX y UART_AR4_RX.
- Se prueban las entradas discretas (DISC_IN_1_AR4_D2, DISC_IN_2_AR4_D4 y DISC_IN_3_AR4_D7) poniendo a nivel activo y se comprueba que el puerto de salida correspondiente esté a nivel activo (DISC_OUT_1_AR4_D8, DISC_OUT_2_AR4_D12 y DISC_OUT_3_AR4_D13).
- Se prueban las entradas discretas (DISC_IN_1_AR4_D2, DISC_IN_2_AR4_D4 y DISC_IN_3_AR4_D7) poniendo a nivel inactivo y se comprueba que el puerto de salida correspondiente esté a nivel inactivo (DISC_OUT_1_AR4_D8, DISC_OUT_2_AR4_D12 y DISC_OUT_3_AR4_D13).

Se usa un FPGA_MAIN_MODULE_TB.do para ejecutar la simulación. El archivo fuente que implementa las mencionadas funcionalidades se llama FPGA_MAIN_MODULE_TB.vhd. El informe de resultados es el archivo FPGA_MAIN_MODULE_RESULTS.txt.

Resultados de la integración en FPGA

Una vez finalizada la implementación con la herramienta Quartus II se generan los informes de uso de recursos. La Tabla 119 muestra el resumen de uso de recursos de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 119. Resumen de uso de recursos de Análisis y Síntesis, versión 3 de firmware.

Resumen de uso de recursos de Análisis y Síntesis	
Recurso	Uso
Registros lógicos totales estimados	972 / 4,608 (21 %)
Combinacional sin registrar	499
Sólo registros	24
Combinacional registrada	449
Uso de elementos lógicos por número de entradas de LUT	
-- Funciones de 4 entradas	389
-- Funciones de 3 entradas	149
-- Funciones de menos de 2 entradas	410
-- Sólo registrados	24
Elementos lógicos por modo	
-- Modo normal	813
-- Modo aritmético	135
Registros totales	473 / 4851 (10 %)
-- Registros lógicos dedicados	473 / 4608 (10 %)
-- Registros de Entrada/Salida	0 / 243 (0 %)
Pines de Entrada/Salida	86 / 89 (97 %)
Nodo de máximo fan-out	CLK_50MHZ

Resumen de uso de recursos de Análisis y Síntesis	
Recurso	Uso
Fan-out máximo	473
Fan-out total	4058
Fan-out promedio	2,66

La Tabla 120 muestra el resumen de uso de recursos por entidad de la FPGA utilizados obtenidos de la herramienta Quartus II.

Tabla 120. Uso de recursos por entidad de Análisis y Síntesis, versión 3 de firmware.

Resumen de uso de recursos por entidad de Análisis y Síntesis		
Entidad jerárquica	Celdas Lógicas	Registros
FPGA_MAIN_MODULE	972 (63)	473 (29)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_1	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_2	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_3	9 (9)	9 (9)
DISCRETE_COMM_ARDUINO:INST_DISCRETE_COMM_ARDUINO_4	9 (9)	9 (9)
MAIN_PROCESSOR:INST_MAIN_PROCESSOR	348 (348)	170 (170)
UART_ARDUINO:INST_UART_ARDUINO_1	107 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	50 (50)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_2	110 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	53 (53)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_3	107 (0)	49 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	50 (50)	23 (23)
UART_ARDUINO:INST_UART_ARDUINO_4	110 (0)	49 (0)

Resumen de uso de recursos por entidad de Análisis y Síntesis		
Entidad jerárquica	Celdas Lógicas	Registros
UART_RX_MACHINE:INST_UART_RX_MACHINE	57 (57)	26 (26)
UART_TX_MACHINE:INST_UART_TX_MACHINE	53 (53)	23 (23)
UART_BLUETOOTH:INST_UART_BLUETOOTH	91 (0)	42 (0)
UART_RX_MACHINE:INST_UART_RX_MACHINE	38 (38)	18 (18)
UART_TX_MACHINE:INST_UART_TX_MACHINE	53 (53)	24 (24)

Pruebas de integración hardware-firmware

Tras obtener la versión 3 del firmware se procede a realizar las pruebas de integración hardware-firmware descritas en la sección 7.1.6:

1. Se comprueba que los LED de estado de la FPGA se comportan correctamente.
2. Se comprueba que la FPGA y el módulo bluetooth se comunican correctamente.
3. Se comprueba que el módulo principal es capaz de comunicarse con un PC conectado por bluetooth.

Una vez concluidas las pruebas se comprueba que la versión 3 del firmware está lista para ser usada en las pruebas de integración hardware-firmware-software – ver sección 8.3.

CAPÍTULO 8

8. DESARROLLO SOFTWARE

En este capítulo se recoge toda la información recopilada durante diseño firmware del desarrollo de la arquitectura de la plataforma de robótica educativa. De acuerdo con la arquitectura definida en el capítulo 5, es necesario el desarrollo de software para que las Arduino Nano que forman parte del sistema gestionen las funcionalidades requeridas por parte del software.

8.1. Metodología de desarrollo software

El desarrollo software de la plataforma se encuentra ubicado en la etapa 5, referente a la implementación de la Metodología de Modelo en V usado a lo largo de la elaboración de la arquitectura de la Plataforma, tal y como se describe en el capítulo 5 a lo largo de la sección 3.

Para el desarrollo software de la Plataforma Robótica Educativa se ha elaborado una metodología de desarrollo de software. Esta metodología consta de varios pasos: identificación de las especificaciones, definición de los bloques funcionales, codificación del software y verificación del software de las Arduino de los módulos secundarios.

En primer lugar, se indican las especificaciones identificadas que tienen relación con el software. Posteriormente se identifican los bloques funcionales que van a ser implementados en la Arduino y que serán fundamentales a la hora de codificar las funcionalidades especificadas. En tercer lugar, se procede a la elaboración de los archivos fuente que describen mediante un lenguaje de programación basado en C dichos bloques funcionales. Una vez concluidas los pasos anteriormente mencionados, se compila el proyecto en el interfaz de desarrollo de Arduino para obtener los archivos de programación. En este momento, se realizan pruebas de verificación para comprobar que el diseño software se ajusta a las especificaciones. Para ello, será necesario contar con diferentes elementos hardware, y en este paso se realizará una integración total de la Plataforma, integración hardware-firmware-software.

Para codificar el software, grabar los programas en las Arduino y realizar tareas de depuración a través del puerto serie se usará el interfaz de desarrollo de Arduino. La Figura 115 muestra un ejemplo del interfaz de desarrollo de Arduino en el que se identifica la parte del código fuente que contiene la configuración de la Arduino (función void setup)

y la parte del código fuente que contiene el programa que ejecuta la Arduino (función void loop).

The image shows a screenshot of the Arduino IDE interface. At the top, the title bar reads 'sketch_nov08a Arduino 1.6.7'. Below the title bar is a menu bar with 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. A toolbar with icons for file operations is visible. The main editor area shows the following code:

```
sketch_nov08a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Figura 115. Interfaz de desarrollo de Arduino.

Si se detectase algún error de diseño o incumplimiento de cualquier especificación, se evaluará si es posible continuar con las siguientes etapas, y si fuese necesario, se realizarán tareas de rediseño hasta que se obtenga una versión de software que cumpla con las especificaciones fijadas.

Las sucesivas secciones muestran toda la información de diseño obtenida a lo largo del desarrollo de las versiones de software, incluyendo errores de diseño o funcionalidades que no se cumplen, también se indica el momento o paso en el que se detectaron y soluciones propuestas para la correspondiente subsanación.

8.1.1. PASO 1: ESPECIFICACIONES SOFTWARE

Las especificaciones de software es un conjunto de requisitos a nivel funcional y no funcional que un sistema debe cumplir. No solo restricciones, sino características, funcionalidades, etc. Estas especificaciones se usan para la toma de decisiones de diseño en las siguientes etapas. Estas decisiones de diseño se orientan, por un lado, en la elección de los puertos de entrada/salida, y, por otro lado, las funcionalidades asociadas a dichos puertos.

Del diseño hardware se obtienen las conexiones que hay entre las Arduinos de los módulos secundarios y la FPGA del módulo principal. Cada Arduino tiene un puerto de comunicaciones serie conectado a la FPGA. Además, los puertos de cada Arduino D0, ..., D13 están conectados a la FPGA. Por otro lado, cada Arduino tiene todas las conexiones propias que tienen las tarjetas Arduino con los Shields compatibles con Arduino. La Tabla 121 describe los puertos para comunicación serie usados en las tarjetas Arduino y la funcionalidad de cada uno de ellos.

Tabla 121. Puertos de comunicación serie de las Arduino.

Nombre del puerto	Funcionalidad
D0	Puerto de recepción de datos de la Arduino para las comunicaciones serie.
D1	Puerto de transmisión de datos de la Arduino para las comunicaciones serie.

La Tabla 122 describe los puertos PWM usados en las tarjetas Arduino y la funcionalidad de cada uno de ellos.

Tabla 122. Puertos PWM de las Arduino.

Nombre del puerto	Funcionalidad
D3	Puerto PWM 1 para generar pulsos PWM.
D5	Puerto PWM 2 para generar pulsos PWM.
D6	Puerto PWM 3 para generar pulsos PWM.
D9	Puerto PWM 4 para generar pulsos PWM.
D10	Puerto PWM 5 para generar pulsos PWM.
D11	Puerto PWM 6 para generar pulsos PWM.

Del diseño firmware se obtienen las interconexiones entre las Arduino y la FPGA, además de los canales de comunicación discreta entre las Arduino. La Tabla 123 describe las conexiones discretas entre las tarjetas Arduino y la funcionalidad de cada uno de ellos.

Tabla 123. Puertos de la comunicación discreta Arduino.

Identificador de la tarjeta Arduino	Nombre del puerto	Funcionalidad
Arduino 1	D2	Entrada discreta 1. Tiene conexión con la salida discreta 1 de la Arduino 4. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D4	Entrada discreta 2. Tiene conexión con la salida discreta 2 de la Arduino 2. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D7	Entrada discreta 3. Tiene conexión con la salida discreta 3 de la Arduino 3. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D8	Salida discreta 1. Tiene conexión con la entrada discreta 1 de la Arduino 2. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D12	Salida discreta 2. Tiene conexión con la entrada discreta 2 de la Arduino 4. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D13	Salida discreta 3. Tiene conexión con la entrada discreta 3 de la Arduino 3. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
Arduino 2	D2	Entrada discreta 1. Tiene conexión con la salida discreta 1 de la Arduino 1. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.

Identificador de la tarjeta Arduino	Nombre del puerto	Funcionalidad
	D4	Entrada discreta 2. Tiene conexión con la salida discreta 2 de la Arduino 3. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D7	Entrada discreta 3. Tiene conexión con la salida discreta 3 de la Arduino 4. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D8	Salida discreta 1. Tiene conexión con la entrada discreta 1 de la Arduino 3. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D12	Salida discreta 2. Tiene conexión con la entrada discreta 2 de la Arduino 1. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D13	Salida discreta 3. Tiene conexión con la entrada discreta 3 de la Arduino 4. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
Arduino 3	D2	Entrada discreta 1. Tiene conexión con la salida discreta 1 de la Arduino 2. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D4	Entrada discreta 2. Tiene conexión con la salida discreta 2 de la Arduino 4. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D7	Entrada discreta 3. Tiene conexión con la salida discreta 3 de la Arduino 1. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D8	Salida discreta 1. Tiene conexión con la entrada discreta 1 de la Arduino 4. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D12	Salida discreta 2. Tiene conexión con la entrada discreta 2 de la Arduino 2. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D13	Salida discreta 3. Tiene conexión con la entrada discreta 3 de la Arduino 1. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
Arduino 4	D2	Entrada discreta 1. Tiene conexión con la salida discreta 1 de la Arduino 3. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D4	Entrada discreta 2. Tiene conexión con la salida discreta 2 de la Arduino 1. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D7	Entrada discreta 3. Tiene conexión con la salida discreta 3 de la Arduino 2. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D8	Salida discreta 1. Tiene conexión con la entrada discreta 1 de la Arduino 1. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D12	Salida discreta 2. Tiene conexión con la entrada discreta 2 de la Arduino 3. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.
	D13	Salida discreta 3. Tiene conexión con la entrada discreta 3 de la Arduino 2. Su uso es para aplicaciones genéricas en el contexto de la lógica digital.

El software de cada Arduino deberá ser capaz de:

- Mandar un mensaje de identificación por el puerto serie. Este mensaje se manda con una periodicidad de un segundo.
- Generar señales PWM para controlar un servo respecto a la posición relativa de la Arduino. De manera que la Arduino 1 con la salida PWM 1 colocará el servo en la posición de 30 grados. El ángulo será modificado de la siguiente manera:
 - Incrementará 5 grados la posición después del primer segundo.
 - Incrementará 5 grados la posición un segundo después.
 - Incrementará 5 grados la posición un segundo después.
 - Reiniciará la posición un segundo después.
- Cada Arduino generará:
 - Al inicio pondrá las tres salidas discretas a nivel bajo.
 - Cambiará a nivel alto su salida discreta 1 después del primer segundo.
 - Cambiará a nivel alto su salida discreta 2 un segundo después que la salida discreta 1.
 - Cambiará a nivel alto su salida discreta 3 un segundo después que la salida discreta 2.
 - Reiniciará el contador temporal un segundo después que la salida discreta 3 haya cambiado a nivel alto.

La Figura 116 muestra un esquema de las funcionalidades indicadas en las especificaciones de software y los elementos externos a las Arduino con los que interactúan dichas funcionalidades.

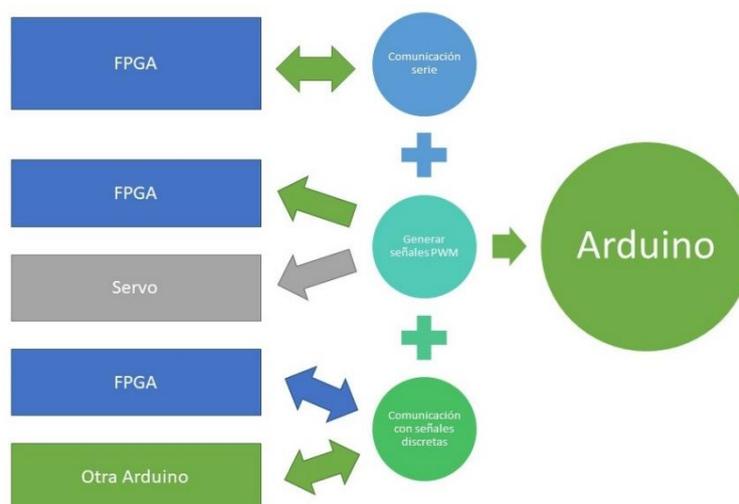


Figura 116. Esquema de funcionalidades de las Arduino.

La Figura 117 muestra un diagrama de flujo de la comunicación basada en señales discretas y el comportamiento de la Arduino cuando implementa esta funcionalidad. Este comportamiento tiene en cuenta el valor de inicio de las señales y el estado de las mismas a lo largo del tiempo.



Figura 117. Diagrama de flujo de la comunicación con señales discretas.

8.1.2. PASO 2: BLOQUES FUNCIONALES

Para cumplir con las especificaciones (Comunicación serie, generación de señales PWM y comunicación con señales discretas) se definen los siguientes bloques funcionales:

- UART para comunicaciones series usando el puerto nativo serie de la Arduino.
- PWM para generar las señales de control de servos usando la biblioteca Servo.h.
- Control de entradas y salidas discretas.

8.1.3. PASO 3: CODIFICACIÓN DEL SOFTWARE

El software Arduino está compuesto de dos partes. Por un lado, la configuración, que es común para las cuatro Arduino. Y, por otro lado, el hilo de ejecución que es independiente para cada una de las Arduino.

Configuración

Como se ha mencionado anteriormente, la configuración de las cuatro placas Arduino es común. En primer lugar, el puerto de comunicación serie se configura con una velocidad de 9600 baudios.

Por otro lado, se define la configuración relacionada con los puertos PWM. Cada una de las Arduino tiene seis objetos del tipo Servo, uno para cada una de las salidas PWM. Cada una de ellas se nombra como PWM_1, ..., PWM_6. Se usan constantes de tipo entero para definir los puertos de entrada/salida discretos. La Tabla 124 describe los puertos PWM usados en las tarjetas Arduino.

Tabla 124. Constantes para definir las salidas PWM.

Nombre de la constante	Valor
PWM_1	3
PWM_2	5
PWM_3	6
PWM_4	9
PWM_5	10
PWM_6	11

Adicionalmente, se define una serie de constantes para ser usadas en la gestión de las señales PWM. La Tabla 125 describe las constantes relacionadas con la generación de los pulsos PWM, su valor y su uso.

Tabla 125. Constantes relacionadas con la generación de los pulsos PWM.

Nombre de la constante	Valor	Descripción
ARDUINO_ID	1	Identificador de la tarjeta Arduino.
	2	
	3	
	4	
SERVO_OFFSET	30	Offset para indicar el ángulo mínimo (30 grados).
SERVO_STEP	5	Paso para variar el ángulo de giro (5 grados).

También se añaden variables que son usadas a lo largo del hilo de ejecución del programa. La Tabla 126 incluye estas variables, su valor inicial y una breve descripción de cada una de ellas.

Tabla 126. Variables de propósito general.

Nombre de la variable	Valor	Descripción
initialized	0	Variable para detectar el arranque de la Arduino.
seconds_counter	0	Variable para almacenar los incrementos temporales en segundos.

Los puertos PWM se configuran con un rango entre 771 μ s y 1798 μ s.

Por último, los puertos de entrada discreta y los puertos de salida discreta se configuran usando las constantes que la Tabla 127 detalla.

Tabla 127. Constantes para definir las entradas/salidas discretas.

Nombre de la constante	Valor
INPUT_1	2
INPUT_2	4
INPUT_3	7
OUTPUT_1	8

Nombre de la constante	Valor
OUTPUT_2	12
OUTPUT_3	13

A continuación, se detallan los hilos de ejecución para cada Arduino. Los hilos de ejecución son diferentes para cada una de las Arduino.

Hilo de ejecución de la Arduino 1

En primer lugar, se aplica los valores de los pulsos PWM a cada una de las salidas. Para ello, se le asignan los valores que la Tabla 128 incluye.

Tabla 128. Grados asignados a cada uno de los puertos PWM de la Arduino 1.

Nombre del puerto PWM	Valor en grados
PWM_1	35
PWM_2	40
PWM_3	45
PWM_4	50
PWM_5	55
PWM_6	60

En cada ejecución del bucle principal del programa (cada segundo), estos valores se irán incrementado en 5 grados. De esta manera el puerto PWM_1 generará pulsos para un giro de 35 grados, un segundo después generará pulsos para un giro de 40 grados, un segundo después generará pulsos para un giro de 45 grados, un segundo después generará pulsos para un giro de 50 grados y, por último, volverá a generar pulsos para un giro de 35 grados. El resto de los puertos se comportará de manera similar, pero el valor de giro inicial será el definido por la Tabla 128. La Figura 118 muestra un diagrama de flujo de la ejecución del hilo de ejecución de la Arduino 1 para la gestión de la señal PWM_1.

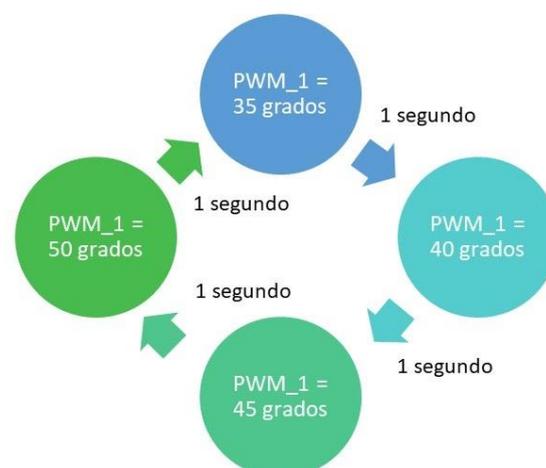


Figura 118. Diagrama de flujo PWM del hilo de ejecución de la Arduino 1.

A continuación, si es la primera vez que se recorre el bucle principal, la Arduino indicará por el puerto serie que ha sido inicializada, e inicializará las tres salidas discretas a nivel bajo.

Posteriormente, a la variable initialized se le dará un valor de 1 para indicar que ya ha sido inicializada.

También, se fija el valor de las salidas discretas en función de los segundos que hayan transcurrido desde el arranque. La Tabla 129 describe los valores de los puertos discretos en función del tiempo.

Tabla 129. Valores de los puertos discretos en función del tiempo de la Arduino 1.

Nombre del puerto	Valor en tiempo $t = 0$ s	Valor en tiempo $t = 1$ s	Valor en tiempo $t = 2$ s	Valor en tiempo $t = 3$ s
OUTPUT_1	Nivel bajo	Nivel alto	Nivel alto	Nivel alto
OUTPUT_2	Nivel bajo	Nivel bajo	Nivel alto	Nivel alto
OUTPUT_3	Nivel bajo	Nivel bajo	Nivel bajo	Nivel alto

Al final del bucle principal, se controla el tiempo de ejecución con la variable `seconds_counter`. En la primera ejecución del bucle esta variable tiene un valor de 0. En la segunda ejecución tiene un valor de 1, y así sucesivamente hasta que la variable ha alcanzado el valor de 3. En este momento el valor de la variable se inicializa a 0 de nuevo. La Figura 119 muestra un diagrama de flujo de la ejecución del hilo de ejecución de la Arduino 1 para la gestión de las señales discretas.

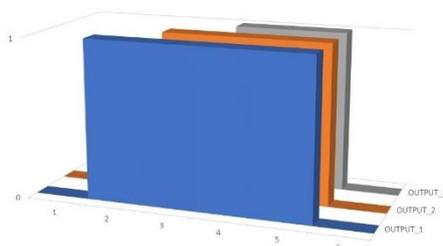


Figura 119. Respuesta temporal de señales discretas del hilo de ejecución de la Arduino 1.

El Sketch usa 4266 bytes (13%) del espacio de almacenamiento de programa. El máximo es 30720 bytes. Las variables Globales usan 564 bytes (27%) de la memoria dinámica, dejando 1484 bytes para las variables locales. El máximo es 2048 bytes.

Hilo de ejecución de la Arduino 2

En primer lugar, se aplica los valores de los pulsos PWM a cada una de las salidas. Para ello, se le asignan los valores que incluye la Tabla 130.

Tabla 130. Grados asignados a cada uno de los puertos PWM de la Arduino 2.

Nombre del puerto PWM	Valor en grados
PWM_1	65
PWM_2	70
PWM_3	75
PWM_4	80
PWM_5	85
PWM_6	90

En cada ejecución del bucle principal del programa (cada segundo), estos valores se irán incrementado en 5 grados. De esta manera el puerto PWM_1 generará pulsos para un giro de 65 grados, un segundo después generará pulsos para un giro de 70 grados, un segundo después generará pulsos para un giro de 75 grados, un segundo después generará pulsos para un giro de 80 grados y, por último, volverá a generar pulsos para un giro de 65 grados. El resto de los puertos se comportará de manera similar, pero el valor de giro inicial será el definido por la Tabla 130. La Figura 120 muestra un diagrama de flujo de la ejecución del hilo de ejecución de la Arduino 2 para la gestión de la señal PWM_1.

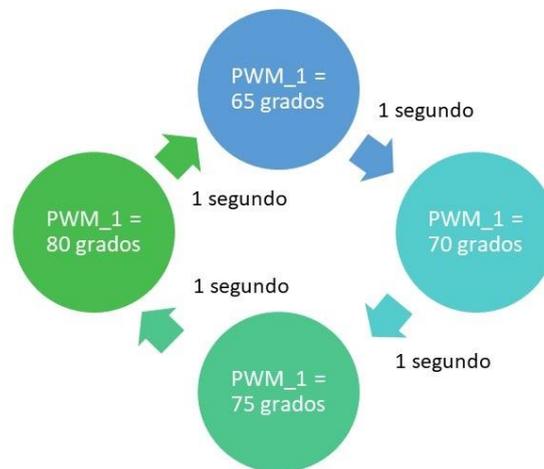


Figura 120. Diagrama de flujo PWM del hilo de ejecución de la Arduino 2.

A continuación, si es la primera vez que se recorre el bucle principal, la Arduino indicará por el puerto serie que ha sido inicializada, e inicializará las tres salidas discretas a nivel bajo.

Posteriormente, a la variable initialized se le dará un valor de 1 para indicar que ya ha sido inicializada.

También, se fija el valor de las salidas discretas en función de los segundos que hayan transcurrido desde el arranque. La Tabla 131 describe los valores de los puertos discretos en función del tiempo.

Tabla 131. Valores de los puertos discretos en función del tiempo de la Arduino 2.

Nombre del puerto	Valor en tiempo t = 0 s	Valor en tiempo t = 1 s	Valor en tiempo t = 2 s	Valor en tiempo t = 3 s
OUTPUT_1	Nivel bajo	Nivel alto	Nivel alto	Nivel alto
OUTPUT_2	Nivel bajo	Nivel bajo	Nivel alto	Nivel alto
OUTPUT_3	Nivel bajo	Nivel bajo	Nivel bajo	Nivel alto

Al final del bucle principal, se controla el tiempo de ejecución con la variable `seconds_counter`. En la primera ejecución del bucle esta variable tiene un valor de 0. En la segunda ejecución tiene un valor de 1, y así sucesivamente hasta que la variable ha alcanzado el valor de 3. En este momento el valor de la variable se inicializa a 0 de nuevo. El comportamiento de las señales discretas manejadas por la Arduino 2 es igual al mostrado en la Figura 119.

El Sketch usa 4266 bytes (13%) del espacio de almacenamiento de programa. El máximo es 30720 bytes. Las variables Globales usan 564 bytes (27%) de la memoria dinámica, dejando 1484 bytes para las variables locales. El máximo es 2048 bytes.

Hilo de ejecución de la Arduino 3

En primer lugar, se aplica los valores de los pulsos PWM a cada una de las salidas. Para ello, se le asignan los valores que incluye la Tabla 132.

Tabla 132. Grados asignados a cada uno de los puertos PWM de la Arduino 3.

Nombre del puerto PWM	Valor en grados
PWM_1	95
PWM_2	100
PWM_3	105
PWM_4	110
PWM_5	115
PWM_6	120

En cada ejecución del bucle principal del programa (cada segundo), estos valores se irán incrementado en 5 grados. De esta manera el puerto PWM_1 generará pulsos para un giro de 95 grados, un segundo después generará pulsos para un giro de 100 grados, un

segundo después generará pulsos para un giro de 105 grados, un segundo después generará pulsos para un giro de 110 grados y, por último, volverá a generar pulsos para un giro de 95 grados. El resto de los puertos se comportará de manera similar, pero el valor de giro inicial será el definido por la Tabla 132. La Figura 121 muestra un diagrama de flujo de la ejecución del hilo de ejecución de la Arduino 3 para la gestión de la señal PWM_1.

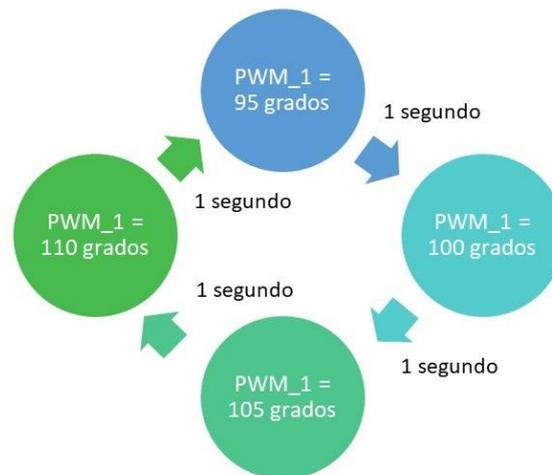


Figura 121. Diagrama de flujo PWM del hilo de ejecución de la Arduino 3.

A continuación, si es la primera vez que se recorre el bucle principal, la Arduino indicará por el puerto serie que ha sido inicializada, e inicializará las tres salidas discretas a nivel bajo.

Posteriormente, a la variable initialized se le dará un valor de 1 para indicar que ya ha sido inicializada.

También, se fija el valor de las salidas discretas en función de los segundos que hayan transcurrido desde el arranque. La Tabla 133 describe los valores de los puertos discretos en función del tiempo.

Tabla 133. Valores de los puertos discretos en función del tiempo de la Arduino 3.

Nombre del puerto	Valor en tiempo t = 0 s	Valor en tiempo t = 1 s	Valor en tiempo t = 2 s	Valor en tiempo t = 3 s
OUTPUT_1	Nivel bajo	Nivel alto	Nivel alto	Nivel alto
OUTPUT_2	Nivel bajo	Nivel bajo	Nivel alto	Nivel alto
OUTPUT_3	Nivel bajo	Nivel bajo	Nivel bajo	Nivel alto

Al final del bucle principal, se controla el tiempo de ejecución con la variable seconds_counter. En la primera ejecución del bucle esta variable tiene un valor de 0. En la

segunda ejecución tiene un valor de 1, y así sucesivamente hasta que la variable ha alcanzado el valor de 3. En este momento el valor de la variable se inicializa a 0 de nuevo. El comportamiento de las señales discretas manejadas por la Arduino 3 es igual al mostrado en la Figura 119.

El Sketch usa 4266 bytes (13%) del espacio de almacenamiento de programa. El máximo es 30720 bytes. Las variables Globales usan 564 bytes (27%) de la memoria dinámica, dejando 1484 bytes para las variables locales. El máximo es 2048 bytes.

Hilo de ejecución de la Arduino 4

En primer lugar, se aplica los valores de los pulsos PWM a cada una de las salidas. Para ello, se le asignan los valores que incluye la Tabla 134.

Tabla 134. Grados asignados a cada uno de los puertos PWM de la Arduino 4.

Nombre del puerto PWM	Valor en grados
PWM_1	125
PWM_2	130
PWM_3	135
PWM_4	140
PWM_5	145
PWM_6	150

En cada ejecución del bucle principal del programa (cada segundo), estos valores se irán incrementado en 5 grados. De esta manera el puerto PWM_1 generará pulsos para un giro de 125 grados, un segundo después generará pulsos para un giro de 130 grados, un segundo después generará pulsos para un giro de 135 grados, un segundo después generará pulsos para un giro de 140 grados y, por último, volverá a generar pulsos para un giro de 125 grados. El resto de los puertos se comportará de manera similar, pero el valor de giro inicial será el definido por la Tabla 134. La Figura 122 muestra un diagrama de flujo de la ejecución del hilo de ejecución de la Arduino 4 para la gestión de la señal PWM_1.

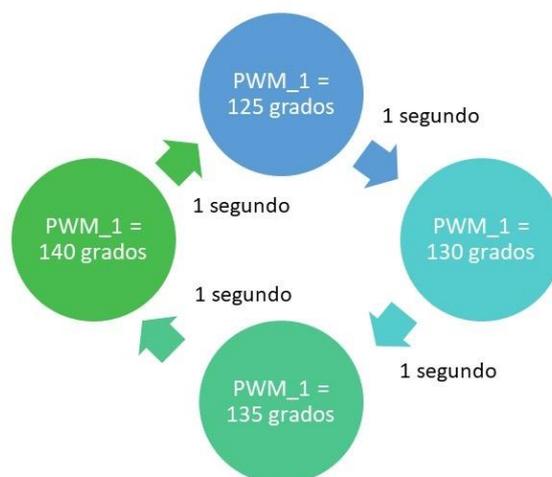


Figura 122. Diagrama de flujo PWM del hilo de ejecución de la Arduino 4.

A continuación, si es la primera vez que se recorre el bucle principal, la Arduino indicará por el puerto serie que ha sido inicializada, e inicializará las tres salidas discretas a nivel bajo.

Posteriormente, a la variable initialized se le dará un valor de 1 para indicar que y ha sido inicializada.

También, se fija el valor de las salidas discretas en función de los segundos que hayan transcurrido desde el arranque. La Tabla 135 describe los valores de los puertos discretos en función del tiempo.

Tabla 135. Valores de los puertos discretos en función del tiempo de la Arduino 4.

Nombre del puerto	Valor en tiempo t = 0 s	Valor en tiempo t = 1 s	Valor en tiempo t = 2 s	Valor en tiempo t = 3 s
OUTPUT_1	Nivel bajo	Nivel alto	Nivel alto	Nivel alto
OUTPUT_2	Nivel bajo	Nivel bajo	Nivel alto	Nivel alto
OUTPUT_3	Nivel bajo	Nivel bajo	Nivel bajo	Nivel alto

Al final del bucle principal, se controla el tiempo de ejecución con la variable seconds_counter. En la primera ejecución del bucle esta variable tiene un valor de 0. En la segunda ejecución tiene un valor de 1, y así sucesivamente hasta que la variable ha alcanzado el valor de 3. En este momento el valor de la variable se inicializa a 0 de nuevo. El comportamiento de las señales discretas manejadas por la Arduino 4 es igual al mostrado en la Figura 119.

El Sketch usa 4266 bytes (13%) del espacio de almacenamiento de programa. El máximo es 30720 bytes. Las variables Globales usan 564 bytes (27%) de la memoria dinámica, dejando 1484 bytes para las variables locales. El máximo es 2048 bytes.

8.1.4. PASO 4: VERIFICACIÓN DEL SOFTWARE

Para la realización de las pruebas de verificación del software de cada Arduino se usa la placa, el monitor serie del IDE de Arduino, cables y un LED RGB.

Con el monitor serie del IDE de Arduino se comprueba, por un lado, que el puerto está bien configurado y, por otro lado, se usa como puerto de depuración para observar los valores que tienen los puertos de entrada discretos.

La prueba de los puertos PWM son específicas del hilo de ejecución de cada Arduino

Por lo tanto, hay dos tipos de pruebas, unas pruebas que están relacionadas con la configuración (común a cada Arduino) y otras pruebas están relacionadas con el hilo de ejecución. Estas últimas al tener funcionalidades distintas y distinta programación necesitan de bancos de prueba diferentes.

Pruebas de configuración

La comprobación de las entradas discretas se realiza conectando cables que, por un lado, están conectados al puerto de 5V de la propia Arduino y, por otro lado, se van conectando a cada puerto de entrada discreto. En el monitor serie del IDE de Arduino se observa que dichos puertos toman el valor de nivel alto cuando se les conecta a al puerto de alimentación de 5 V. Si, por el contrario, se conecta cables que, por un lado, están conectados al puerto de tensión de referencia GND de la propia Arduino y, por otro lado, se van conectando a cada puerto de entrada discreto. En el monitor serie del IDE de Arduino se observa que dichos puertos toman el valor de nivel bajo cuando se les conecta a al puerto de tensión de referencia GND.

La comprobación de los puertos de salida discretos se realiza con un LED RGB. Cada uno de los pines del LED RGB se conecta a cada uno de los puertos de salida discretos. La Tabla 136 describe las conexiones entre la Arduino y el LED RGB.

Tabla 136. Conexiones entre la Arduino y el LED RGB.

Puerto Arduino	Pin LED RGB
OUTPUT_1	R
OUTPUT_2	G
OUTPUT_3	B

A medida que avanza el tiempo de ejecución se ve como el LED RGB va variando el color, mostrando cada una de las tonalidades que corresponden a la secuencia. La Tabla 137 describe la secuencia de iluminación del LED RGB durante la prueba de verificación.

Tabla 137. Secuencia de iluminación del LED RGB durante la prueba de verificación.

Instante temporal	Pin LED RGB
t = 0 s	Apagado.
t = 1 s	Luce rojo.
t = 2 s	Luce rojo más verde.
t = 3 s	Luce rojo más verde más azul.
t = 4 s	Apagado.

Pruebas hilo de ejecución de la Arduino 1

Para realizar la prueba de las salidas PWM en la Arduino 1 se conecta un servo a cada una de las salidas para observar la secuencia de giro del servo cuando es gestionado por cada una de las salidas. La Tabla 138 describe la secuencia de giro del servo durante la prueba de verificación.

Tabla 138. Secuencia de giro del servo durante la prueba de verificación para la Arduino 1.

Instante temporal	Posición del servo en grados					
	PWM_1	PWM_2	PWM_3	PWM_4	PWM_5	PWM_6
t = 0 s	35	40	45	50	55	60
t = 1 s	40	45	50	55	60	65
t = 2 s	45	50	55	60	65	70
t = 3 s	50	55	60	65	70	75
t = 4 s	55	60	65	70	75	80

Como se observa en la Tabla 138, los valores medidos concuerdan con los valores especificados en la Tabla 128 y la secuencia descrita por la Figura 118.

Pruebas hilo de ejecución de la Arduino 2

Para realizar la prueba de las salidas PWM en la Arduino 2 se conecta un servo a cada una de las salidas para observar la secuencia de giro del servo cuando es gestionado por cada una de las salidas. La Tabla 139 describe la secuencia de giro del servo durante la prueba de verificación.

Tabla 139. Secuencia de giro del servo durante la prueba de verificación para la Arduino 2.

Instante temporal	Posición del servo en grados					
	PWM_1	PWM_2	PWM_3	PWM_4	PWM_5	PWM_6
t = 0 s	65	70	75	80	85	90

Instante temporal	Posición del servo en grados					
	PWM_1	PWM_2	PWM_3	PWM_4	PWM_5	PWM_6
t = 1 s	70	75	80	85	90	95
t = 2 s	75	80	85	90	95	100
t = 3 s	80	85	90	95	100	105
t = 4 s	85	90	95	100	105	110

Como se observa en la Tabla 139, los valores medidos concuerdan con los valores especificados en la Tabla 130 y la secuencia descrita por la Figura 120.

Pruebas hilo de ejecución de la Arduino 3

Para realizar la prueba de las salidas PWM en la Arduino 3 se conecta un servo a cada una de las salidas para observar la secuencia de giro del servo cuando es gestionado por cada una de las salidas. La Tabla 140 describe la secuencia de giro del servo durante la prueba de verificación.

Tabla 140. Secuencia de giro del servo durante la prueba de verificación para la Arduino 3.

Instante temporal	Posición del servo en grados					
	PWM_1	PWM_2	PWM_3	PWM_4	PWM_5	PWM_6
t = 0 s	95	100	105	110	115	120
t = 1 s	100	105	110	115	120	125
t = 2 s	105	110	115	120	125	130
t = 3 s	110	115	120	125	130	135
t = 4 s	115	120	125	130	135	140

Como se observa en la Tabla 140, los valores medidos concuerdan con los valores especificados en la Tabla 132 y la secuencia descrita por la Figura 121.

Pruebas hilo de ejecución de la Arduino 4

Para realizar la prueba de las salidas PWM en la Arduino 4 se conecta un servo a cada una de las salidas para observar la secuencia de giro del servo cuando es gestionado por cada una de las salidas. La Tabla 141 describe la secuencia de giro del servo durante la prueba de verificación.

Tabla 141. Secuencia de giro del servo durante la prueba de verificación para la Arduino 4.

Instante temporal	Posición del servo en grados					
	PWM_1	PWM_2	PWM_3	PWM_4	PWM_5	PWM_6
t = 0 s	125	130	135	140	145	150
t = 1 s	130	135	140	145	150	155
t = 2 s	135	140	145	150	155	160
t = 3 s	140	145	150	155	160	165

Instante temporal	Posición del servo en grados					
	PWM_1	PWM_2	PWM_3	PWM_4	PWM_5	PWM_6
t = 4 s	145	150	155	160	165	170

Como se observa en la Tabla 141, los valores medidos concuerdan con los valores especificados en la Tabla 134 y la secuencia descrita por la Figura 122.

8.1.5. PASO 6: PRUEBAS DE INTEGRACIÓN HARDWARE-FIRMWARE-SOFTWARE

Las pruebas de integración hardware-firmware son pruebas que tratan de detectar la ausencia de errores de diseño firmware y en el comportamiento de la FPGA integrada en el hardware del módulo principal.

Las pruebas de integración hardware-firmware que se han de pasar son las siguientes:

1. Prueba de comunicación serie: se comprueban las comunicaciones serie entre la tarjeta Arduino y la FPGA del módulo modulo principal
2. Prueba de generación de señales PWM: se comprueba que la tarjeta Arduino es capaz de generar señales PWM y la FPGA detecta el ancho de los pulsos generados
3. Prueba de comunicación mediante señales discretas: se comprueba que las tarjetas Arduino son capaces de intercambiar información a través de los puertos de entrada/salida discretos

8.2. Primera versión de software, versión 1.0

Se trata de la primera versión de software. Esta versión permite el uso de todas las funcionalidades:

- Comunicaciones serie con módulo modulo principal
- Generación de señales PWM para poder ser monitorizadas por el módulo principal
- Comunicación mediante señales discretas con el módulo principal

Pruebas de integración hardware-firmware-software

Tras obtener la versión 1.0 del software se procede a realizar las pruebas de integración hardware-firmware-software descritas en la sección 8.1.5.

En este punto, se usa el módulo principal versión 1.2 (sección 6.4), los 4 módulos secundarios versión 1.2 (sección 6.4), la versión 2 de firmware (sección 7.3) la versión 1.0 del software (descrita en esta sección). Por la complejidad que tiene integrar la placa de la

FPGA, el módulo principal y los módulos secundarios, en un primer paso se incorpora un solo módulo secundario.

Durante las pruebas con un solo módulo secundario comprobamos un correcto funcionamiento del puerto serie de comunicaciones. A la hora de comprobar la funcionalidad de los puertos de entrada/salida discreta, observamos que no hay unos valores que concuerden con las señales aplicadas, por lo que se determina que los niveles de tensión de 3,3 Vdc de la FPGA no son compatibles con los niveles de 5 Vdc. Esto motiva un cambio en el hardware (sección 6.5). La solución propuesta como resultado de estas pruebas de integración hardware-firmware-software consiste en usar un componente que convierta los niveles lógicos de 3,3 Vdc de la FPGA a niveles lógicos de 5 Vdc de las Arduino y viceversa.

Debido a los errores detectados durante esta integración hardware-firmware-software de la versión 1.2 del módulo principal, la versión 1.2 del módulo secundario (ver sección 6.4), la versión 2 de firmware (sección 7.3) y la versión 1.0 de software (descrita en esta sección) es necesario aplicar cambios en el hardware, ver sección 6.5.

8.3. Segunda versión de software, versión 1.1

Esta versión contiene los cambios detectados como resultado de la integración hardware-firmware-software de la versión 2 del firmware con la versión 1.2 del hardware (sección 8.2). De acuerdo con la metodología descrita en el capítulo 5, los cambios aplicados para la segunda versión de software, versión 1.1 son motivados por las pruebas de integración de hardware-firmware-software (sección 8.2) y las modificaciones realizadas al hardware (sección 6.5) y al firmware (sección 7.4). La Figura 123 muestra la transición que proviene del desarrollo de firmware y afecta al desarrollo de firmware.

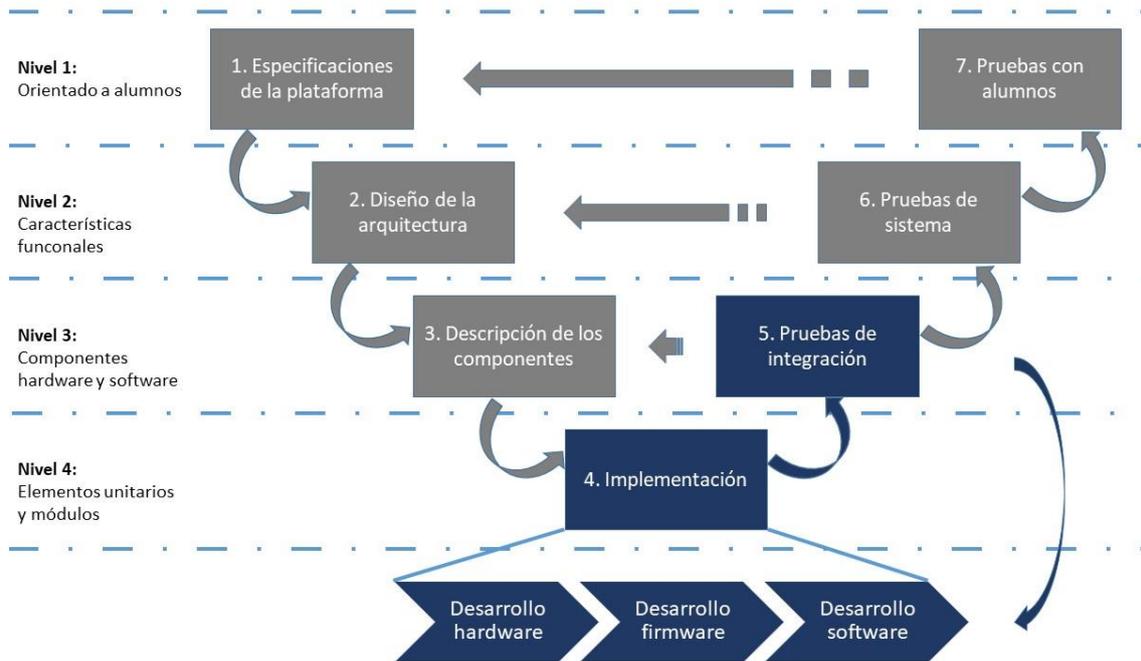


Figura 123. Diagrama de bloques del primer cambio software.

Esta versión de software contiene las mismas funcionalidades descritas en la sección 7.2 a excepción de la generación de señales PWM para poder ser monitorizadas por el módulo principal.

Pruebas de integración hardware-firmware-software

Tras obtener la versión 1.1 del software se procede a realizar las pruebas de integración hardware-firmware-software descritas en la sección 8.1.5 a excepción relativa a las señales PWM.

En este punto, se usa el módulo principal versión 1.3 (sección 6.5), los 4 módulos secundarios versión 1.3 (sección 6.5), la versión 3 de firmware (sección 7.4) y la versión 1.1 del software (descrita en esta sección). Por la complejidad que tiene integrar la placa de la FPGA, el módulo principal y los módulos secundarios, en un primer paso se incorpora un solo módulo secundario:

1. Se comprueba que las comunicaciones serie entre la tarjeta Arduino y la FPGA del módulo modulo principal funcionan correctamente

2. Se comprueba que las tarjetas Arduino son capaces de intercambiar información a través de los puertos de entrada/salida discretos

Una vez concluidas las pruebas se comprueba que el sistema compuesto por el módulo principal versión 1.3 (sección 6.5), los 4 módulos secundarios versión 1.3 (sección 6.5), la versión 3 de firmware (sección 7.4) y la versión 1.1 del software (descrita en esta sección) puede usarse en pruebas con alumnos.

CAPÍTULO 9

9. PRUEBAS CON ALUMNOS DE LA PLATAFORMA

Con el objetivo de validar la plataforma desarrollada en su primera versión que lo forman el módulo principal versión 1.3 (sección 6.5), los 4 módulos secundarios versión 1.3 (sección 6.5), la versión 3 de firmware (sección 7.4) y la versión 1.1 del software (sección 8.3). De acuerdo con la metodología descrita en el capítulo 5, todavía quedan dos fases por cubrir. En primer lugar, las pruebas de sistemas. A continuación, las pruebas con alumnos.

En este capítulo se detallarán las pruebas anteriormente mencionadas con el objetivo de validar la plataforma.

9.1. Pruebas de sistema de la plataforma

Las pruebas de sistema, tal y como se describe en la sección 5.3.6, consisten en el uso de la plataforma como parte de un escenario educativo. La Figura 124 muestra la etapa de pruebas de sistema en el contexto de la metodología empleada.

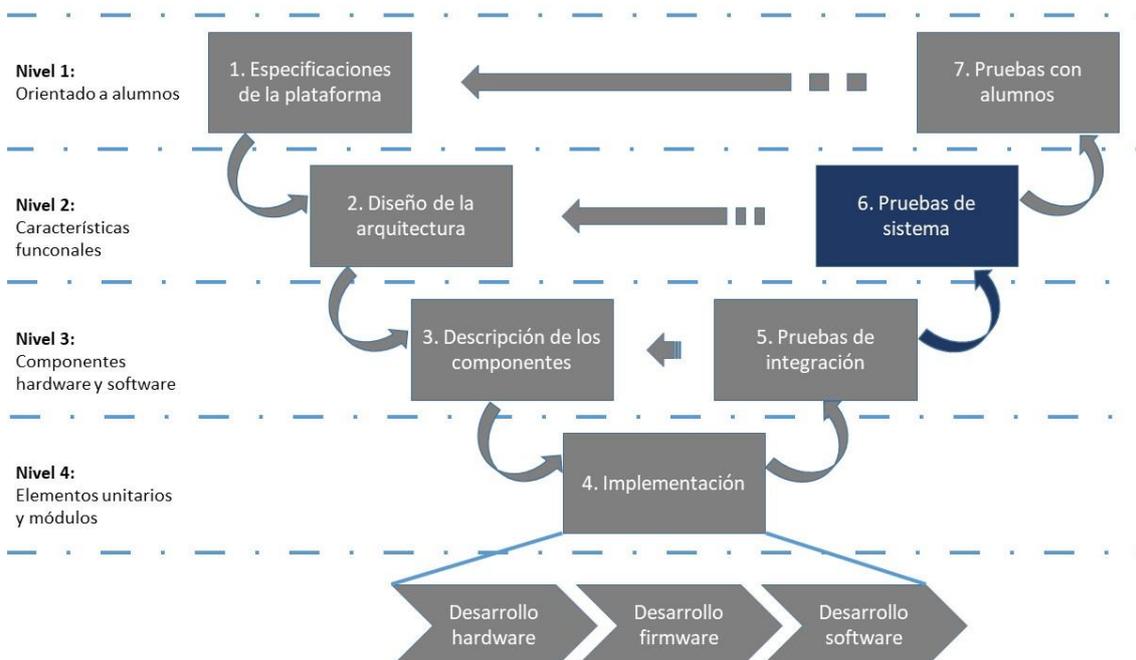


Figura 124. Diagrama de bloques de pruebas de sistema.

En el ámbito del control de plataformas robóticas, esta plataforma es capaz de manejar un robot de cuatro ruedas con brazo manipulador. Este tipo de robots dispone de cuatro

motores DC (Direct Current) que, por facilidad de uso, deben ser manejados por Shields de Arduino de control de motores. Además, incluye un brazo manipulador constituido por seis servomotores. Estos servos motores pueden ser controlados directamente por los módulos secundarios. También se han añadido dos sensores de proximidad infrarrojos para la detección de obstáculos.

Dos alumnos se pueden encargar del control de los motores, otro alumno de la gestión de los sensores para detectar objetos u obstáculos, un cuarto alumno se puede encargar del manejo del brazo manipulador. Deben hacer un trabajo colaborativo para la coordinación de las acciones que lleven a cabo de acuerdo con los objetivos fijados en la sesión que realicen. La Figura 125 muestra el robot de 4 ruedas con brazo manipulador descrito.



Figura 125. Robot de 4 ruedas y brazo manipulador para fines educativos.

9.1.1. DRON EDUCATIVO

Otro escenario educativo se podría enfocar en el control de un Dron de cuatro motores. Cada motor puede ser gestionado por cada alumno, compartiendo la información recibida por un sensor de inclinación de tres ejes y de sensores de proximidad infrarrojos con el fin de realizar una navegación a lo largo de un circuito aéreo. La Figura 126 muestra el Dron Educativo descrito.

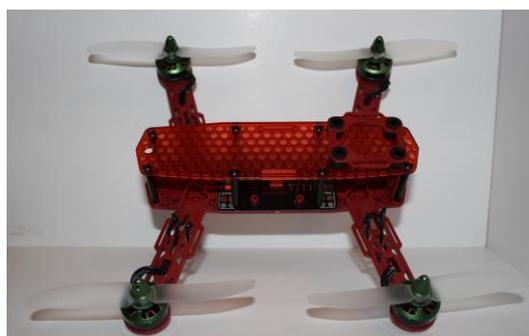


Figura 126. Dron Educativo.

9.1.2. PROTOTIPADO CREATIVO

Con este escenario se pretende potenciar la creatividad y la colaboración en un entorno de prototipado electrónico. La base de este escenario educativo son Shields de Arduino con protoboard que permiten conectar componentes electrónicos de una manera sencilla.

9.1.3. ROTACIÓN DE LUZ CON LED

Se trata de una particularización del escenario Prototipado creativo. Los alumnos deben añadir un LED y una resistencia en el Shields de Arduino con protoboard. El valor de la resistencia debe ser el adecuado para el brillo que deseen que emita el LED cuando lo enciendan.

9.1.4. RED DE COMUNICACIONES LIFI

Se propone una arquitectura de escenario que permite usar la tecnología Li-Fi (Light Fidelity) en el aula. Cada módulo secundario será capaz de transmitir y recibir información a través de la luz sin conexión física.

9.1.5. RED DE SEMÁFOROS

Con este escenario educativo, los alumnos deberán construir una red de semáforos para la gestión del tráfico. Cada módulo secundario se encargará de controlar un conjunto de semáforos. Entre los cuatro módulos secundarios deberán coordinar las señales semafóricas para gestionar el tráfico de una zona de ciudad completa.

9.1.6. CONTROL DE PARKING

En este escenario educativo, los alumnos gestionarán el funcionamiento de un parking. En primer lugar, controlarán la barrera de acceso y la señal luminosa. También deben detectar las plazas de aparcamiento que hay libres y ocupadas para indicarlo con señales luminosas e indicar a los vehículos que entren cuántas plazas hay libres y cuál es su localización. Además, gestionarán el control de salida de los vehículos que hayan validado el tique antes de salir. Por último, no permitirán el acceso a ningún vehículo si el aforo del parking está completo.

9.1.7. ENTORNO EDUCATIVO CON SMART DEVICES

También se propone la elaboración de actividades que incluyan un control remoto de la aplicación escogida mediante Smart Devices: un Smartphone, una Tablet, una Raspberry Pi o un PC.

La conexión entre la Herramienta Colaborativa Robótica Educativa basada en Lógica Programable y Arduino se realiza a través del módulo Bluetooth integrado en el módulo principal. De esta manera, no existe conexión física entre la herramienta educativa y el Smart Device. Los alumnos pueden realizar actividades de programación de las Arduino Nano de manera que, posteriormente, sean capaces de interactuar con los programas educativos elaborados y sus propios Smartphones, u otros dispositivos que se podrían facilitar para tales fines.

9.1.8. ENTORNO REMOTO CON INTERNET

Por último, se propone un control remoto a través de Internet. De esta manera se podrían desarrollar actividades de control de robots sin la necesidad que el alumno esté físicamente en el mismo lugar en el que se encuentra la plataforma.

El enlace entre el usuario de Internet y la Herramienta Colaborativa Robótica Educativa basada en Lógica Programable y Arduino se puede implementar con un servidor remoto de bajo coste, como podría ser Raspberry Pi, un PC, o incluso, añadiendo al módulo principal el Shield de Arduino Yun para darle a la plataforma conexión WiFi.

Con esta última opción, sería posible integrar la Herramienta Colaborativa Robótica Educativa basada en Lógica Programable y Arduino como parte de un Laboratorio Remoto, o se podría implementar un Laboratorio Remoto basado en la propia plataforma.

9.1.9. ESCENARIO PARA LAS PRUEBAS DE SISTEMA

El escenario educativo escogido es el siguiente:

3. Se basa en el prototipado electrónico
4. Los componentes externos a la plataforma son LEDs y resistencias que los propios alumnos pueden incluir a medida que se enfrenten a retos durante la experiencia educativa
5. Se usan protoboard para que los alumnos puedan realizar diferentes montajes

La Figura 127 muestra el montaje descrito anteriormente.

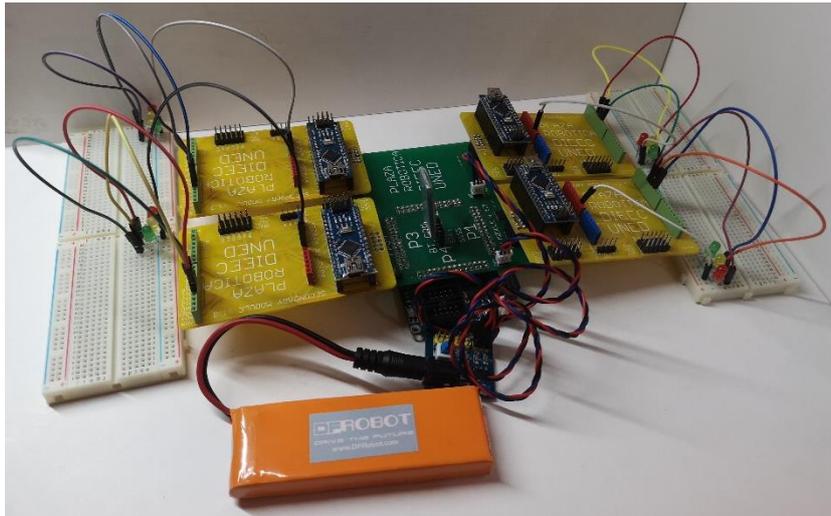


Figura 127. Montaje de la plataforma para pruebas de sistema.

Con el sistema descrito se pueden proponer los siguientes retos:

- Hacer parpadear un LED
- Montar una señal semafórica en cada módulo secundario
- Sincronizar las señales semafóricas conectadas en cada módulo secundario de manera que se comporte como un cruce
- Montar una señal semafórica para vehículos y para peatones
- Sincronizar dichas señales semafóricas como se proponía anteriormente

9.2. Pruebas con alumnos

Las pruebas con alumnos, tal y como se describe en la sección 5.3.7, consisten en el uso de la plataforma con alumnos a lo largo de una experiencia educativa. La Figura 128 muestra la etapa de pruebas con alumnos en el contexto de la metodología empleada.

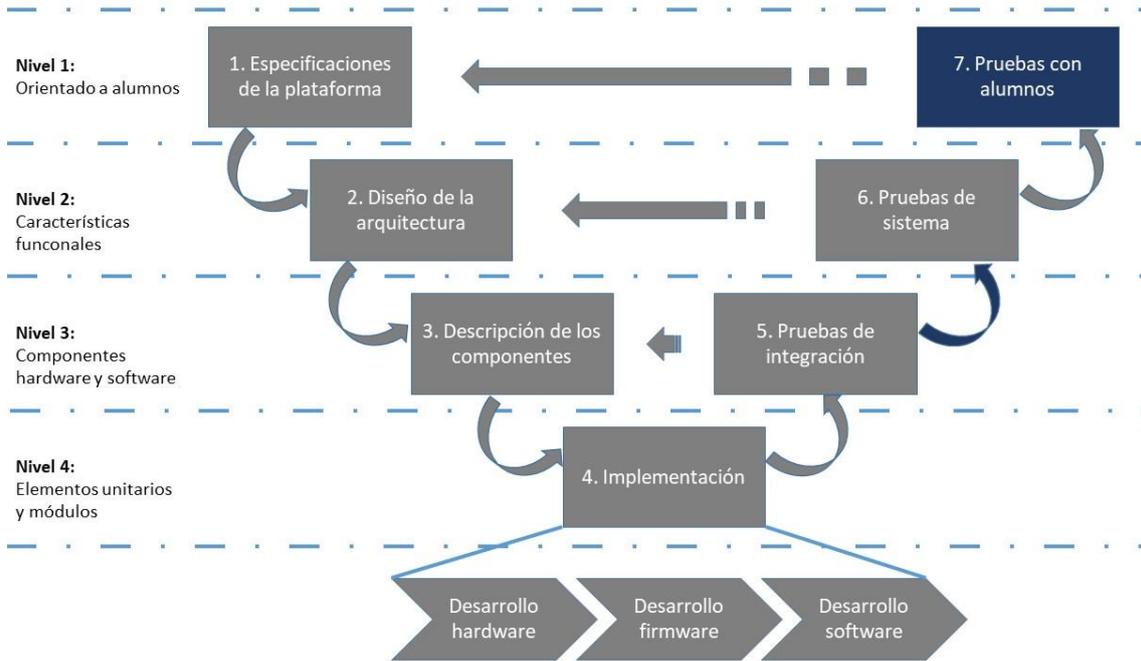


Figura 128. Diagrama de bloques de pruebas con alumnos.

El formato de las pruebas con alumnos es de taller. El objetivo es dar a conocer el uso de plataforma en un contexto de introducción a la robótica.

El taller se desarrolló en la Academia PIXEL [PIXEL, 2018].

El grupo de estudiantes estaba formado por ocho estudiantes de entre 7 y 13 años. Un estudiante era menor de 8 años y el resto tenía entre 9 y 11 años. Todos los estudiantes son niños. La Tabla 142 resume las calificaciones de los estudiantes. Además, la Tabla 142 incluye la correspondencia entre el nivel educativo español y el nivel ISCED.

Tabla 142. Nivel académico de los estudiantes.

Grado educativo español	Nivel ISCED	Número de estudiantes
Educación Primaria	1	6
Educación Secundaria Obligatoria	2 y 3	2

Antes de comenzar el módulo, se realizó una prueba previa para obtener las inquietudes acerca de su asistencia al taller. También se les preguntó acerca de sus conocimientos previos sobre programación, robótica y el uso de la herramienta Arduino.

La Tabla 143 recoge las respuestas a las otras preguntas relacionadas con los conocimientos previos sobre programación, robótica y uso de la herramienta Arduino. Ningún estudiante había tenido experiencias previas con Arduino, ni tenían conocimientos sobre esta herramienta. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 143. Respuestas del pretest.

Pregunta	Nivel de los asistentes				
	1	2	3	4	5
Experiencia en programación	0	0	1	5	2
Conocimientos de programación	0	0	2	4	2
Conocimientos de robótica	0	0	1	5	2
Experiencia con Arduino	8	0	0	0	0
Conocimiento de Arduino	8	0	0	0	0

El taller está dividido en 2 partes. Primero, los estudiantes tienen su primer contacto con la plataforma. Están aprendiendo sobre la parte de hardware. En este punto, aprenden las posibilidades que ofrece una placa Arduino Nano integrada en la plataforma. En un siguiente paso, los estudiantes comienzan a realizar tareas sencillas con un LED conectado al puerto trece (13) del módulo secundario. Durante un segundo, el LED se enciende y luego durante un segundo se apaga. En este paso los alumnos realizan su primer montaje con componentes electrónicos. Utilizan un LED, una resistencia y cables para conectar estos componentes entre sí y con el módulo secundario. Los estudiantes entonces hacen cambios en los tiempos de encendido y de apagado. Finalmente, se les plantea el reto de identificar el tiempo mínimo en que el ojo humano puede percibir un flash LED.

En la segunda parte del taller se les plantean retos con sinergias de aplicaciones reales. En esta parte se les plantea a los alumnos que realicen un semáforo para el control del tráfico de vehículos. Para ello, deben montar un LED rojo, un LED amarillo y un LED verde. Los alumnos deben programar una secuencia para que su montaje se comporte como un semáforo. Es decir, cada puesto de la plataforma debe mostrar un aspecto rojo durante un tiempo. A continuación, dar un aspecto verde durante un tiempo. Y finalmente, dar un aspecto amarillo intermitente. El siguiente reto es trabajar en el comportamiento del semáforo para los peatones. Finalmente, los estudiantes sincronizan cada uno de los módulos secundarios de la plataforma para que cada señal semafórica forme parte del control de vehículos y peatones de un cruce.

La Figura 129 muestra un ejemplo de funcionamiento de la plataforma con la solución de montaje semafórico.

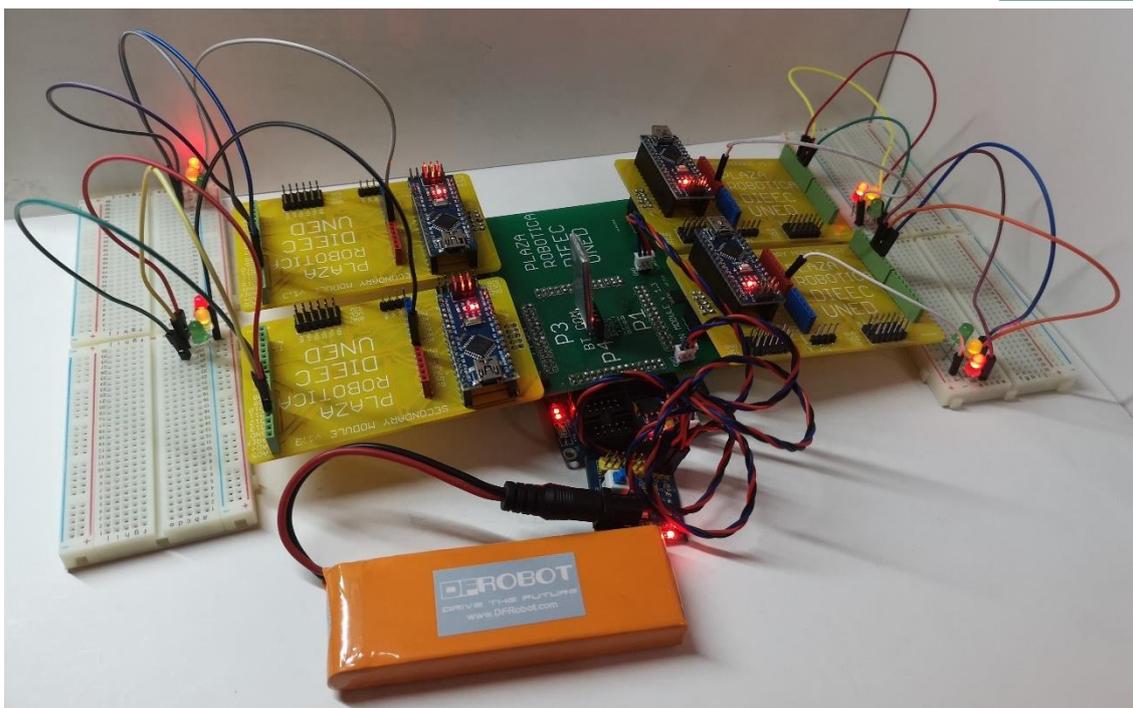


Figura 129. Montaje de la plataforma con solución semafórica.

En la primera parte, todos los estudiantes tuvieron su primer contacto la plataforma. Por otro lado, un importante número de estudiantes ya habían usado herramientas como Scratch, Crumble y Arduino. La Tabla 144 resume la información relativa a las actividades realizadas por los alumnos.

Tabla 144. Resultados de las actividades de las sesiones de la primera parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Hardware de la plataforma	0 (0 %)	8 (100 %)	0 (0 %)
Montaje básico	0 (0 %)	0 (0 %)	8 (100 %)
Montaje intermedio	2 (25 %)	4 (50 %)	2 (25 %)
Percepción visual	2 (25 %)	3 (37,5 %)	3 (37,5 %)

La Tabla 145 se resume la información relativa a las actividades realizadas por los alumnos a lo largo de la segunda parte.

Tabla 145. Resultados de las actividades de las sesiones de la segunda parte.

Título de la sesión	Completado sin ayuda	Completado con algo de ayuda	Completado con ayuda
Semáforo vehículos	4 (50 %)	4 (50 %)	0 (0 %)
Semáforo peatones	4 (50 %)	3 (37,5 %)	1 (12,5 %)
Semáforo completo	8 (100 %)	0 (0 %)	0 (0 %)
Cruce con semáforos	8 (100 %)	0 (0 %)	0 (0 %)

Además, al final del módulo, se llevó a cabo un cuestionario para obtener la opinión del estudiante sobre los resultados del módulo. La Tabla 146 recoge la opinión de los estudiantes sobre el módulo.

Tabla 146. Opinión de los asistentes acerca del curso.

Tema del taller	Número de asistentes			
	Lo mejor	Lo peor	Lo más fácil	Lo más difícil
Programar	1	0	2	2
Encender las luces	6	0	5	0
Trabajar con semáforos	1	1	1	2
Sincronización de semáforos	0	2	0	4
Nada	0	5	0	0

Por otro lado, la Tabla 147 recoge las respuestas a las otras preguntas relacionadas con los conocimientos adquiridos sobre programación, robótica y uso de la plataforma. La mayoría de los estudiantes aumentaron su percepción sobre programación, robótica y conocimientos de la plataforma. Además, la mayoría de los estudiantes mostraron un aumento en su curiosidad por la robótica. Un 1 representa un nivel muy bajo y un 5 representa un nivel muy alto.

Tabla 147. Respuestas del post-test.

Pregunta	Nivel de los asistentes (1 – Nada; 5 – Mucho)				
	1	2	3	4	5
Conocimientos de programación	0	0	1	4	3
Conocimientos de robótica	0	0	1	4	3
Interés por la robótica	0	1	2	2	3
Conocimiento de la plataforma	0	0	0	5	3

Al final del módulo pudimos ver cómo todos los estudiantes aumentaron su motivación para programar, crear proyectos sencillos y trabajar con elementos físicos de robótica educativa.

CAPÍTULO 10

10. COSTES DE LOS DESARROLLOS

Toda investigación implica una inversión económica, puesto que exige contar con una cantidad de materiales y recursos que se dedicarán, en la medida que se requieran, para alcanzar los objetivos plasmados en el plan de investigación.

La elaboración del análisis de costes incluye dos tipos:

- Los costes directos: que son fáciles de cuantificar.
- Los costes indirectos: son aquellos que se requieren para procesos generales de administración, suministro de electricidad, etc. Habitualmente resulta complicado y tedioso calcularlo con exactitud. Por ello, este tipo de costes no se han tenido en cuenta, tomando como hipótesis, que su valor no es significativo en el marco de aplicación de este análisis.

Para este análisis se atiende a la siguiente clasificación de costes directos para la realización de los cálculos:

- Costes de personal: se refieren a los salarios básicos para cada tipo de actividad.
- Materiales consumibles: son todos los productos requeridos para llevar a cabo la investigación. Se enumeran los equipos necesarios para la ejecución de la investigación con sus correspondientes precios y cantidades empleadas.
- Equipamiento: se enumeran los equipos necesarios para la ejecución de la investigación con sus correspondientes precios, cantidades empleadas y tiempo de uso. Para este tipo de costes se toma como hipótesis un tiempo de amortización de los equipos de diez años.
- Viajes: Si como parte de la investigación se han realizado traslados.
- Otros costes: En esta categoría se incluirán aquellos materiales no relacionados directamente con la ejecución de la investigación, pero necesarios para actividades colaterales, como impresiones, combustible, etc.

Uno de los objetivos principales de la herramienta de robótica educativa es lograr una plataforma eficiente en coste. Desde el punto de vista económico, no debería suponer una gran inversión, ya que, si se quiere usar en aulas de educación formal, estas aulas

suelen estar formadas por 30 alumnos, la plataforma es capaz de servir como laboratorio para 4 alumnos, u 8 si se ponen por parejas. Esto implica que dotar a un aula con la herramienta de robótica educativa supone usar entre 4 y 8 unidades.

También se integran a lo largo de este capítulo los costes relaciones con los diferentes desarrollos realizados durante las diferentes fases de hardware, firmware y software.

Seguidamente se detallan los costes relacionados con todas las actividades formativas que se han realizado en el contexto de esta tesis doctoral.

Por último, se agrupan todos los costes en los que se han incurrido con la finalidad de evaluar el número de unidades que sería necesario comercializar, además del precio de dichas unidades, para lograr financiar por completo esta investigación.

10.1. Costes del desarrollo de la arquitectura

El desarrollo de la arquitectura de la plataforma está compuesto por el estudio del estado del arte, la identificación de los objetivos a cumplir, la elección de las funcionalidades que la plataforma debe tener, la definición de la arquitectura que iba constituir la plataforma, la elección de la metodología a usar para los diferentes desarrollos y la planificación de las diferentes etapas que constituyen el desarrollo de la plataforma de robótica educativa. Por todo esto, los costes principales relacionados con el desarrollo de la arquitectura son los propios de un trabajo de investigación y de arquitectura de sistemas. Esto implica que el trabajo ha sido elaborado por dos tipos de perfiles:

- Perfil de investigador: el salario medio de un investigador en España es de 35.000 € anuales. Suponiendo un calendario laboral de 1.800 horas de trabajo al año, el salario de un investigador es de 19,44 € / hora [JobTonic, 2018].
- Perfil de arquitecto de sistemas. el salario medio de un investigador en España es de 46.000 € anuales. Suponiendo un calendario laboral de 1.800 horas de trabajo al año, el salario de un investigador es de 25,55 € / hora [JobTonic, 2018].

Durante la fase del estudio del estado del arte (perfil de investigador) se realizaron las siguientes tareas:

- Análisis de la literatura: durante el análisis de la literatura más de 500 fuentes de información entre publicaciones en conferencia, publicaciones en revistas científicas, proyectos fin de carrera, trabajos fin de grado, trabajos fin de máster, tesis doctorales y páginas web. Esto supuso un esfuerzo de aproximadamente 650 horas.

- Identificación de los objetivos a cumplir: una vez analizada la literatura, se identificaron los objetivos que se querían cumplir a lo largo del desarrollo de la investigación. Esto supuso un esfuerzo de aproximadamente 80 horas.

Una vez fijados los objetivos, se definieron las funcionalidades de la plataforma (perfil de investigador). Esto supuso un esfuerzo de aproximadamente 40 horas. Con una idea clara de las funcionalidades a cumplir, se definió la arquitectura del sistema (arquitecto de sistemas). Esto supuso un esfuerzo de aproximadamente 80 horas.

Posteriormente se dio paso a la elección del tipo de metodología que mejor encajase con el desarrollo de la plataforma (arquitecto de sistemas). Esto supuso un esfuerzo de aproximadamente 160 horas. Por último, se detallaron las diferentes etapas que componen la planificación del desarrollo de la plataforma de robótica educativa (arquitecto de sistemas).

Como equipamiento se usó un ordenador personal. El coste del ordenador personal fue de 1200 euros. Se amortiza el equipo en 10 años. Cada año se puede trabajar con el ordenador 1800 horas. Por todo ello, el coste a la hora del uso de este equipo es de 0,07 € / hora.

Durante las actividades realizadas se han usado diferentes tipos de material de oficina como por ejemplo bolígrafos, cuadernos, etc. Para simplificar la contabilidad se toma como valor de referencia 10 € para este tipo de coste.

A lo largo del estudio del estado del arte se mantuvieron una serie de reuniones de seguimiento. Se ha contabilizado el coste asociado a estos desplazamientos con un valor de 0,24 € / km. Este coste incluye combustible, mantenimiento y amortización del vehículo usado para estos desplazamientos.

La Tabla 148 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 148. Listado de los costes del desarrollo de la arquitectura.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de investigador	19,44 €	770 horas	14.968,80 €
Costes de personal	Perfil de arquitecto de sistemas	25,55 €	240 horas	6.132,00 €
Materiales consumibles	Impresiones	0,045 €	150 copias	6,75 €
Materiales consumibles	Material de oficina	10 €	1	10,00 €
Equipamiento	Ordenador personal	0,07 €	1010 horas	70,70 €
Otros costes	Traslados	0,24 €	408 km	97,92 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo de la arquitectura igual a 27.283,77 €.

10.2. Costes del desarrollo hardware

El desarrollo hardware de la plataforma está compuesto por el diseño del hardware asociado a la plataforma, la fabricación de los PCB (Printed Circuit Board), el ensamblado de las PBA (Printed Board Assembly) y las pruebas de integridad. Estas etapas se repetirán para cada una de las versiones hardware realizadas cuando sea necesario. Por todo esto, los costes principales relacionados con el desarrollo hardware son los propios de un trabajo de ingeniero de hardware. Esto implica que el trabajo ha sido elaborado por este tipo de perfil: el salario medio de un ingeniero de hardware en España es de 21.000 € anuales. Suponiendo un calendario laboral de 1.800 horas de trabajo al año, el salario de un investigador es de 11,67 € / hora [JobTonic, 2018].

Durante la fase del diseño del hardware asociado a la plataforma se realizaron las siguientes tareas:

- Especificaciones hardware: usando la arquitectura del sistema como referencia, se definieron las especificaciones que debía cumplir el hardware. Esto supuso un esfuerzo de aproximadamente 120 horas.
- Definición de los elementos hardware: para cumplir estas especificaciones, se seleccionaron los elementos hardware que se iban a usar. Esto supuso un esfuerzo de aproximadamente 150 horas.
- Diseño de los esquemáticos: se diseñaron las conexiones entre los diferentes elementos hardware. Esto supuso un esfuerzo de aproximadamente 40 horas.

- Diseño físico del PCB: se fijó la localización de todos los componentes y se trazaron las conexiones entre ellos. Esto supuso un esfuerzo de aproximadamente 20 horas.
- Archivos de fabricación: se generaron los archivos de fabricación de los PCB. Esto supuso un esfuerzo de aproximadamente 10 horas.
- Ensamblado de la PBA: se montaron todos los componentes en el PCB y se soldaron. Esto supuso un esfuerzo de aproximadamente 10 horas.
- Pruebas de integridad: una vez obtenido un prototipo se le realizan una serie de pruebas para comprobar que todo está correcto. Es decir, no hay conexiones no deseadas, tampoco hay ausencia de conexiones eléctricas que deberían estar, la alimentación es correcta, etc. Esto supuso un esfuerzo de aproximadamente 15 horas.

Tras completar todos los pasos mencionados anteriormente, se obtuvo la primera versión del hardware. Supuso mucho más tiempo de desarrollo que las demás. Esto se debe a que en la primera versión se partía de cero. Generar sucesivas versiones requería aplicar cambios controlados, lo cual aceleraba bastante el proceso de rediseño.

Posteriormente se dio paso al desarrollo del segundo prototipo. Realizar las correcciones necesarias para el segundo prototipo supuso un esfuerzo de 12 horas. Además, a esta dedicación hay que sumarle la dedicación para las tareas de diseño físico del PCB, archivos de fabricación, ensamblado y pruebas de integridad.

Un tercer prototipo fue necesario. Realizar las correcciones necesarias para el tercer prototipo supuso un esfuerzo de 8 horas. Asimismo, a esta dedicación hay que sumarle la dedicación para las tareas de diseño físico del PCB, archivos de fabricación, ensamblado y pruebas de integridad.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo hardware estas tareas han supuesto un esfuerzo de 190 horas desempeñadas por un perfil de doctor.

Durante las actividades realizadas se han usado diferentes tipos de material de soldadura como por ejemplo soldador, estaño, multímetro, etc. Para simplificar la contabilidad se toma como valor de referencia 30 € para este tipo de coste.

El coste de fabricación de los PCB de un prototipo supone un coste de 40 €. El coste de los conectores usados durante el ensamblado supone un coste de 15 €. El coste de un kit de desarrollo de FPGA es de 24 €. El coste de una tarjeta Arduino Nano es de 20 €.

La Tabla 149 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 149. Listado de los costes del desarrollo hardware.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de ingeniero de hardware	11,67 €	495 horas	5.776,65 €
Materiales consumibles	PCB	40,00 €	6 unidades	240,00 €
Materiales consumibles	Conectores	15,00 €	1 unidad	15,00 €
Materiales consumibles	Material de soldadura	10,00 €	1 unidad	10,00 €
Equipamiento	Ordenador personal	0,07 €	495 horas	34,65 €
Equipamiento	Kit FPGA	24,00 €	1 unidad	24,00 €
Equipamiento	Arduino Nano	20,00 €	4 unidades	80,00 €
Otros costes	Traslados	0,24 €	204 km	48,96 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del hardware igual a 9.580,86 €.

10.3. Costes del desarrollo firmware

El desarrollo hardware de la plataforma está compuesto por las especificaciones firmware, el diseño del firmware asociado a la plataforma, la descripción del firmware, la simulación, la integración en FPGA (Field Programmable Gate Array) y las pruebas de integración hardware-firmware. Estas etapas se repetirán para cada una de las versiones firmware realizadas cuando sea necesario. Por todo esto, los costes principales relacionados con el desarrollo firmware son los propios de un trabajo de ingeniero de firmware. Esto implica que el trabajo ha sido elaborado por este tipo de perfil: el salario medio de un ingeniero de firmware en España es de 27.801 € anuales. Suponiendo un calendario laboral de 1.800 horas de trabajo al año, el salario de un investigador es de 15,45 € / hora [Indeed, 2018].

Durante la fase del diseño del firmware asociado a la plataforma se realizaron las siguientes tareas:

- Especificaciones firmware: usando la arquitectura del sistema como referencia y los resultados del desarrollo de firmware, se definieron las especificaciones que debía cumplir el hardware. Esto supuso un esfuerzo de aproximadamente 10 horas.
- Diseño firmware: usando la arquitectura del sistema como referencia y los resultados del desarrollo de firmware, se definieron las especificaciones que debía cumplir el hardware. Esto supuso un esfuerzo de aproximadamente 180 horas.
- Descripción del firmware: para cumplir estas especificaciones, se describió el firmware que se iba a usar. Esto supuso un esfuerzo de aproximadamente 140 horas.
- Simulación: para probar el diseño firmware se realizaron bancos de pruebas de cada uno de los bloques y del diseño completo. Estos bancos de pruebas se utilizaron durante las simulaciones. También se desarrollaron scripts de simulación para automatizar el proceso de simulación con la finalidad de simplificar el proceso de verificación para versiones posteriores y asegurar una cobertura máxima de las pruebas. Esto supuso un esfuerzo de aproximadamente 160 horas.
- Integración en FPGA: se implementó el diseño en la FPGA. Esto supuso un esfuerzo de aproximadamente 25 horas.
- Pruebas de integración hardware-firmware: se realizaron pruebas de la FPGA con el diseño y de la FPGA montada en el módulo principal. Esto supuso un esfuerzo de aproximadamente 20 horas.

Tras completar todos los pasos mencionados anteriormente, se obtuvo la primera versión del firmware. Supuso mucho más tiempo de desarrollo que las demás. Esto se debe a que en la primera versión se partía de cero. Generar sucesivas versiones requería aplicar cambios controlados, lo cual aceleraba bastante el proceso de rediseño.

Posteriormente se dio paso al desarrollo de la segunda versión de firmware. Realizar las correcciones necesarias para el segundo prototipo supuso un esfuerzo de 28 horas. La tarea de simulación supuso un esfuerzo mucho menor gracias a la estrategia de automatización adoptada. Se pasó de 160 horas hasta un total de 10 horas. También el esfuerzo necesario para completar la tarea de integración en FPGA se redujo considerablemente, desde 25 horas para la primera versión, hasta la cantidad de 5 horas. Asimismo, a esta dedicación hay que sumarle la dedicación para la tarea de las pruebas de integración hardware-firmware.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo firmware estas tareas han supuesto un esfuerzo de 220 horas desempeñadas por un perfil de doctor.

Cuando se manda fabricar un PCB, el fabricante proporciona 10 copias. Por lo tanto, para calcular el coste de un prototipo hay que tener esto en cuenta. También hay que tener en cuenta que un prototipo requiere un PCB del módulo principal y cuatro PCB del módulo secundario. Haciendo uso de los datos mostrados en la sección anterior, el precio de un prototipo hardware es igual a 137 €.

La Tabla 150 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 150. Listado de los costes del desarrollo firmware.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de ingeniero de firmware	15,45 €	598 horas	9.239,10 €
Equipamiento	Ordenador personal	0,07 €	588 horas	41,16 €
Equipamiento	Prototipo hardware	137,00 €	1 unidad	137,00 €
Otros costes	Traslados	0,24 €	68 km	16,32 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del firmware igual a 13.314,38 €.

10.4. Costes del desarrollo software

El desarrollo software de la plataforma está compuesto por las especificaciones del software asociado a la plataforma, los bloques funcionales, la codificación del software, la verificación del software y las pruebas de integración hardware-firmware-software. Estas etapas se repetirán para cada una de las versiones firmware realizadas cuando sea necesario. Por todo esto, los costes principales relacionados con el desarrollo software son los propios de un trabajo de ingeniero de software. Esto implica que el trabajo ha sido elaborado por este tipo de perfil: el salario medio de un ingeniero de software en España es de 29.783 € anuales. Suponiendo un calendario laboral de 1.800 horas de trabajo al año, el salario de un investigador es de 16,55 € / hora [Indeed, 2018].

Durante la fase del diseño del software asociado a la plataforma se realizaron las siguientes tareas:

- Especificaciones software: usando la arquitectura del sistema como referencia y los resultados del desarrollo de hardware y de firmware, se definieron las especificaciones que debía cumplir el software. Esto supuso un esfuerzo de aproximadamente 40 horas.
- Boques funcionales: para cumplir estas especificaciones, se describieron los bloques funcionales de software que se iban a usar. Esto supuso un esfuerzo de aproximadamente 5 horas.
- Codificación del software: posteriormente se realizó la codificación de dichos bloques funcionales de acuerdo con las especificaciones software. Esto supuso un esfuerzo de aproximadamente 20 horas.
- Verificación del software: Una vez terminada la codificación, se realizaron una batería de pruebas para comprobar que el código cumplía con las especificaciones. Esto supuso un esfuerzo de aproximadamente 10 horas.
- Pruebas de integración hardware-firmware-software: por último, se realizaron pruebas donde se integraba todo el desarrollo. Esto supuso un esfuerzo de aproximadamente 15 horas.

Tras completar todos los pasos mencionados anteriormente, se obtuvo la primera versión del software.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo software estas tareas han supuesto un esfuerzo de 25 horas desempeñadas por un perfil de doctor.

La Tabla 151 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 151. Listado de los costes del desarrollo software.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de ingeniero de software	16,55 €	90 horas	1.489,50 €
Equipamiento	Ordenador personal	0,07 €	90 horas	6,30 €
Equipamiento	Prototipo hardware	137,00 €	1 unidad	137,00 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del software igual a 2.073,80 €.

10.5. Costes de las pruebas con alumnos

A continuación, se detallan los costes asociados a las actividades realizadas durante las pruebas con alumnos. Estas actividades son cursos y talleres que se han impartido.

10.5.1. COSTES DEL CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2016/2017 – MÓDULO 1

En el contexto de un curso para introducir a los niños en la robótica. El módulo de Scratch es el primer paso. Para la realización de este módulo realizó el desarrollo del material educativo, la preparación del aula, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados. Por todo esto, los costes principales relacionados con el módulo de Scratch son los propios de un trabajo de desarrollo de contenidos educativos, de técnico de marketing y de instructor. Esto implica que el trabajo ha sido elaborado por tres tipos de perfiles:

- Perfil de desarrollo de contenidos educativos: de acuerdo con conversaciones con academias y centros de formación no reglada, un diseñador de contenidos educativos cobra en torno a 10 € / hora.
- Perfil de técnico de marketing: la banda salarial de un técnico de marketing en España está comprendida entre los 20.000 € y los 33.500 €. Se ha tomado el valor medio para el cálculo de costes. El valor usado es de 26.750 € anuales. Suponiendo un calendario laboral de 1.800 horas de trabajo al año, el salario de un técnico de marketing es de 14,86 € / hora [Reasonwhy, 2018].
- Perfil de instructor: de acuerdo con conversaciones con academias y centros de formación no reglada, un diseñador de contenidos educativos cobra en torno a 6 € / hora.

El desarrollo del material educativo incluyó tareas tales como crear la estructura del curso, describir las diferentes actividades que se desplegaron durante el curso y preparar material de apoyo para el instructor. Esto supuso un esfuerzo de aproximadamente 20 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Antes de comenzar las clases fue necesario preparar el aula. Se prepararon un total de 5 ordenadores antiguos con el sistema operativo Ubuntu y se instalaron las herramientas de robótica educativa que se iban a usar durante todo el curso. Esto supuso un esfuerzo de aproximadamente 15 horas.

Asimismo, se impartieron un total de 8 sesiones con una duración de 2 horas cada sesión. Como hubo dos grupos de alumnos, esto supuso un esfuerzo de aproximadamente 32 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del curso. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 25 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del módulo de Scratch 2017 estas tareas han supuesto un esfuerzo de 21 horas desempeñadas por un perfil de doctor.

Como equipamiento se usó un ordenador portátil. El coste del ordenador personal fue de 400 euros. Se amortiza el equipo en 10 años. Cada año se puede trabajar con el ordenador 1800 horas. Por todo ello, el coste a la hora del uso de este equipo es de 0,02 € / hora.

Los traslados al centro donde se impartió el curso supusieron un total de 720 kilómetros.

La Tabla 152 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 152. Listado de los costes del módulo de Scratch 2017.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	50 horas	500,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	32 horas	192,00 €
Equipamiento	Ordenador portátil	0,02 €	92 horas	1,84 €
Otros costes	Traslados	0,24 €	720 km	172,80 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del módulo de Scratch 2017 igual a 1.385,68 €.

10.5.2. COSTES DEL CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2016/2017 – MÓDULO 2

En el contexto de un curso para introducir a los niños en la robótica. El módulo de Crumble es el segundo paso. Para la realización de este módulo realizó el desarrollo del material educativo, la preparación de kits robóticos basados en Crumble, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados.

El desarrollo del material educativo supuso un esfuerzo de aproximadamente 20 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Antes de comenzar las clases fue necesario preparar kits robóticos basados en Crumble para que los usasen los alumnos. Se prepararon un total de 8 kits que incluían luces, motores y sensores. Esto supuso un esfuerzo de aproximadamente 32 horas. Los materiales de los kits robóticos supusieron un coste económico de 50 € / kit, lo que hace un total de 400 €.

Asimismo, se impartieron un total de 8 sesiones con una duración de 2 horas cada sesión. Como hubo dos grupos de alumnos, esto supuso un esfuerzo de aproximadamente 32 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del curso. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 25 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del estudio del módulo de Crumble 2017 estas tareas han supuesto un esfuerzo de 21 horas desempeñadas por un perfil de doctor.

Los traslados al centro donde se impartió el curso supusieron un total de 720 kilómetros.

La Tabla 153 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 153. Listado de los costes del módulo de Crumble 2017.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	77 horas	770,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	32 horas	192,00 €
Equipamiento	Ordenador portátil	0,02 €	119 horas	2,38 €
Equipamiento	Kit robótico Crumble	50,00 €	8 unidades	400,00 €
Otros costes	Traslados	0,24 €	720 km	172,80 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del módulo de Crumble 2017 igual a 2.056,22 €.

10.5.3. COSTES DEL CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2016/2017 – MÓDULO 3

En el contexto de un curso para introducir a los niños en la robótica, el módulo de Arduino es el tercer paso. Para la realización de este módulo se realizó el desarrollo del material educativo, la preparación de kits robóticos basados en Arduino, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados.

El desarrollo del material educativo supuso un esfuerzo de aproximadamente 20 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Antes de comenzar las clases fue necesario preparar kits robóticos basados en Arduino para que los usasen los alumnos. Se prepararon un total de 8 kits que incluían luces, motores y sensores. Esto supuso un esfuerzo de aproximadamente 32 horas. Los materiales de los kits robóticos supusieron un coste económico de 70 € / kit, lo que hace un total de 560 €.

Asimismo, se impartieron un total de 16 sesiones con una duración de 2 horas cada sesión. Como hubo dos grupos de alumnos, esto supuso un esfuerzo de aproximadamente 64 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del mismo. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 25 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del estudio del módulo de Arduino 2017 estas tareas han supuesto un esfuerzo de 39 horas desempeñadas por un perfil de doctor.

Los traslados al centro donde se impartió el curso supusieron un total de 1440 kilómetros.

La Tabla 154 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 154. Listado de los costes del módulo de Arduino 2017.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	77 horas	770,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	64 horas	384,00 €
Equipamiento	Ordenador portátil	0,02 €	151 horas	3,02 €
Equipamiento	Kit robótico Arduino	70,00 €	8 unidades	560,00 €
Otros costes	Traslados	0,24 €	720 km	172,80 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del módulo de Arduino 2017 igual a 2.726,38 €.

10.5.4. TALLER SCRATCH DAY 2017

Para la realización de este taller se desarrolló el material educativo, se prepararon encuestas de inicio y de fin, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados.

El desarrollo del material educativo supuso un esfuerzo de aproximadamente 10 horas. La preparación de las encuestas de inicio y de fin supuso un esfuerzo de 5 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Asimismo, se impartieron un total de 4 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del curso. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 10 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del taller Scratch 2017 estas tareas han supuesto un esfuerzo de 7 horas desempeñadas por un perfil de doctor.

La Tabla 155 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 155. Listado de los costes del taller Scratch Day 2017.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	25 horas	250,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	4 horas	24,00 €
Equipamiento	Ordenador portátil	0,02 €	39 horas	0,78 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del taller Scratch 2017 igual a 546,86 €.

10.5.5. TALLER CRUMBLE DAY 2017

Para la realización de este taller se desarrolló el material educativo, se prepararon encuestas de inicio y de fin, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados.

El desarrollo del material educativo supuso un esfuerzo de aproximadamente 10 horas. La preparación de las encuestas de inicio y de fin supuso un esfuerzo de 5 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Asimismo, se impartieron un total de 4 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del curso. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 10 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del taller Crumble Day 2017 estas tareas han supuesto un esfuerzo de 7 horas desempeñadas por un perfil de doctor.

La Tabla 156 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 156. Listado de los costes del taller Crumble Day 2017.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	25 horas	250,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	4 horas	24,00 €
Equipamiento	Ordenador portátil	0,02 €	39 horas	0,78 €
Equipamiento	Kit robótico Crumble	50,00 €	5 unidades	250,00 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del taller Crumble Day 2017 igual a 796,86 €.

10.5.6. TALLER DE ROBÓTICA EDUCATIVA MULTI-PLATAFORMA 2017

Para la realización de este taller se desarrolló del material educativo, se prepararon encuestas de inicio y de fin, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados.

El desarrollo del material educativo supuso un esfuerzo de aproximadamente 10 horas. La preparación de las encuestas de inicio y de fin supuso un esfuerzo de 5 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Asimismo, se impartieron un total de 4 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del curso. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 10 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del taller de robótica educativa multi-plataforma 2017 estas tareas han supuesto un esfuerzo de 7 horas desempeñadas por un perfil de doctor.

La Tabla 157 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 157. Listado de los costes del taller de robótica educativa multi-plataforma 2017.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	25 horas	250,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	4 horas	24,00 €
Equipamiento	Ordenador portátil	0,02 €	39 horas	0,78 €
Equipamiento	Kit robótico Crumble	50,00 €	8 unidades	400,00 €
Equipamiento	Kit robótico Arduino	70,00 €	8 unidades	560,00 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del taller de robótica educativa multi-plataforma 2017 igual a 1506,86 €.

10.5.7. CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2017/2018 – SCRATCH BÁSICO

En el contexto de un curso para introducir a los niños en la robótica. El módulo de Scratch básico es el primer paso. Para la realización de este módulo se realizó el desarrollo del material educativo, la preparación del aula, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados.

El desarrollo del material educativo incluyó tareas tales como crear la estructura del curso, describir las diferentes actividades que se desplegaron durante el mismo, preparar encuestas de inicio y de fin, y preparar material de apoyo para el instructor. Esto supuso un esfuerzo de aproximadamente 25 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Antes de comenzar las clases se actualizaron un total de 5 ordenadores al sistema operativo Ubuntu y se volvieron a instalar las herramientas de robótica educativa que se iban a usar durante el mismo. Esto supuso un esfuerzo de aproximadamente 15 horas.

Asimismo, se impartieron un total de 12 sesiones con una duración de 1 hora cada sesión. Como hubo dos grupos de alumnos, esto supuso un esfuerzo de aproximadamente 24 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del curso. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 25 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del módulo de Scratch básico estas tareas han supuesto un esfuerzo de 19 horas desempeñadas por un perfil de doctor.

Los traslados al centro donde se impartió el curso supusieron un total de 720 kilómetros.

La Tabla 158 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 158. Listado de los costes del módulo de Scratch básico.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	55 horas	550,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	24 horas	144,00 €
Equipamiento	Ordenador portátil	0,02 €	89 horas	1,78 €
Otros costes	Traslados	0,24 €	720 km	172,80 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del módulo de Scratch básico igual a 1.281,78 €.

10.5.8. CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2017/2018 – SCRATCH INTERMEDIO

La estructura de costes del módulo de Scratch intermedio es prácticamente igual a la estructura de costes del módulo Scratch básico. La diferencia es que se redujo el esfuerzo relacionado con la preparación de los ordenadores, esto supone una reducción del esfuerzo en 15 horas.

La Tabla 159 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 159. Listado de los costes del módulo de Scratch intermedio.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	40 horas	400,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	24 horas	144,00 €
Equipamiento	Ordenador portátil	0,02 €	74 horas	1,48 €
Otros costes	Traslados	0,24 €	720 km	172,80 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del módulo de Scratch básico igual a 1.131,48 €.

10.5.9. CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2017/2018 – SCRATCH AVANZADO

La estructura de costes del módulo de Scratch avanzado es exactamente igual a la estructura de costes del módulo Scratch intermedio. Por lo tanto, también se obtiene un coste asociado al desarrollo del módulo de Scratch avanzado igual a 1.131,48 €.

10.5.10. CURSO ANUAL DE INTRODUCCIÓN A LA ROBÓTICA 2017/2018 – CRUMBLE BÁSICO

En el contexto de un curso para introducir a los niños en la robótica, el módulo de Crumble básico es el primer nivel de dificultad con el uso de Crumble. El desarrollo de las habilidades de los alumnos inscritos en este curso es del 33 %. Para la realización de este módulo realizó el desarrollo del material educativo, la preparación del aula, la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados.

El desarrollo del material educativo incluyó tareas tales como crear la estructura del curso, describir las diferentes actividades que se desplegaron durante el mismo, preparar encuestas de inicio y de fin, y preparar material de apoyo para el instructor. Esto supuso un esfuerzo de aproximadamente 25 horas.

Posteriormente se dio paso a la elaboración de la campaña de marketing y publicidad. Es decir, elaborar información del curso para hacerlo atractivo a los alumnos. También se hizo una campaña de comunicación en redes sociales. Esto supuso un esfuerzo de aproximadamente 10 horas.

Asimismo, se impartieron un total de 12 sesiones con una duración de 1 hora cada sesión. Como hubo dos grupos de alumnos, esto supuso un esfuerzo de aproximadamente 24 horas.

Tras la finalización del curso se realizó un análisis de las experiencias y resultados obtenidos a lo largo del curso. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 25 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del módulo de Scratch básico estas tareas han supuesto un esfuerzo de 19 horas desempeñadas por un perfil de doctor.

Los traslados al centro donde se impartió el curso supusieron un total de 720 kilómetros.

La Tabla 160 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 160. Listado de los costes del módulo de Scratch básico.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	40 horas	400,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	10 horas	148,60 €
Costes de personal	Perfil de instructor	6,00 €	24 horas	144,00 €
Equipamiento	Ordenador portátil	0,02 €	74 horas	1,48 €
Equipamiento	Kit robótico Crumble	50,00 €	8 unidades	400,00 €
Otros costes	Traslados	0,24 €	720 km	172,80 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del módulo de Scratch básico igual a 1.531,48 €.

10.5.11. TALLERES DE ROBÓTICA EDUCATIVA MULTI-PLATAFORMA 2018

Para la realización de este taller se reutilizó el material desarrollado para el taller de robótica educativa multi-plataforma 2018. Sin embargo, los esfuerzos de la campaña de marketing y publicidad, la impartición de las clases y el análisis de los resultados sí que fueron necesarios.

La campaña de marketing y publicidad supuso un esfuerzo de aproximadamente 5 horas.

Asimismo, se impartieron 4 horas por taller. Se realizaron 3 talleres. Esto supuso un esfuerzo total de 12 horas.

Tras la finalización de los talleres se realizó un análisis de las experiencias y resultados obtenidos a lo largo del taller. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Y, por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 20 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo de los talleres de robótica educativa multi-plataforma 2018 estas tareas han supuesto un esfuerzo de 5 horas desempeñadas por un perfil de doctor.

La Tabla 161 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 161. Listado de los costes de los talleres de robótica educativa 2018.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	20 horas	200,00 €
Costes de personal	Perfil de técnico de marketing	14,86 €	5 horas	74,30 €
Costes de personal	Perfil de instructor	6,00 €	12 horas	72,00 €
Equipamiento	Ordenador portátil	0,02 €	37 horas	0,74 €
Equipamiento	Kit robótico Crumble	50,00 €	8 unidades	400,00 €
Equipamiento	Kit robótico Arduino	70,00 €	8 unidades	560,00 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo de los talleres de robótica educativa multi-plataforma 2018 igual a 1.395,24 €.

10.5.12. TALLER DE ROBÓTICA EDUCATIVA MULTI-PLATAFORMA EDUCON 2018

Para la realización de este taller se adaptó el material desarrollado para el taller de robótica educativa multi-plataforma 2018. Por lo tanto, los esfuerzos relacionados con el material educativo supusieron tan sólo 10 horas.

Asimismo, se impartieron 2 horas de taller.

Tras la finalización del taller se realizó un análisis de las experiencias y resultados obtenidos a lo largo del mismo. Por un lado, para evaluar la respuesta de los alumnos al contenido educativo presentado. Por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas. Esto supuso un esfuerzo de aproximadamente 10 horas.

Además de las tareas anteriormente descritas, también ha habido tareas relacionadas con la revisión, tutela y supervisión. Para el desarrollo del taller de robótica educativa multi-plataforma EDUCON 2018 estas tareas han supuesto un esfuerzo de 5 horas desempeñadas por un perfil de doctor.

La Tabla 162 resume los diferentes tipos de costes (Costes de personal, materiales consumibles, equipamiento, viajes y otros costes), la identificación del coste, el valor unitario, el número de unidades y el valor total del coste.

Tabla 162. Listado de los costes del taller de robótica educativa EDUCON.

Tipo de coste	Identificación del coste	Valor unitario	Número de unidades	Coste total
Costes de personal	Perfil de desarrollo de contenidos educativos	10,00 €	20 horas	200,00 €
Costes de personal	Perfil de instructor	6,00 €	2 horas	12,00 €
Equipamiento	Ordenador portátil	0,02 €	22 horas	0,44 €
Equipamiento	Kit robótico Crumble	50,00 €	1 unidad	50,00 €
Equipamiento	Kit robótico Arduino	70,00 €	1 unidad	70,00 €

Tras el análisis mostrado, se obtiene un coste asociado al desarrollo del taller de robótica educativa multi-plataforma EDUCON 2018 igual a 420,64 €.

10.6. Costes totales de desarrollo de la herramienta de robótica educativa

Los costes totales de desarrollo de la herramienta de robótica educativa se han dividido en dos grandes grupos. Un grupo incluye todos los costes relacionados con el desarrollo de la herramienta (arquitectura, hardware, firmware y software). El otro grupo incluye todos los costes relacionados con las actividades educativas cuyos resultados son aprovechados e integrados en la propia plataforma de robótica educativa. Por todo ello, los costes de la plataforma es la suma de todos los costes mencionados.

Como se ha mostrado a lo largo de este capítulo, los costes de desarrollo relacionados con la arquitectura, el hardware, el firmware y el software suponen más de la mitad de los costes. Pese a esto, los costes relacionados con el desarrollo de actividades formativas ya sean cursos o talleres, también tienen un valor considerable a la hora de calcular los costes totales. La Tabla 163 lista los costes totales de cada uno de los tipos de costes analizados a lo largo de este capítulo.

Tabla 163. Resumen de los costes totales de desarrollo de la herramienta.

Tipo de coste	Valor
Desarrollo de la arquitectura	27.283,77 €
Desarrollo del hardware	9.580,86 €
Desarrollo del firmware	13.314,38 €
Desarrollo del software	2.073,80 €
Desarrollo del módulo de Scratch 2017	1.385,68 €
Desarrollo del módulo de Scratch 2017	2.056,22 €
Desarrollo del módulo de Arduino 2017	2.726,38 €
Desarrollo del taller Scratch 2017	546,86 €
Desarrollo del taller Crumble Day 2017	796,86 €
Desarrollo del taller de robótica educativa multi-plataforma 2017	1506,86 €
Desarrollo del módulo de Scratch básico	1.281,78 €
Desarrollo del módulo de Scratch básico	1.131,48 €
Desarrollo del módulo de Scratch básico	1.531,48 €
Desarrollo de los talleres de robótica educativa multi-plataforma 2018	1.395,24 €
Desarrollo del taller de robótica educativa multi-plataforma EDUCON 2018	420,64 €
Desarrollo de la plataforma de robótica educativa	1.395,24 €
Desarrollo de cursos anuales	1.395,24 €
Desarrollo de talleres	1.395,24 €
Desarrollo de actividades formativas	1.395,24 €

La Tabla 164 incluye los costes de desarrollo de la plataforma y los costes de desarrollo de las actividades formativas realizadas a lo largo de la investigación como parte de la elaboración de esta tesis doctoral.

Tabla 164. Costes totales de desarrollo de la plataforma y de las actividades formativas.

Tipo de coste	Valor
Desarrollo de la plataforma de robótica educativa	52.252,81 €
Desarrollo de cursos anuales	9.460,44 €
Desarrollo de talleres	4.666,46 €
Desarrollo de actividades formativas	14.126,90 €
Costes totales	66.379,71 €

10.7. Conclusiones

Llegados a este punto, se va a evaluar económicamente la solución proporcionada. Para ello, se va a calcular el precio de venta para poder financiar todos los costes de desarrollo anteriormente descritos. De acuerdo con la “Teoría de los Precios” de Milton Friedman [Friedman, 1990], la fórmula para estimar costes de un producto es la siguiente:

- Costes fijos totales + Cálculo de costes variables totales = La suma de costes fijos y variables.
- La suma de costes fijos y variables / Su producción total estimada = Coste por unidad de producción.

El valor de los costes fijos totales es igual al valor de los costes totales de desarrollo, es decir, la cantidad de 66.379,71 €. Los costes variables son los identificados como prototipo hardware, su valor es de 137,00 €.

Para el Precio de Venta al Público (PVP) se tiene en cuenta el IVA (Impuesto sobre el Valor Añadido) y el margen sobre las ventas. Los tipos de IVA impositivos vigentes están divididos en tres grupos:

- IVA superreducido (4 %): es aplicado a los productos de primera necesidad como el pan y la leche. Otros productos con este tipo de IVA superreducido son los libros y periódicos.
- IVA reducido (10%): en este tipo impositivo entran muchísimos productos, como los alimentos en general (salvo los del IVA superreducido) y el transporte de viajeros.
- IVA general (21%): el porcentaje que se aplica a casi todos los productos y servicios, salvo los que están incluidos en los anteriores apartados.

Por lo tanto, el valor de IVA para los cálculos es el 21 %. La cifra sobre el margen de las ventas se fija en un 30 % que es un valor comúnmente utilizado en la venta de artículos electrónicos tales como PC, aparatos electrónicos de consumo y demás. Este margen se puede modificar por un valor del 20 % o del 10 % para campañas promocionales o para aplicar a ventas por volumen, por ejemplo. Pero estos cálculos están basados en la cifra de margen igual al 30 %.

Con todo lo anterior se define la siguiente fórmula para el cálculo:

$$PVP = IVA \times Margen \times \left(\frac{Costes\ fijos}{Número\ de\ unidades} + Costes\ variables \right) \quad (1)$$

Las variables de la fórmula (1) son el PVP y el número de unidades. La Tabla 18 muestra diferentes valores de PVP y número de unidades para tener una idea del PVP que podría tener la herramienta de robótica educativa. Si se despeja de la fórmula (1) el número de

unidades se obtiene la fórmula siguiente con la que se obtiene el número de unidades que es necesario vender para cubrir los costes de desarrollo en función del PVP:

$$\text{Número de unidades} = \frac{\text{Costes fijos}}{\frac{\text{PVP}}{\text{IVA} \times \text{Margen}} - \text{Costes variables}} \quad (2)$$

De esta manera, se puede saber con qué PVP y número de unidades se podrían cubrir los costes de desarrollo. Para calcular el valor límite de PVP que hace que la fórmula (2) no tenga como resultado un valor de número de unidades negativo se usa la siguiente fórmula:

$$\frac{\text{PVP}}{\text{IVA} \times \text{Margen}} - \text{Costes variables} = 0 \quad (3)$$

El valor de PVP obtenido es igual a 215, 50 €. Con un valor de PVP inferior a 215, 50 € no es posible cubrir los costes de desarrollo. Con valores de PVP superiores a 215, 50 € será posible cubrir los costes de desarrollo. La Tabla 165 proporciona un conjunto de valores de PVP y de número de unidades que cumplen con las fórmulas (1) y (2).

Tabla 165. PVP y número de unidades de la plataforma para cubrir los costes.

Número de unidades	PVP
10000	225,94 €
3026	250,00 €
1000	319,92 €
776	350,00 €
100	1.259,65 €

De la Tabla 165 se extraen las siguientes conclusiones:

- Parece que el precio razonable debería estar en torno a los 319,92 €, ya que un volumen de ventas de 1000 unidades no parece ser un número muy elevado.
- Con un incremento en el PVP hasta el valor de 350 € tampoco se reduce mucho el volumen de ventas considerado como razonable. Sólo se reduce el número de unidades hasta 776.
- Con una reducción en el PVP hasta el valor de 250 € se incrementa considerablemente el volumen de ventas considerado como razonable. El valor prácticamente se triplica.

CAPÍTULO 11

11. LECCIONES APRENDIDAS

La guía PMBOK® indica que dentro de los activos de los procesos de la organización está la base del conocimiento, mencionando que las lecciones aprendidas son una de las herramientas fundamentales dentro de la propia organización [PMBOK®, 2018].

En multitud de ocasiones, dentro del ámbito de la gestión de proyectos, surge el debate sobre los motivos de fracaso más relevantes. Entre todas las posibles causas, se enumeran de manera general, se identifican los factores internos y externos y se asigna un grado de importancia. Pese a esto, en rara ocasión se encuentra una aplicación de manera específica en las organizaciones.

Una de las herramientas más potentes para organizaciones dedicadas a la gestión de proyectos son las lecciones aprendidas. Esto implica elaborar una base de conocimiento sobre las experiencias obtenidas durante el desarrollo de los proyectos acometidos con la finalidad de evitar que se repitan los errores cometidos y garantizar que las buenas prácticas o casos de éxito sirvan de guía durante el desarrollo de los nuevos proyectos. Por ello, la elaboración del documento de lecciones aprendidas se ha identificado como una de las prácticas más importantes. Sin embargo, la elaboración de este documento se suele relegar a un segundo plano y no se realizan ejercicios de lecciones aprendidas, o lo que es peor, en caso de realizarse no se analizan y reaprovechan posteriormente [Walker, 2008].

A modo general, para tener un orden de magnitud, según Pamela Babcock [Babcock, 2004], las empresas dentro del ranking Fortune 500 pierden anualmente del orden de 31.500 millones de dólares por no compartir conocimiento, de cualquier tipo.

Los autores en [Trevino, 2008] proponen una estrategia de gestión de las lecciones aprendidas basada en cinco pasos:

- Capturar lecciones: como primer paso, se debe definir las lecciones (el tipo y el contenido) que necesitan captar y luego identificar las áreas de conocimiento específicas que se relacionan con sus esfuerzos para mejorar las prácticas y los procesos de gestión de proyectos.
- Analizar las lecciones: retener y analizar las lecciones aprendidas puede reducir el riesgo de cometer los mismos errores una y otra vez.

- Almacenamiento de lecciones: se debe elegir un medio de almacenamiento accesible que facilite el acceso a la base de conocimiento de las lecciones aprendidas.
- Difusión de las lecciones: es de gran importancia que se difunda las lecciones aprendidas de manera adecuada para que puedan ser utilizadas.
- Aprovechar eficazmente las lecciones aprendidas: las organizaciones deben promover las lecciones aprendidas para que los gerentes de proyectos se beneficien y aprendan de otros proyectos.

Dada la importancia que tienen las lecciones aprendidas en el ámbito empresarial, y la utilidad que se ha identificado de esta herramienta de gestión, se ha decidido aportar como parte del documento de esta tesis doctoral un capítulo que contiene las lecciones aprendidas durante el trabajo realizado en esta investigación.

A lo largo de todas las actividades desarrolladas durante la elaboración de esta tesis doctoral se han tenido éxitos y fracasos en diferentes ámbitos. El objetivo de este capítulo es compartir todos los resultados, tanto los buenos, como los no tan buenos. Paralelo, se han tenido en cuenta las siguientes premisas recomendadas en la gestión de proyectos para la elaboración del documento de lecciones aprendidas:

- Se ha tomado una actitud positiva: siempre se identifica el valor añadido en conocimiento, no se castiga a la persona que cometió un fallo.
- No se culpa a nadie de los errores: muy ligado con el punto anterior, es decir, no se trata de ser un verdugo, y sí se trata de tener un punto de vista objetivo.
- Se centra en los éxitos, así como en los fracasos: se identifica qué fue exitoso y qué fue un error, pero de forma objetiva.
- Se identifican las estrategias: se indica qué estrategias han contribuido al éxito.

Cada sección de este capítulo se articula siguiendo la siguiente estructura:

- Descripción de las condiciones, causas o factores.
- Descripción de la solución
- Análisis de los resultados.

11.1. Seguimiento del avance de la investigación

Para el seguimiento del avance de esta tesis doctoral, la Escuela de Doctorados de la UNED indica el uso de dos documentos: el plan de investigación y el documento de actividades.

En plan de investigación se definen las siguientes fases de la investigación:

- FASE 1: Estudio del estado del arte.
- FASE 2: Desarrollo de la plataforma.
- FASE 3: Elaboración de los escenarios de aplicación de la plataforma:
- FASE 4: Pruebas en aulas de los escenarios de aplicación.
- FASE 5: Análisis de los datos obtenidos durante las pruebas
- FASE 6: Elaboración de las conclusiones extraídas.

El documento de actividades contiene las siguientes actividades:

- Actividades transversales
 - Iniciación al programa de doctorado. m.i.1. gestión de entornos virtuales en la formación a distancia de investigadores
 - Módulo ii: búsqueda y gestión de bases de información científica. m.ii.1. manejo de búsquedas bibliográficas, niveles básico y avanzado
 - Módulo ii: búsqueda y gestión de bases de información científica. m.ii.2. herramientas de gestión de bases de datos bibliográficos
 - Módulo ii: búsqueda y gestión de bases de información científica. m.ii.3. evaluación cualitativa de fuentes bibliográficas
 - Módulo ii: búsqueda y gestión de bases de información científica. m.ii.4. cómo referenciar bibliografía científica
 - Módulo iii: gestión y análisis de datos científicos detalle y planificación
 - Módulo iv: gestión de los procesos de comunicación, difusión e intercambio de los trabajos de investigación realizados (cursos y talleres)
 - Módulo iv: gestión de los procesos de comunicación, difusión e intercambio de los procesos de los trabajos de investigación realizados. m.iv.2. jornadas de doctorandos
- Actividades específicas
 - Módulo iv. gestión de los procesos de comunicación, difusión e intercambio de los trabajos de investigación realizados.m.iv.3. actualización e intercambios científicos. asistencia a seminarios y congresos
 - Publicaciones

11.1.1. DESCRIPCIÓN DE LAS CONDICIONES, CAUSAS O FACTORES

El contenido de estos documentos es de gran utilidad para realizar un seguimiento del avance de la investigación a alto nivel, pero no permite dar visibilidad a bajo nivel cuando se cuenta con un volumen de actividades como el que se ha tenido a lo largo de

esta investigación. Tampoco facilita la visibilidad cuando se tiene tantas áreas de trabajo como las que se han tenido durante el trabajo presentado.

En una reunión mantenida el 18 de enero de 2016 se identificaron las siguientes acciones a realizar:

- Concentrar esfuerzos en el desarrollo de la plataforma.
- Terminar la versión del TAEE 2016 para su revisión a finales de enero.
- Ir enviando la documentación del doctorado para su revisión.
- Trabajar en el Work in Progress para el FIE 2016.
- Preparar artículo con el diseño actual de la plataforma y sus posibilidades.
- Hablar con el profesor Manuel Blázquez para presentarle la plataforma y estudiar la manera de incluirla en sus asignaturas.
- Hablar con Centros Formativos de Ciencia y Robótica de Alcalá de Henares para proponerles talleres con la Plataforma.
- Preparar artículo con contenido educativo de las pruebas realizadas.

11.1.2. DESCRIPCIÓN DE LA SOLUCIÓN

Con la finalidad de facilitar el proceso de seguimiento, se decidió realizar un reporte mensual. De esta manera, se puede saber:

- Qué decisiones se han tomado en la última reunión de seguimiento y qué acciones hay pendientes desde la última reunión.
- Qué acciones se han completado desde el último reporte.
- Qué acciones hay en progreso.
- Qué acciones hay pendientes desde el último reporte.
- Cuál es el roadmap:
 - Los objetivos generales a año vista.
 - Mapa mental de actividades.
- Estado de las publicaciones propuestas en revistas:
 - Artículos en revisión.
 - Artículos rechazados.
- Estado de las publicaciones propuestas para conferencia.

11.1.3. ANÁLISIS DE LOS RESULTADOS

Gracias a los reportes mensuales se ha podido controlar las acciones que ha sido necesario realizar, conocer las fechas límites y poder replanificar a medida que se han ido produciendo desviaciones en la planificación.

Durante el desarrollo de la investigación se han generado 28 reportes mensuales, los cuales han servido para gestionar:

- Una media de 10,6 acciones pendientes. El periodo que más acciones pendientes hubo fue desde junio de 2016 hasta agosto de 2016 con un total de 17 acciones pendientes. El mes que menos acciones pendientes hubo fue desde abril de 2018 hasta mayo de 2018 con un total de 5 acciones pendientes.
- Una media de 6,9 acciones completadas. El mes que más acciones se completaron fue el mes de abril de 2018 con un total de 16 acciones pendientes. El mes que menos acciones completadas hubo fueron los meses de abril de 2016 y el mes de mayo de 2018 con un total de 2 acciones pendientes.
- Durante el curso 2016/2107 se gestionaron un total de 4 objetivos generales a año vista. El número de objetivos generales gestionados durante el curso 2017/2018 fue igual a 3.
- Una media de 6,1 artículos en revisión. El mes que más artículos en revisión hubo fue marzo de 2018 con un total de 8 artículos en revisión. El periodo que menos artículos en revisión hubo fue desde septiembre de 2017 hasta diciembre de 2017 con un total de 5 artículos en revisión.
- Un total de 10 artículos descartados para ser publicados en revista.
- Una media de 3,1 artículos para conferencia. Los meses que más artículos para conferencia se gestionaron fueron agosto de 2016 y abril de 2018 con un total de 6 artículos para conferencia. Hubo meses que no fue necesario gestionar artículos para conferencia. Sin tener en cuenta estos meses, el mes que hubo que gestionar menos artículos para conferencia fue abril de 2017 con la gestión de un artículo para conferencia.

11.2. Canales de comunicación

Cuando se trata de trabajo colaborativo de cierta complejidad, es importante identificar los canales de comunicación adecuados y cuándo usarlos.

11.1.1. DESCRIPCIÓN DE LAS CONDICIONES, CAUSAS O FACTORES

Debido a la complejidad de la investigación, la cantidad de tareas y su diversidad, y las fechas límite, es importante tener claro qué canal de comunicación usar, cómo usarlo y cuándo hacer uso de él. Todo ello para evitar descuidos, malentendidos y una efectividad de las tareas realizadas.

11.1.2. DESCRIPCIÓN DE LA SOLUCIÓN

Se han usado reuniones presenciales para aclaración de conceptos complejos, evaluación global de avances, definición de la estrategia a seguir y para definir las acciones a corto-medio plazo.

Se ha usado Skype [Skype, 2018] de la misma manera que para las reuniones presenciales, pero algún participante requerido para la reunión no podía asistir.

Se han usado llamadas telefónicas para aclaración de conceptos sencillos o dudas simples. También se ha usado este medio de comunicación cuando el tema a tratar era tenía cierta urgencia.

Se ha usado la aplicación de móvil WhatsApp [WhatsApp, 2018] de la misma manera que las llamadas telefónicas, pero el tema a tratar no tenía tanta urgencia.

El correo electrónico ha sido usado para temas que no son urgentes y no se trataba de asuntos triviales que requerían de explicaciones más detalladas. También se ha escogido este medio de comunicación para grupos de trabajo con un número grande de miembros.

11.1.3. ANÁLISIS DE LOS RESULTADOS

Tras la experiencia obtenida, ha sido muy útil usar la variedad de medios de comunicación anteriormente expuestos. Han proporcionado flexibilidad, agilidad y comodidad a la hora de desempeñar las diferentes tareas asociadas a la investigación.

11.3. Arquitectura del sistema

La arquitectura presentada tiene un cierto grado de complejidad, intervienen fases de desarrollo de hardware, fases de desarrollo de firmware, fases de desarrollo de software, fases de desarrollo de escenarios educativos, fases de pruebas con alumnos y fases de realimentación.

11.3.1. DESCRIPCIÓN DE LAS CONDICIONES, CAUSAS O FACTORES.

Desde el principio de la definición de la arquitectura ya se apreciaba el grado de complejidad del sistema completo y la necesidad de hacer fases de realimentación entre unas fases y sus predecesoras. Por tanto, estaba claro que la definición de la arquitectura, de los elementos y de los procesos debía ser lo más flexible, debía permitir la reutilización y tendría que aportar unas especificaciones lo más atomizadas posible.

11.3.2. DESCRIPCIÓN DE LA SOLUCIÓN

Las especificaciones están definidas de la manera más atomizada posible, se ha definido la arquitectura de la manera más modular posible y se ha tratado de reaprovechar el mayor número de desarrollos posibles.

11.3.3. ANÁLISIS DE LOS RESULTADOS

Las especificaciones no son generalistas ni vacías. Esto permite una trazabilidad sencilla desde el alto nivel hasta los detalles de más bajo nivel.

La arquitectura está basada en dos tipos de módulos: el módulo principal y el módulo secundario. La plataforma completa utiliza cuatro módulos secundarios. Este tipo de modularidad tiene dos ventajas principales: simplifica los desarrollos al poder realizar muchas tareas de manera independiente y reduce los tiempos de desarrollo. Los cuatro módulos secundarios sólo se desarrollan una vez y se aprovechan en las cuatro copias.

Para el caso particular de la FPGA (Field Programmable Gate Array) se ha usado un kit de FPGA que reduce considerablemente el desarrollo hardware. Para el caso de Arduino se ha usado una tarjeta Arduino Nano que implica una reducción similar al de la FPGA.

11.4. Desarrollo firmware

El mantenimiento del firmware desarrollado debe ser fácil de mantener. Por un lado, para poder corregir errores de manera sencilla. También para poder aplicar mejoras que no supongan grandes esfuerzos. Asimismo, esta plataforma está pensada para ser usada tal y como se comparte o distribuye, pero también para ser adaptado por el profesorado de acuerdo con sus necesidades. Por último, una característica que aporta la plataforma de robótica educativa es que los alumnos pueden interactuar con ella a todos los niveles, es decir, los alumnos pueden añadir hardware externo, reprogramar las Arduino e incluso pueden reprogramar la FPGA.

11.4.1. DESCRIPCIÓN DE LAS CONDICIONES, CAUSAS O FACTORES.

El desarrollo del firmware se tiene que poder mantener por diferentes personas y no debe ser necesario que tengan una extensa experiencia en el desarrollo de firmware. También es importante que no requiera grandes esfuerzos para aplicar los cambios.

11.4.2. DESCRIPCIÓN DE LA SOLUCIÓN

El diseño firmware ha sido verificado con una metodología de pruebas automatizadas. Se ejecuta un script de pruebas que utiliza el diseño y los bancos de pruebas. El resultado son unos archivos de texto que incluyen información sobre el bloque verificado y los resultados de las pruebas. De esta manera, con una simple inspección de los archivos de texto es posible saber si el comportamiento del diseño es acorde a las especificaciones de las pruebas. A continuación, se muestra un ejemplo sencillo de reporte:

Simulation started at 18:18:22-08/24/17

Testing output ports reset value...

Output ports reset value has been passed.

Testing the frame message 0...

The frame message 0 test has been passed.

...

Testing the frame message 255...

The frame message 255 test has been passed.

Simulation finished at 18:18:31-08/24/17

El reporte mostrado es el resultado de una prueba de 256 caracteres enviados al bloque de comunicaciones UART, se comprueba que llegue el frame enviado correctamente. Por simplicidad al mostrar el ejemplo, sólo se muestran 8 líneas de reporte, el reporte completo tiene un total de 516 líneas.

11.4.3. ANÁLISIS DE LOS RESULTADOS

Gracias a esta práctica se ha ahorrado una gran cantidad de tiempo durante verificación de diseños modificados durante el desarrollo de firmware. Se estima que, para el primer

cambio de firmware, el ahorro en esfuerzo fue de aproximadamente 50 horas, lo que supone un ahorro en coste de desarrollo de 772,2 €. Además, supone un ahorro de tiempo para las futuras verificaciones de nuevos diseños de firmware que sean resultado de cambios de la versión de firmware de la plataforma.

11.5. Gestión de contenidos educativos

La gestión de los contenidos educativos implica desarrollar la estructura del curso, el contenido de las sesiones, los materiales de apoyo al instructor y los indicadores para la evaluación, tanto del desarrollo del aprendizaje de los alumnos, como la eficacia del contenido educativo.

11.5.1. DESCRIPCIÓN DE LAS CONDICIONES, CAUSAS O FACTORES.

Los contenidos educativos se han usado, por un lado, en cursos anuales y, por otro lado, en talleres corta duración, dado que la temática es muy similar, pero varía el tiempo de aprendizaje. Por ejemplo, el módulo de Scratch del curso anual de introducción a la robótica 2016/2017 contaba con un total de 16 horas para conocer la herramienta e introducir a los alumnos en robótica. Desde el otro punto de vista, el taller Scratch Day 2017 tuvo una duración de 4 horas.

11.5.2. DESCRIPCIÓN DE LA SOLUCIÓN

La solución proporcionada es el diseño de un contenido educativo modular. Se han definido unos bloques educativos que se pueden fácilmente adaptar al tipo de formación. Por ejemplo, un módulo para curso tendrá un número de actividades mayor y se podrá profundizar más en los conceptos que incluye ese módulo. Sin embargo, si ese módulo se usa en un taller, se trabajan menos actividades y se profundizará mucho menos en los conceptos subyacentes incluidos en el módulo.

11.5.3. ANÁLISIS DE LOS RESULTADOS

Gracias a esta metodología de diseño de contenidos educativos modulares, ha resultado muy sencillo usar módulos en cursos y en talleres. Además, el esfuerzo requerido es muy bajo. Asimismo, usando esta metodología se ha podido realizar un mantenimiento del contenido educativo y reutilizar correcciones detectadas en cursos para su uso en talleres y viceversa.

11.6. Análisis de resultados en acciones educativas

Tras la finalización de los cursos y talleres se realiza un análisis de las experiencias y resultados obtenidos. En primer lugar, para evaluar la respuesta de los alumnos al contenido educativo presentado y, por otro lado, para evaluar la calidad del contenido educativo y recoger información para corregir defectos detectados y aprovechar las oportunidades de mejora identificadas.

11.6.1. DESCRIPCIÓN DE LAS CONDICIONES, CAUSAS O FACTORES.

En el curso anual de introducción a la robótica 2016/2017, este análisis simplemente incluía la percepción del instructor, con lo cual, se dejaba mucha información fuera del análisis. Durante el taller Scratch Day 2017 se comenzó a trabajar con encuestas de inicio y de fin, de manera que se obtenía información del perfil y trasfondo de los alumnos y se evaluaba los conocimientos y experiencias adquiridos. Pese a esto, la gestión era totalmente manual y en formato físico.

11.6.2. DESCRIPCIÓN DE LA SOLUCIÓN

La solución empleada es el uso de Google Forms [Forms, 2018]. Esta herramienta proporciona una interfaz amigable para la gestión de encuestas. Se pueden generar batería de preguntas de forma sencilla y rápida. También ofrece un tratamiento de los datos bastante visual. Gracias a esta herramienta de gestión de encuestas se puede prescindir del papel. Los alumnos pueden cumplimentar las encuestas inicio antes de empezar el curso o el taller, y pueden responder las encuestas de fin justo al acabar la actividad educativa o posteriormente. Además, el contenido de las encuestas se puede realimentar con experiencias previas.

11.6.3. ANÁLISIS DE LOS RESULTADOS

Gracias a Google Forms se ha podido gestionar multitud de datos de una manera sencilla y con una dedicación bastante baja. La gestión de las encuestas para el taller Scratch Day 2017 supuso un esfuerzo de 5 horas. Por el contrario, para otros talleres o cursos en los que se usó Google Forms para almacenar los datos de las encuestas, la gestión de los datos supuso un esfuerzo de 1 hora. Esto nos ha permitido dedicar más tiempo y poder dedicar más esfuerzos a otras áreas de las actividades educativas.

11.7. Gestión de marketing y publicidad

Cuando se pretende hacer un curso o un taller es muy importante la imagen. Si nadie te conoce ni sabe que cuándo vas a hacer la actividad formativa, dónde va a ser, qué contenido va a tener, o qué le aporta, difícilmente se inscriba y asista. Por todo ello, es básico que los posibles alumnos te conozcan, sepan cuándo y dónde vas a realizar tus actividades, qué contenido va a desplegarse y qué les va a aportar.

11.7.1. DESCRIPCIÓN DE LAS CONDICIONES, CAUSAS O FACTORES.

El curso anual de introducción a la robótica 2016/2017 fue la primera lección de las cosas que se pueden mejorar a la hora de gestionar el marketing y la publicidad. No se trabajó demasiado la imagen en Internet, sólo se trabajó el boca a boca y la cartelería. El resultado fue que las personas inscritas eran conocidos, amigos y demás.

11.7.2. DESCRIPCIÓN DE LA SOLUCIÓN

A medida que se han ido haciendo casos y talleres se ha ido trabajando la imagen en Internet y la presencia en redes sociales. En la actualidad se cuenta con presencia en Internet en la página [Plaza Robotica, 2018] donde se informa de los cursos y talleres y se comparte diferente tipo de información relacionada con la robótica. También hay presencia en redes sociales:

- Página de Facebook de Plaza Robotica [Plaza Robotica, 2018b].
- Twitter de Plaza Robotica [Plaza Robotica, 2018c].
- Instagram de Plaza Robotica [Plaza Robotica, 2018d].
- Página de Facebook de la Rama de Estudiantes del IEEE de la UNED [Rama IEEE, 2018].

Además, se ha recibido apoyo y difusión de diferentes entidades y grupos:

- Departamento de Ingeniería Eléctrica, Electrónica y Control de la UNED [DIEEC UNED, 2018]
- IEEE Young Professionals Spain [YP Spain, 2018]
- IEEE Women in Engineering Spain [WIE Spain, 2018]
- Incomaz [Incomaz, 2018]
- FlexiEspacio [FlexiEspacio, 2018]
- La Estera [La Estera, 2018]
- El Círculo [El Circulo, 2018]

11.7.3. ANÁLISIS DE LOS RESULTADOS

Gracias al trabajo realizado en torno a la imagen de los cursos y talleres, y a la difusión en Internet y redes sociales se ha podido contar con un elevado número de alumnos para las pruebas que se detallan en este documento.

CAPÍTULO 12

12. CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS

Las conclusiones alcanzadas tras finalizar el trabajo de investigación desarrollado y presentado a lo largo de este documento pueden ser establecidas desde diferentes puntos de vista, partiendo para ello de la hipótesis que se planteó en el primer capítulo y que se recoge aquí de nuevo:

El objetivo principal que se busca con esta tesis doctoral es desarrollar una plataforma de robótica educativa que cumpla con las siguientes premisas:

- *Reconfiguración del sistema para permitir diferentes configuraciones de escenarios en función de las necesidades.*
- *Comunicaciones concurrentes que preservan los retrasos no deseados del sistema.*
- *Escalabilidad para adaptar los costes del sistema o la complejidad del sistema.*
- *Se requiere flexibilidad en una base modular con el objetivo de implementar cualquier ocurrencia sin trabajo pesado.*
- *Protección del sistema para mitigar los riesgos de pérdida, mal uso y mala administración del sistema.*
- *Protección menor en los periféricos, como el control de tensión y corriente y la corrección correspondiente.*
- *Facilidad de uso para el desarrollo de prototipos.*
- *Compatibilidad con PC y Smart Devices como Smartphones o Tablets que permite conexiones con diferentes dispositivos adicionales añadidos como parte del proceso de aprendizaje.*

El objetivo final de la plataforma es ser una herramienta que ayude al formador a despertar la curiosidad de los alumnos mediante experimentos que potencien el trabajo en equipo y comunicación entre los alumnos.

Además, este trabajo de investigación no concluye con el material presentado. Por ello, se ha reservado una sección en este capítulo que recoge los desarrollos y aplicaciones futuras que han surgido tras analizar los resultados obtenidos y expuesto a través de los capítulos que componen este documento.

12.1. Conclusiones y aportaciones de este trabajo

Revisados y evaluados los resultados obtenidos mediante las diferentes pruebas llevadas a cabo, se puede considerar validada la hipótesis enunciada, ya que:

- Se proporciona una solución de hardware y software abierto [Repositorio, 2018].
- Se trata de un sistema reconfigurable ya que permite diferentes configuraciones de escenarios en función de las necesidades.
- Al estar basado en FPGA (Field Programmable Gate Array) y contar con un diseño de firmware que es capaz de gestionar 6 canales de comunicación serie de manera concurrente, y ser capaz de gestionar 24 líneas de señales discretas, se garantiza que los retrasos no deseados del sistema son mínimos.
- Debido a la arquitectura desarrollada, este sistema es totalmente escalable. Con ello, este sistema permite adaptar los costes o la complejidad del sistema con total libertad.
- La carga de trabajo necesaria para realizar adaptaciones es muy reducida debido al diseño modular que se ha mantenido en todos los niveles del desarrollo.
- El sistema proporciona suficiente transparencia como para que sea posible el desarrollo de prototipos basados en la plataforma de robótica educativa.
- Se ha demostrado cómo es posible añadir PC y Smart Devices como parte del proceso de aprendizaje.

Por diversos motivos no ha sido posible dar cobertura a los siguientes objetivos planteados en la hipótesis:

- Protección del sistema para mitigar los riesgos de pérdida, mal uso y mala administración del sistema.
- Protección menor en los periféricos, como el control de tensión y corriente y la corrección correspondiente.

Estos dos objetivos son planteados más adelante como parte de las líneas de trabajo futuras.

En relación con los objetivos más específicos planteados junto con la hipótesis que ha marcado este trabajo se pueden reseñar las siguientes aportaciones:

- Esta plataforma de robótica educativa puede ser usada sin ningún tipo de modificación, simplemente conectando periféricos o componentes electrónicos para

que los alumnos experimenten con componentes hardware externos a la plataforma.

- También es posible trabajar con escenarios educativos en los que los alumnos tengan que conectar dispositivos y programar las placas de Arduino para la consecución de los objetivos marcados en el proceso de aprendizaje.
- Para estudiantes con un nivel formativo más avanzado, es posible que estos alumnos puedan, además del uso de dispositivos electrónicos externos a la plataforma y la programación de las placas Arduino, reprogramar la FPGA para obtener la aplicación robótica planteada en las sesiones formativas.
- Asimismo, un profesor tiene la capacidad de usar la plataforma tal y como se muestra en este trabajo, realizar modificaciones software en las placas Arduino, realizar modificaciones firmware en la FPGA, o incluso realizar modificaciones hardware en el módulo principal y el módulo secundario para adaptar la herramienta a sus necesidades en el aula.
- Otra aportación significativa de este trabajo son los materiales educativos. Estos materiales pueden ser usados tal cual, y también es posible modificarlos para su uso en el aula.

La parte experimental de este trabajo doctoral ha estado basada en el despliegue del conjunto de desarrollos que se han implementado en escenarios reales de aplicación. Se han descrito los escenarios educativos desplegados en diferentes cursos y talleres.

Posteriormente y con el objetivo de validar tanto la hipótesis como los objetivos planteados al comienzo de este trabajo, así como para satisfacer las motivaciones que han dado pie al mismo, se ha evaluado la respuesta del desarrollo con usuarios reales, los cuales han realizado una variedad de actividades formativas.

En cuanto a las aportaciones ofrecidas por el presente trabajo en relación con la posibilidad de obtener una plataforma de robótica educativa, se pueden apuntar las siguientes ideas:

- Iniciar el desarrollo de una plataforma partiendo de cero requiere mucho esfuerzo, tiempo y dinero. A pesar de no haber partido de cero (se han reusado kits de desarrollo como la FPGA y las Arduino Nano), se ha demostrado el orden de magnitud que significa acometer un proyecto de esta envergadura.
- Si se quiere proporcionar una herramienta fácil de integrar en otros sistemas y de modificar para adaptar a las diferentes necesidades, es deseable basar los desarrollos en:

- Un interfaz de comunicaciones estandarizado y sencillo.
- Un diseño modular.
- Incluir fases de verificación lo más automatizadas que se pueda.
- Aportar ejemplos de uso sencillos, de dificultad intermedia y con cierta complejidad. De esta manera, una persona que quiera usarlo o modificarlo, podrá familiarizarse con la herramienta de una manera gradual y no tendrá ni barreras de entrada. También se conocerá las limitaciones que pueda tener la herramienta.

12.2. Publicaciones realizadas durante el trabajo de investigación

Durante el proceso de investigación se ha tratado de recoger información y opiniones de los investigadores y grupos relevantes en el área de trabajo, así como de llevar a cabo acciones de diseminación de nuestras actividades y resultados en diferentes foros y congresos.

Para ello se ha acudido a conferencias para presentar los avances en la implementación del material educativo que se ha diseñado y obtener así impresiones sobre el mismo por parte de la comunidad investigadora. Las conferencias a las que se ha acudido son las siguientes:

- REV 2016: 13th International Conference on Remote Engineering and Virtual Instrumentation [REV 2016, 2018].
- TAAE 2016: XII Technology, Learning and Teaching of Electronics Conference [TAAE 2016, 2018].
- FIE 2016: 46th Annual Frontiers in Education Conference [FIE 2016, 2018].
- REV 2017: 14th International Conference on Remote Engineering and Virtual Instrumentation [REV 2017, 2018].
- EDUCON 2017: 8th IEEE Global Engineering Education Conference [EDUCON 2017, 2018].
- ICL2017 – Teaching and Learning in a Digital World: 20th International Conference on Interactive Collaborative Learning, 46th IGIP International Conference on Engineering Pedagogy [ICL 2017, 2018].
- REV 2018: 15th International Conference on Remote Engineering and Virtual Instrumentation [REV 2018, 2018].

- EDUCON 2018: 9th IEEE Global Engineering Education Conference [EDUCON 2018, 2018].
- TAAE 2018: XIII Technology, Learning and Teaching of Electronics Conference [TAAE 2018, 2018].

Asimismo, se tiene previsto acudir a las siguientes conferencias con las experiencias y resultados obtenidos durante esta investigación:

- LWMOOCS 2018: LWMOOCS V Learning with MOOCS 2018 [LWMOOCS 2018, 2018].
- FIE 2018: 48th Annual Frontiers in Education Conference [FIE 2018, 2018].
- TALE 2018: 7th International Conference on Teaching, Assessment and Learning for Engineering [TALE 2018, 2018].

El conjunto de publicaciones realizadas desde la incorporación del autor de esta tesis doctoral al grupo de investigación del Departamento de Ingeniería Eléctrica, Electrónica y Control de la Universidad de Nacional de Educación a Distancia muestra la evolución de este trabajo desde que se comenzó a trabajar en el diseño y desarrollo de la plataforma de robótica educativa, partiendo de un estudio del estado de arte, presentación de la arquitectura planteada, presentación de los primeros prototipos, además de las experiencias educativas llevadas a cabo con alumnos:

- Carro, G., Carrasco, R., Plaza, P., Canas, M.A., Sancristobal, E., Castro, M. & Mur, F., "Mechatronics and robotics as motivational tools in remote laboratories" [Carro, 2015b]
- Plaza, P., Sancristobal, E., Carro, G. & Castro, M., "A Wireless robotic educational platform approach" [Plaza, 2016a]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Pérez, C., "Collaborative robotic educational tool based on programmable logic and Arduino" [Plaza, 2016b]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Pérez, C., "Herramienta Colaborativa Robótica Educativa basada en Lógica Programable y Arduino" [Plaza, 2016c]
- Plaza, P., Sancristobal, E., Carro, G. & Castro, M., "Robotic Educational Tool to engage students on Engineering" [Plaza, 2016d]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Ruiz, E., "Wireless Development Boards to Connect the World" [Plaza, 2017a]

- Plaza, P., Sancristobal, E., Carro, G. & Castro, M., "Home-made robotic education, a new way to explore" [Plaza, 2017b]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Blázquez, M., "From 4 Wheeled Remote Robot to Serious Collaborative Remote Laboratory" [Plaza, 2017c]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M., Blázquez, M., Muñoz, J. & Álvarez, M., "Scratch as Educational Tool to Introduce Robotics" [Plaza, 2017d]
- A. Peixoto et al., "Diversity and inclusion in engineering education: Looking through the gender question" [Peixoto, 2018a]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M., Blázquez, M. & Peixoto, A., "Traffic lights through multiple robotic educational tools" [Plaza, 2018a]
- A. Peixoto et al., "Robotics tips and tricks for inclusion and integration of students" [Peixoto, 2018b]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Blázquez, M., "Scratch day to introduce robotics" [Plaza, 2018b]
- Plaza P., Sancristobal E., Carro G., Garcia-Loro F., Blázquez M. & Castro M., "European Robotics Week to introduce robotics and promote engineering" [Plaza, 2018c]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blázquez, M., Garcia-Loro F. & Muñoz, J., "Crumble como herramienta educativa para introducir en robótica" [Plaza, 2018d]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blázquez, M., Garcia-Loro F., "Iluminando a través de la robótica educativa" [Plaza, 2018e]

Asimismo, las siguientes publicaciones han sido aceptadas y están pendientes de ser publicadas:

- Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blázquez, M., Garcia-Loro F. & Muñoz, J." [Plaza, 2018f]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blázquez, M., Garcia-Loro F., "Lighting through educational robotics" [Plaza, 2018g]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blázquez, M., Garcia-Loro F., "Crumble Day to introduce robotics" [Plaza, 2018h]
- Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blázquez, M., Garcia-Loro F., "Multiplatform educational robotics course to introduce children in robotics" [Plaza, 2018i]

- Plaza, P., Sancristobal, E., Carro, G., Blazquez, M., Menacho, A., Castro, M., Perez, C. & Garcia-Loro F., "Local MOOC Solution For Thight Budgets Or Limited Internet Access" [Plaza, 2018j]
- Plaza, P., Sancristobal, E., Carro, G., Blazquez, M., Garcia-Loro F., Martin, S., Perez, C., & Castro, M., "Arduino as educational tool to introduce robotics" [Plaza, 2018k]

Además, las siguientes publicaciones están en proceso de revisión:

- Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Ruiz, E., "Internet of Things development boards for Robotic Education" [Plaza, 2018l]
- Plaza, P., Sancristobal, E., Carro, G., Blazquez, M., Garcia-Loro F. & Castro, M., "Combining Programmable Logic and Arduino for Robotic STEM Education" [Plaza, 2018m]

12.3. Desarrollos y aplicaciones futuras

El trabajo presentado en este documento plantea una serie de alternativas a los problemas previamente descritos y relacionados con la educación STEM y la robótica educativa en nuestra sociedad. En este contexto, aún quedan soluciones que aportar y mejoras que realizar respecto a los desarrollos presentados. Llegados a este punto, cabe destacar, entre otras, las siguientes posibles líneas futuras de investigación:

- El trabajo presentado describe el desarrollo de la primera línea de base para la plataforma de robótica educativa. Esta línea de base se basa en prototipos de placas. En un futuro próximo, se prevé trabajar en dos líneas de base más que aprovechen los resultados obtenidos en las anteriores. Después de las pruebas de escenarios educativos, la segunda línea de base integrará los cambios y mejoras adquiridos a partir de la primera línea de base. Para la tercera línea de base, una integración completa usando los dos tipos de placas: módulo principal y módulo secundario. El módulo principal integrará el hardware FPGA y el módulo de comunicaciones Bluetooth. Por otro lado, el módulo secundario integrará el hardware de Arduino como parte de la placa. Por lo tanto, se proponen dos líneas de trabajo que podrían realizarse como dos tesis doctorales. Dado que los siguientes objetivos planteados en la hipótesis no se llegaron a incluir en la primera línea de base, sí que se podrían integrar tanto en la segunda, como en la tercera línea de base:

- Protección del sistema para mitigar los riesgos de pérdida, mal uso y mala administración del sistema.
- Protección menor en los periféricos, como el control de tensión y corriente y la corrección correspondiente.
- Otra de las aportaciones de esta tesis doctoral es el desarrollo de material educativo para la introducción a la robótica. Se propone continuar usando este material educativo con las herramientas Scratch, Crumble y Arduino. Una tesis doctoral podría tener el foco de su investigación en validar el material educativo con un mayor número de alumnos y experiencias y desarrollar una evolución del material educativo.
- Asimismo, y ya que no se ha integrado la plataforma en sistemas de enseñanza remota tales como los MOOCs. Se podría considerar desarrollar, en el contexto de un trabajo fin de grado, un MOOC usando la plataforma de robótica educativa.
- Además, se puede explorar el uso de la herramienta de robótica educativa en escenarios educativos relacionados con Smart Cities, Internet of Energy y Electrónica de Potencia. Cada uno de estos caminos planteados podría ser realizado en el contexto de un trabajo fin de grado o trabajo fin de máster.

12.4. Observaciones finales

Los esfuerzos realizados a lo largo de esta investigación han dado como fruto unos resultados interesantes y nuevas líneas de trabajo. En la actualidad, la educación STEM está en auge, y no parece que el interés de la sociedad y de la comunidad científica vaya a disminuir a corto, largo plazo. Si no que, todo apunta, a que la educación STEM va a estar en el foco, tanto de la sociedad, como del entorno educativo e incluso el mundo empresarial. Por todo ello, se plantea darle continuidad a los resultados obtenidos en este trabajo, tanto en el contexto de futuras tesis doctorales, como de trabajos fin de master, como de trabajos fin de grado y en el contexto de actividades a pequeña escala para promover grupos de trabajo tales como el Capítulo Español de la Sociedad de Educación del IEEE, la Rama de Estudiantes del IEEE de la UNED, el grupo de afinidad de Jóvenes Profesionales del IEEE, el grupo de afinidad de Mujeres en la Ingeniería del IEEE y otros grupos interesados en aprovechar las sinergias de la robótica educativa y la educación STEM.

Fuera del marco de esta tesis doctoral, se están llevando a cabo una serie de acciones que podrían contribuir a los resultados mostrados ya que han surgido del trabajo desarrollado durante esta tesis doctoral, estas acciones son las siguientes:

- Uso de una solución basada en red local y el uso de un LMS (Learning Management System) para entornos con poco presupuesto o dificultades de acceso a Internet [Plaza, 2018j]. Esta línea de trabajo surgió durante el EDUCON 2018.
- Uso de Scratch para dinamizar las clases de matemáticas. Esta línea de trabajo también surgió durante el EDUCON 2018. Se trata de una colaboración entre investigadores de diferentes países: España, Brasil y Canadá.

CAPÍTULO 13

13. CURRÍCULUM VITAE

13.1. Datos personales

Nombre y Apellidos: Pedro Plaza Merino

Fecha de nacimiento: 20 de enero de 1982

Correo electrónico: pplaza@plazarobotica.es

Domicilio: Torrejón de Ardoz España



13.2. Extracto:

Obtuve el título de Máster por la ETSII (Escuela Técnica Superior de Ingeniería Industrial) de la UNED, Madrid, España, en 2013. Asimismo, soy Ingeniero Industrial con doble especialidad (Electrónica y Automática, y Electricidad) por la Universidad Carlos III de Madrid (UC3M), en 2010. Actualmente soy Ingeniero de Proyectos de I+D en Siemens Rail Automation. Tengo más de 10 años de experiencia con FPGAs. En los últimos 8 años he desarrollado tareas relativas al diseño de sistemas electrónicos orientados a los enclavamientos y control de trenes. Además, soy miembro del Consejo Editorial de International Journal of Automation and Robotics Technology en Inderscience. También soy autor de varias publicaciones en prestigiosas revistas y conferencias. He colaborado en varios proyectos de investigación. Soy miembro del IEEE desde el año 2011 (Número de Miembro: 92624558), de la IEEE Robotics and Automation Society, de la IEEE Education Society, de la IEEE Young Professionals y de la IEEE Women in Engineering (WIE). Actualmente ostento el cargo de presidente en la rama estudiantil IEEE de la UNED, así como moderador de WIE en la rama estudiantil IEEE de la UNED. De los profesionales con los que colaboro valoro la sinceridad, el compromiso y la iniciativa.

13.3. Experiencia laboral:

SIEMENS RAIL AUTOMATION S.A.U., Tres Cantos, España, de mayo 2013 hasta la fecha: Ingeniero de hardware de I+D

DIMETRONIC S.A.U., San Fernando de Henares, España, de octubre 2009 hasta abril de 2013: Ingeniero de Proyectos de I+D

Grupo de Diseño Microelectrónico y Aplicaciones, UC3M, Leganés, España, de 2007 a 2010: Becario

13.4. Educación:

Módulo de Iniciativa Emprendedora en la Universidad

Escuela de Organización Industrial (EOI) / UNED, noviembre-diciembre 2013

Modelo de Plan de Negocio: Greenwatch.

Máster Universitario en Ingeniería Industrial en Electricidad, Electrónica y Control Industrial, intensificación en Electricidad y Electrónica

UNED, 2013

Trabajo Fin de Máster: Robótica Basada en FPGA con Fines Educativos. Durante el desarrollo se usó la FPGA Cyclone II de Altera para el control remoto de un robot humanoide de 19 grados de libertad.

Ingeniería Industrial, especialidad Electrónica y Automática, y Electricidad

Universidad Carlos III de Madrid, 2010

Proyecto Fin de Carrera: Sistema de Emulación Autónoma con comunicación USB 2.0 integrado en un entorno con MicroBlaze. Durante el desarrollo se usaron las FPGAs Virtex 4 and Virtex 5 de Xilinx para el control de un emulador de fallos usando USB como protocolo de comunicación entre la FPGA y el PC. También se desarrolló una interfaz gráfica de usuario en Visual Studio para poder introducir los parámetros de configuración, tanto de la FPGA usada, como del propio emulador de fallos.

Ingeniería Técnica Industrial, especialidad Electrónica y Automática Industrial

Universidad Carlos III de Madrid, 2007

Proyecto Fin de Carrera: Diseño e implementación del control de un convertidor CC/CC basado en un microcontrolador PIC. Durante el desarrollo se implementaron los tipos de control P, PD, PI y PID de un regulador de CC-CC con topología forward mediante un microcontrolador PIC de Microchip.

13.5. Habilidades

Personales:

- Comunicación efectiva; Comunicación intercultural; Resolución de problemas/Creatividad; Habilidades interpersonales; Habilidades de trabajo en equipo; Planificación y organización; Liderazgo y gestión.
- Flexibilidad y adaptabilidad; Profesionalidad y ética en el trabajo; Actitud positiva y energía.

Diseño para FPGA usando VHDL:

- Interfaces serie como RS232, I2C y SPI. Interfaces series a medida. CRC. DSP como RMS, THD, Frecuencia and Cálculo de raíz cuadrada basado en métodos iterativos. Comunicación con hardware externo como EEPROM, microprocesadores y diferentes tipos de sensores.

Dispositivos de desarrollo de aplicaciones de robótica educativa:

- Arduino.
- Raspberry Pi.
- Laboratorios remotos.

Uso de equipos especializados:

- Osciloscopio, Analizador Lógico y Fuente Programable de AC/DC.
- Otros equipos de laboratorio usados comúnmente para verificación y validación de diseños eléctricos y electrónicos.

Informáticas:

- Lenguajes de programación: C, C++, HTML, PHP, Python, Matlab script y TCL.
- Lenguajes de descripción de hardware: VHDL (CPLDs and FPGAs).
- Herramientas software y aplicaciones: Matlab, Modelsim, Active-HDL, Altera Quartus II, Xilinx ISE, Synposys Synplify, EAGLE, DOORS, Synergy y Change.
- Sistemas operativos: DOS / UNIX / LINUX, Raspbian y WINDOWS.

Idiomas:

- Español: lengua materna.
- Inglés: competencia profesional completa. Nivel B2 certificado por la UNED.

HOBBIES:

- Pasar tiempo con la familia y con amigos.
- Nuevas Tecnologías.
- Pádel.

CAPÍTULO 14

14. BIBLIOGRAFÍA

- [Accenture, 2018] Accenture News Room [Online]. Disponible: <https://newsroom.accenture.com/>, visitada el 6 de julio de 2018
- [Adams, 2011] Adams, R., Evangelou, D., English, L., De Figueiredo, A. D., Mousoulides, N., Pawley, A. L., Schiefellite, C., Stevens, R., Svinicki, M., Trenor, J. M. & Wilson, D. M. (2011), Multiple Perspectives on Engaging Future Engineers. *Journal of Engineering Education*, 100: 48–88. doi: 10.1002/j.2168-9830.2011.tb00004.x
- [Agenda, 2018] Agenda Digital para Europa. [Online]. Disponible en: https://europa.eu/european-union/file/1501/download_es?token=3l7D0Fil, visitada el 6 de julio de 2018
- [Alam, 2012] Alam, M. R., Reaz, M. B. I. & Ali, M. A. M., "A Review of Smart Homes—Past, Present, and Future," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190-1203, Nov. 2012. doi: 10.1109/TSMCC.2012.2189204
- [Alexander, 2015] Alexander, P. J. & Radhakrishnan, N., "Remote lab implementation on an embedded web server," in *Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on*, vol., no., pp.1-5, 19-20 March 2015. doi: 10.1109/ICCPCT.2015.7159525
- [Al-Kaseem, 2017] Al-Kaseem, B. R. & Al-Rawashidyhamed, H. S., "SD-NFV as an Energy Efficient Approach for M2M Networks Using Cloud-Based 6LoWPAN Testbed," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1787-1797, Oct. 2017. doi: 10.1109/JIOT.2017.2704921
- [Almerich, 2016] Almerich, G., Orellana, N., Suárez-Rodríguez, J. & Díaz-García, I., "Teachers' information and communication technology competences: A structural approach". *Computers & Education*, Volume 100, September 2016, Pages 110-125. doi: 10.1016/j.compedu.2016.05.002
- [Altera, 2018] Página web de la familia Cyclone II de Altera. [Online]. Disponible en: <https://www.altera.com/products/fpga/cyclone-series/cyclone-ii/overview.html>, visitada el 6 de julio de 2018
- [Ang, 2017] Ang, L. M., Seng, K. P., Zungeru, A. M. & Ijamaru, G. K., "Big Sensor Data Systems for Smart Cities," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1259-1271, Oct. 2017. doi: 10.1109/JIOT.2017.2695535
- [Anzalone, 2013] Anzalone, G. C., Glover, A. G. & Pearce, J. M., "Open-Source Colorimeter". *Sensors* 2013, 13, 5338-5346. doi: 10.3390/s130405338

- [Araújo, 2013] Araújo, A., Portugal, D., Couceiro, M. S. & Rocha, R. P., "Integrating Arduino-based educational mobile robots in ROS," 2013 13th International Conference on Autonomous Robot Systems, Lisbon, 2013, pp. 1-6. doi: 10.1109/Robotica.2013.6623520
- [Arduino, 2018a] Arduino [Online]. Disponible en: <https://www.arduino.cc/>, visitada el 6 de julio de 2018
- [Arduino, 2018b] Arduino WiFi Shield [Online]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoWiFiShield>, visitada el 6 de julio de 2018
- [Arduino, 2018c] Arduino Yún Shield [Online]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoYunShield>, visitada el 6 de julio de 2018
- [Arduino, 2018d] Arduino Yún Linino OS [Online]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoBoardYun>, visitada el 6 de julio de 2018
- [Arduino, 2018e] Arduino MKR1000 [Online]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoMKR1000>, visitada el 6 de julio de 2018
- [Arduino, 2018f] Arduino forum [Online]. Disponible en: <https://forum.arduino.cc/>, visitada el 6 de julio de 2018
- [Arduino, 2018g] Arduino WiFi Shield shop [Online]. Disponible en: <https://store.arduino.cc/product/A000058>, visitada el 6 de julio de 2018
- [Arduino, 2018h] Arduino Yún Linino OS shop [Online]. Disponible en: <https://store.arduino.cc/product/A000008>, visitada el 6 de julio de 2018
- [Arduino, 2018i] Página web de la tarjeta Arduino Nano. [Online]. Disponible en: <http://www.arduino.cc/en/Main/ArduinoBoardNano>, visitada el 6 de julio de 2018
- [Arduino, 2018j] Arduino web editor. [Online]. Disponible en: <https://create.arduino.cc/editor>, visitada el 6 de julio de 2018
- [ARM, 2018] ARM. [Online]. Disponible en: <https://www.arm.com/>, visitada el 6 de julio de 2018
- [Aroca, 2013] Aroca, R. V. et al., "Increasing Students' Interest With Low-Cost CellBots," in IEEE Transactions on Education, vol. 56, no. 1, pp. 3-8, Feb. 2013. doi: 10.1109/TE.2012.2214782. doi: 10.1109/TE.2012.2214782
- [Ashton, 1948] Ashton, T.S. (1948), "The Industrial Revolution (1760–1830)". Oxford University Press.
- [Assante, 2016] Assante, D., Fornario, C., Sayed, A. E. & Salem, S. A., "Edutronics: Gamification for introducing kids to electronics," 2016 IEEE Global Engineering Education Conference (EDUCON), Abu Dhabi, 2016, pp. 905-908. doi: 10.1109/EDUCON.2016.7474659
- [August, 2015] August, S. E., Hammers, M. L., Murphy, D. B., Neyer, A., Gueye, P. & Thames, R. Q., "Virtual Engineering Sciences Learning Lab: Giving STEM Education a Second Life," in IEEE Transactions on Learning Technologies, vol. 9, no. 1, pp. 18-30, Jan.-March 1 2016. doi: 10.1109/TLT.2015.2419253

- [Axup, 2014] Axup, J., Thomas, A., Waldman, A., Faulkner, S., Odman-Govender, C., St.Leger, J., Polka, J., Gregg, M. & Johnson, B.D., "The World of Making," in *Computer*, vol.47, no.12, pp.24-40, Dec. 2014
- [Babcock, 2004] Babcock, P., "Shedding Light on Knowledge Management." *HR Magazine* 49 (5). 2004. Disponible en: <https://www.shrm.org/hr-today/news/hr-magazine/Pages/0504covstory.aspx>, visitada el 6 de julio de 2018
- [Backer, 2007] Backer, A., "Computational Physics Education with Python", in *Computing in Science & Engineering*, vol. 9, no. 3, pp. 30-33, May-June 2007. doi: 10.1109/MCSE.2007.48
- [Balaguer, 2017] Balaguer Álvarez, I. J., "Introduction to Robotics: Importance of a Summer Camp as a Recruiting Tool for Future University Students," in *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 12, no. 2, pp. 71-75, May 2017. doi: 10.1109/RITA.2017.2697739
- [Balakrishnan, 2016] Balakrishnan, B. & Low, F. S., "Learning Experience and Socio-Cultural Influences on Female Engineering Students' Perspectives on Engineering Courses and Careers". *Minerva*, 2016, 54(2), pp.219-239. doi: 10.1007/s11024-016-9295-8
- [Barai, 2017] Barai, S., Biswas, D. & Sau, B., "Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique," 2017 IEEE Conference on Antenna Measurements & Applications (CAMA), Tsukuba, 2017, pp. 170-173. doi: 10.1109/CAMA.2017.8273392
- [Barak, 2016] Barak, M. & Levenberg, A. (2016), Flexible thinking in learning: An individual differences measure for learning in technology-enhanced environments, *Computers & Education*, Volume 99, August 2016, Pages 39-52. doi: 10.1016/j.compedu.2016.04.003
- [Bart, 2017] Bart, A. C., Tibau, J., Tilevich, E., Shaffer, C. A. & Kafura, D., "BlockPy: An Open Access Data-Science Environment for Introductory Programmers", in *Computer*, vol. 50, no. 5, pp. 18-26, May 2017. doi: 10.1109/MC.2017.132
- [Basanta-Val, 2016] Basanta-Val, P., Audsley, N. C., Wellings, A. J., Gray, I. & Fernández-García, N., "Architecting Time-Critical Big-Data Systems," in *IEEE Transactions on Big Data*, vol. 2, no. 4, pp. 310-324, Dec. 1 2016.. doi: 10.1109/TBDDATA.2016.2622719
- [BAUL, 2016] El Baúl de Alexandra [Online]. Disponible: <http://elbauldealexandra.com/>, visitada el 6 de julio de 2018
- [Beagleboard, 2018] Beagleboard. [Online]. Disponible en: <http://beagleboard.org>, visitada el 6 de julio de 2018
- [Beasley, 2014] Beasley, C., "Passionate Projects: This Is What I Made", in *Computer*, vol. 47, no. 12, pp. 42-54, Dec. 2014. doi: 10.1109/MC.2014.364
- [Beck, 2000] Beck, K., "Extreme Programming Explained: Embrace Change", Addison Wesley, 2000, ISBN: 0201616416

- [Bethke, 2013] Bethke Wendell, K. & Rogers, C. (2013), Engineering Design-Based Science, Science Content Performance, and Science Attitudes in Elementary School. *Journal of Engineering Education*, 102: 513–540. doi: 10.1002/jee.20026
- [Beyers, 2017] Beyers, R. N. van der Merwe, L., "Initiating a pipeline for the computer industry: Using scratch and LEGO robotics," 2017 Conference on Information Communication Technology and Society (ICTAS), Umhlanga, 2017, pp. 1-7. doi: 10.1109/ICTAS.2017.7920646
- [Bicchi, 2005] Bicchi, A., Caiti, A., Pallottino, L. & Tonietti, G., "Online robotic experiments for tele-education at the University of Pisa", *J. Robotic Syst.*, vol. 22, no. 4, pp. 217-230, Apr. 2005. doi: 10.1002/rob.20061
- [Blázquez, 2013] Blázquez, M. (2013), Teaching technology with CLIL methodology: A case study, *Global Engineering Education Conference (EDUCON)*, 2013 IEEE, Berlin, 2013, pp. 581-590. doi: 10.1109/EduCon.2013.6530164
- [BlueTerm, 2018] Página Web de BlueTerm. [Online]. Disponible en: <http://py-masde.es/blueterm>, visitada el 6 de julio de 2018
- [Boaroli, 2017] Boaroli, L., Spacek, A. D., Izidoro, C. L., Mota Neto, J., Maestrelli, E. & Ando Junior, O. H., "Data Monitoring and Hardware Control for App Android by Bluetooth Communication for Laboratory Teaching in Electrical Engineering Courses," in *IEEE Latin America Transactions*, vol. 15, no. 1, pp. 31-39, Jan. 2017. doi: 10.1109/TLA.2017.7827885
- [Bocci, 2014] Bocci, V., Chiodi, G., Iacoangeli, F., Nuccetelli, M. & Recchia, L., "The ArduSiPM a compact trasportable Software/Hardware Data Acquisition system for SiPM detector," 2014 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Seattle, WA, 2014, pp. 1-5. doi: 10.1109/NSSMIC.2014.7431252
- [Booch, 2007] Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J. & Houston, K. A., "Object-Oriented Analysis and Design with Applications", 3rd Edition. Addison-Wesley Professional, 2007.
- [Borg Bartolo, 2017] Borg Bartolo, J., Degano, M., Espina, J. & Gerada, C., "Design and Initial Testing of a High-Speed 45-kW Switched Reluctance Drive for Aerospace Application," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 988-997, Feb. 2017. doi: 10.1109/TIE.2016.2618342
- [Bovier, 2016] Bovier, F., Caggianese, G., De Pietro, G., Gallo, L. & Neroni, P., "An Interactive 3D Holographic Pyramid for Museum Exhibition," 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Naples, 2016, pp. 428-434. doi: 10.1109/SITIS.2016.73

- [Boyi, 2014] Boyi, X., Li D. X., Hongming, C., Cheng, X., Jingyuan, H. & Fenglin B., "Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services," in *Industrial Informatics, IEEE Transactions on*, vol.10, no.2, pp.1578-1586, May 2014.
- [Brennan, 2013] Brennan, K., "Learning Computing through Creating and Connecting," in *Computer*, vol. 46, no. 9, pp. 52-59, September 2013. doi: 10.1109/MC.2013.229
- [Brown, 2002] Brown, D., "An Introduction to Object-Oriented Analysis: Objects and UML in Plain English". Wiley, 2002
- [Brox, 2016] Brox, P., Huertas-Sánchez, G., López-Angulo, A., Álvarez-Mora, M. & Haya, I., "Design of sensory systems using the platform Arduino by undergraduate Physics students," 2016 *Technologies Applied to Electronics Teaching (TAE)*, Seville, 2016, pp. 1-6. doi: 10.1109/TAE.2016.7528356
- [Bruneo, 2017] Bruneo, D., Distefano, S., Longo, F., Merlino, G., Puliafito, A. & Zaia, A., "Head in a Cloud: An approach for Arduino YUN virtualization," 2017 *Global Internet of Things Summit (GIOTS)*, Geneva, 2017, pp. 1-6. doi: 10.1109/GIOTS.2017.8016263
- [Buehler, 2015] Buehler, E., Easley, W., McDonald, S., Comrie, N. & Hurst, A., "Inclusion and education: 3D printing for integrated classrooms". In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility* (pp. 281-290). 2015. ACM. doi: 10.1145/2700648.2809844
- [Calidad, 2018] Calidad de clases. [Online]. Disponible en: <https://desarrolloweb.com/articulos/analisis-diseno-orientado-objetos-objetivos.html>, visitada el 6 de julio de 2018
- [Călinoiu, 2014] Călinoiu, D., Ionel, R., Lascu, M. & Cioablă, A., "Arduino and LabVIEW in educational remote monitoring applications," 2014 *IEEE Frontiers in Education Conference (FIE) Proceedings*, Madrid, 2014, pp. 1-5. doi: 10.1109/FIE.2014.7044027
- [Cápay, 2015] Cápay, M., "Engaging games with the computer science underlying concepts". *International Conference on Interactive Collaborative Learning (ICL) 2015*, Florence, pp. 975-979. doi: 10.1109/ICL.2015.7318160
- [Carro, 2012a] Carro, G., Sancristobal, E., Martin, S., Castro, M. & Mur, F., "Flexibility of wireless technologies in learning in robotic laboratories," in *Global Engineering Education Conference (EDUCON)*, 2012 *IEEE*, vol., no., pp.1-5, 17-20 April 2012. doi: 10.1109/EDUCON.2012.6201060
- [Carro, 2012b] Carro, G., Martin, S., Sancristobal, E., Mur, F. & Castro, M., "Robotics, the New Industrial Revolution", in *IEEE Technology and Society Magazine*, vol. 31, no. 2, pp. 51-58, Summer 2012. doi: 10.1109/MTS.2012.2196595
- [Carro, 2013] Carro, G., Castro, M., Sancristobal, E., Latorre, M., Diaz, G., Martin, S. & Losada, P., "Scaffolding online laboratory experiences as inclusive and motivational tools for students and teachers," 2013 *IEEE Frontiers in Education Conference (FIE)*, Oklahoma City, OK, 2013, pp. 1921-1927. doi: 10.1109/FIE.2013.6685170

- [Carro, 2014a] Carro, G., Castro, M., Sancristobal, E., Diaz, G., Mur, F., Latorre, M., Chaparro, M., Lopez-Rey, A., Salzmann, C. & Gillet, D., "The color of the light: A remote laboratory that uses a smart device that connects teachers and students," in Global Engineering Education Conference (EDUCON), 2014 IEEE , vol., no., pp.854-860, 3-5 April 2014. doi: 10.1109/EDUCON.2014.6826196
- [Carro, 2014b] Carro, G., "Sistema de integración de laboratorios remotos de robótica (silarr): laboratorio remoto de robótica, escalable, versátil y modular". Tesis doctoral. Escuela Técnica Superior de Ingenieros Industriales Universidad Nacional de Educación a Distancia (UNED). Madrid, 2014.
- [Carro, 2015a] Carro, G., Sancristobal, E., Castro, M. & Mur, F., "From RGB led laboratory to servomotor control with websockets and IoT as educational tool," in Remote Engineering and Virtual Instrumentation (REV), 2015 12th International Conference on, vol., no., pp.32-36, 25-27 Feb. 2015. doi: 10.1109/REV.2015.7087259
- [Carro, 2015b] Carro, G., Carrasco, R., Plaza, P., Canas, M.A., Sancristobal, E., Castro, M. & Mur, F., "Mechatronics and robotics as motivational tools in remote laboratories," in Global Engineering Education Conference (EDUCON), 2015 IEEE , vol., no., pp.118-123, 18-20 March 2015. doi: 10.1109/EDUCON.2015.7095960
- [Carvalho, 2016] Carvalho, D. G. & Lins, W. C. B., "LabDuino: An open source tool for science education," 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 2016, pp. 1-5. doi: 10.1109/FIE.2016.7757360
- [Carvalho, 2016] Carvalho, D. G. & Lins, W. C. B., "LabDuino: An open source tool for science education," 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 2016, pp. 1-5. doi: 10.1109/FIE.2016.7757360
- [Chancharoen, 2014] Chancharoen, R., Sripakagorn, A. & Maneeratana, K., "An Arduino kit for learning mechatronics and its scalability in semester projects," 2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), Wellington, 2014, pp. 505-510. doi: 10.1109/TALE.2014.7062606
- [Charlton, 2016] Charlton, P. & Avramides, K., "Knowledge Construction in Computer Science and Engineering when Learning Through Making," in *IEEE Transactions on Learning Technologies*, vol. 9, no. 4, pp. 379-390, Oct.-Dec. 1, 2016. doi: 10.1109/TLT.2016.2627567
- [Checkland, 2006] Checkland, P. B. & Poulter, J., "Learning for Action: A short definitive account of Soft Systems Methodology and its use for Practitioners, teachers and Students". Wiley, 2006. ISBN 0-470-02554-9
- [Chen, 2015] Chen, W. F., Chou, P. N. & Carey, R., "Using 3-D open source software to facilitate students' spatial visualization skills", 2015 IEEE Frontiers in Education Conference (FIE), El Paso, TX, 2015, pp. 1-3. doi: 10.1109/FIE.2015.7344302

- [Chen, 2017] Chen, P. & Huang, R., "Design Thinking in App Inventor Game Design and Development: A Case Study", 2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT), Timisoara, 2017, pp. 139-141. doi: 10.1109/ICALT.2017.161
- [Cheng, 2017] Cheng, S. M., Chen, P. Y., Lin, C. C. & Hsiao, H. C., "Traffic-Aware Patching for Cyber Security in Mobile IoT," in IEEE Communications Magazine, vol. 55, no. 7, pp. 29-35, 2017.. doi: 10.1109/MCOM.2017.1600993
- [Christensen, 2015] Christensen, R., Knezek, G. & Tyler-Wood, T. J Sci Educ Technol (2015) 24: 898. doi: 10.1007/s10956-015-9572-6
- [Chtourou, 2017] Chtourou, S., Kharrat, M., Ben Amor, N., Jallouli, M. & Abid, M., "Easing the development of android apps to create electronic prototypes: IOIO+App Inventor", in IET Circuits, Devices & Systems, vol. 11, no. 4, pp. 310-320, 7 2017. doi: 10.1049/iet-cds.2016.0381
- [CleanSpeak, 2018] Página web de CleanSpeak profanity filter. [Online]. Disponible: <https://www.inversoft.com/products/profanity-filter>, visitada el 6 de julio de 2018
- [Cline, 1999] Cline, A. 1999. "Joint Application Development (JAD) for Requirements Collection and Management." Carolla Development, Inc. [Online]. Disponible: <http://www.carolla.com/wp-jad.htm>, visitada el 6 de julio de 2018
- [Colombo, 2017] Colombo, A. W., Karnouskos, S., Kaynak, O., Shi, Y. & Yin, S., "Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution," in IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 6-16, March 2017.. doi: 10.1109/MIE.2017.2648857
- [Conceptos, 2018a] Conceptos de la metodología orientada a objetos. [Online]. Disponible en: http://profesores.fi-b.unam.mx/carlos/aydoo/conceptos_oo.html, visitada el 6 de julio de 2018
- [Conceptos, 2018b] Conceptos básicos de la metodología orientada a objetos. [Online]. Disponible en: <https://ricondelzorro.wordpress.com/conceptos-basicos-de-pooprogramacion-orientada-objetos/>, visitada el 6 de julio de 2018
- [Conversión, 2018] Conversión de divisas [Online]. Disponible en: <http://www.x-rates.com/calculator/?from=USD&to=EUR>, visitada el 6 de julio de 2018
- [Crumble, 2018] Crumble. [Online]. Disponible en: <http://redfernelectronics.co.uk/crumble>, visitada el 6 de julio de 2018
- [Cruz-Martín, 2012] Cruz-Martín, A., Fernández-Madrigo, J. A., Galindo, C., González-Jiménez, J., Stockmans-Daou, C. & Blanco-Claraco, J.L., "A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses", Computers & Education, Volume 59, Issue 3, November 2012, Pages 974-988. doi: 10.1016/j.compedu.2012.03.026
- [Cubillo, 2017] Cubillo, J., Martín-Gutiérrez, S. & Castro-Gil, M., "Design of a 3D Printing Remote Lab" . DYNA, 92(1). 15. 2017. doi: 10.6036/8044

- [Cui, 2012] Cui, L., Tso, F.P., Yao, D. & Jia, W., "WeFiLab: A Web-Based WiFi Laboratory Platform for Wireless Networking Education," in Learning Technologies, IEEE Transactions on, vol.5, no.4, pp.291-303, Oct.-Dec. 2012. doi: 10.1109/TLT.2012.6
- [Currás-Francos, 2014] Currás-Francos, M. d. C., Diz-Bugarín, J., García-Vila, J. R. & Orte-Caballero, A., "Cooperative Development of an Arduino-Compatible Building Automation System for the Practical Teaching of Electronics," in IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 9, no. 3, pp. 91-97, Aug. 2014. doi: 10.1109/RITA.2014.2340015
- [Daros, 2015] Daros, M. R., de Lima, J. P. C., Rochadel, W., Bento Silva, J. & Simão, J. S., "Remote experimentation in basic education using an architecture with Raspberry Pi," 2015 3rd Experiment International Conference (exp.at'15), Ponta Delgada, 2015, pp. 75-78. doi: 10.1109/EXPAT.2015.7463218
- [Davidson, 1999] Davidson, E. J., "Joint application design (JAD) in practice", Journal of Systems and Software, Volume 45, Issue 3, 15 March 1999, Pages 215-223, ISSN 0164-1212
- [de Kereki, 2016] de Kereki, I. F. & Manataki, A., "'Code Yourself' and 'A Programmer': A bilingual MOOC for teaching computer science to teenagers," 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 2016, pp. 1-9. doi: 10.1109/FIE.2016.7757569
- [de la Guía, 2016] de la Guía, E., Camacho, V. L., Orozco-Barbosa, L., Brea Luján, V. M., Penichet, V. M. R. & Lozano Pérez, M., "Introducing IoT and Wearable Technologies into Task-Based Language Learning for Young Children," in *IEEE Transactions on Learning Technologies*, vol. 9, no. 4, pp. 366-378, Oct.-Dec. 1 2016. doi: 10.1109/TLT.2016.2557333
- [de Matos, 2018] de Matos Ferreira Bastos, A. M. & Ribeiro da Cunha, J. F., "The scratch pilot experience in 1st cycle of basic education schools the students perspective," 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, 2018, pp. 1-7. doi: 10.23919/CISTI.2018.8399251
- [Deanne, 2018] Deanne Bell's Future Engineers Program Connects Students With NASA And STEM. [Online]. Disponible en: <https://www.forbes.com/sites/kevinmurnane/2016/12/12/deanne-bells-future-engineers-program-connects-students-with-nasa-and-stem/#773c6428ff96>, visitada el 6 de julio de 2018
- [Delgado, 2015] Delgado Kloos, C., Muñoz-Merino, P.J., Alario-Hoyos, C., Estevez Ayres, I. & Fernandez-Panadero, C., "Mixing and blending MOOC Technologies with face-to-face pedagogies," in Global Engineering Education Conference (EDUCON), 2015 IEEE, vol., no., pp.967-971, 18-20 March 2015. doi: 10.1109/EDUCON.2015.7096090
- [Delice, 2006] Delice, A. & Roper, T., "Implications of a comparative study for mathematics education in the English education system," in Teaching Mathematics and Its Applications: International Journal

of the IMA, vol. 25, no. 2, pp. 64-72, June 2006. doi: 10.1093/teamat/hri007

- [Devi,, 2018] Devi, S. & Saravanan, M., "An innovative modular device and wireless control system enabling thermal and pressure sensors using FPGA on real-time fault diagnostics of steam turbine functional deterioration", *Mechanical Systems and Signal Processing*, Volume 102, 2018, Pages 312-328. doi: 10.1016/j.ymsp.2017.09.034
- [DFRobot, 2018] Página web de DFRobot. [Online]. Disponible en: <http://www.dfrobot.com>, visitada el 6 de julio de 2018
- [DFRobotShop, 2018] DFRobotShop Rover Tutorial – Control With Android App / Bluetooth. [Online]. Disponible: <http://www.dfrobot.com/index.php?route=DFblog/blog&id=402>, visitada el 6 de julio de 2018
- [Díaz-Báñez, 2017] Díaz-Báñez, J. M., Caraballo, L. E., Lopez, M. A., Bereg, S., Maza, I. & Ollero, A., "A General Framework for Synchronizing a Team of Robots Under Communication Constraints," in *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 748-755, June 2017. doi: 10.1109/TRO.2017.2676123
- [DIEEC UNED, 2018] DIEEC UNED. [Online]. Disponible en:<http://www.ieec.uned.es/>, visitada el 6 de julio de 2018
- [Dobrilovic, 2015] Dobrilovic, D., Stojanov, Z., Odadzic, B. & Sinik, V., "Platform for teaching communication systems based on open-source hardware," in *Global Engineering Education Conference (EDUCON)*, 2015 IEEE , vol., no., pp.737-741, 18-20 March 2015
- [Dusmez, 2014] Dusmez, S. & Khaligh, A., "A Charge-Nonlinear-Carrier-Controlled Reduced-Part Single-Stage Integrated Power Electronics Interface for Automotive Applications," in *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 1091-1103, March 2014. doi: 10.1109/TVT.2013.2284592
- [EAGLE, 2018] Página Web de EAGLE. [Online]. Disponible en: <https://www.autodesk.com/products/eagle/overview>, visitada el 6 de julio de 2018
- [Echeverría, 2017] Echeverría, L., Cobos, R., Machuca, L. & Claros, I., "Using collaborative learning scenarios to teach programming to non-CS majors". *Comput Appl Eng Educ.* 2017;25:719–731. doi: 10.1002/cae.21832
- [EDUCON 2017, 2018] EDUCON 2017. [Online]. Disponible en: <http://www.educon-conference.org/educon2017/>, visitada el 6 de julio de 2018
- [EDUCON 2018, 2018] EDUCON 2018. [Online]. Disponible en: <http://www.educon-conference.org/educon2018/>, visitada el 6 de julio de 2018
- [El Circulo, 2018] El Circulo. [Online]. Disponible: <http://www.ccelcirculo.es/>, visitada el 6 de julio de 2018
- [Elhattab, 2017] Elhattab, M. K., Elmesalawy, M. M. & Ibrahim, I. I., "Opportunistic Device Association for Heterogeneous Cellular Networks With H2H/IoT Co-Existence Under QoS Guarantee," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1360-1369, Oct. 2017. doi: 10.1109/JIOT.2017.2702199

- [Enisz, 2014] Enisz, K., Fodor, D., Szalay, I. & Kovacs, L., "Reconfigurable real-time hardware-in-the-loop environment for automotive electronic control unit testing and verification," in IEEE Instrumentation & Measurement Magazine, vol. 17, no. 4, pp. 31-36, Aug. 2014. doi: 10.1109/MIM.2014.6873729
- [ESP8266, 2018] ESP8266 forum [Online]. Disponible en: <http://www.esp8266.com/viewforum.php?f=17>, visitada el 6 de julio de 2018
- [Estabilidad, 2018] Estabilidad de clases. [Online]. Disponible en: <https://www.monografias.com/trabajos14/progorie/progorie.shtml>, visitada el 6 de julio de 2018
- [EUROPE, 2018] EUROPE 2020. [Online]. Disponible: <http://ec.europa.eu/europe2020/>, visitada el 6 de julio de 2018
- [Fan, 2018] Fan, H., Chen, W., Li, Y., Zhang, J., Ye, X. & Feng, Q., "Promoting engineering education by scientific research," 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, pp. 60-64. doi: 10.1109/EDUCON.2018.8363209
- [Farkhadov, 2016] Farkhadov, M., Vaskovsky, S. & Nadeinsky, L., "Interactive online learning, E-education platform with multimedia computer technologies and internet-based laboratories," 2017 5th International Conference on Control, Instrumentation, and Automation (ICCIA), Shiraz, 2017, pp. 212-215. doi: 10.1109/ICCIAutom.2017.8258680
- [Ferdowsi, 2015] Ferdowsi, M., Benigni, A., Lowen, A., Zargar, B., Monti, A. & Ponci, F., "A Scalable Data-Driven Monitoring Approach for Distribution Systems," Instrumentation and Measurement, IEEE Transactions on , vol.64, no.5, pp.1300,1313, May 2015. doi: 10.1109/TIM.2015.2398991
- [Fernández, 2017] Fernández, C., Vicente, M. A., Galotto, M. M., Martínez-Rach, M. & Pomares, A., "Improving student engagement on programming using app development with Android devices". Comput Appl Eng Educ. 2017;25:659–668. doi: 10.1002/cae.21827
- [FIE 2016, 2018] FIE 2016. [Online]. Disponible en: <http://fie2016.org/>, visitada el 6 de julio de 2018
- [FIE 2018, 2018] FIE 2018. [Online]. Disponible en: <http://fie2018.org/>, visitada el 6 de julio de 2018
- [Fleenor, 2015] Fleenor, H., Alexander, B. & Chouchane, R., "Tech girls shine: Increasing commitment to diversity through outreach," 2015 Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), Charlotte, NC, USA, 2015, pp. 1-1. doi: 10.1109/RESPECT.2015.7296519
- [FlexiEspacio, 2018] Página de Facebook de FlexiEspacio. [Online]. Disponible en: <https://www.facebook.com/flexiespacio/>, visitada el 6 de julio de 2018
- [Florez, 2014] Florez Aristizabal, L., Almario, D.F. & Lopez, J.A., "Development of an Android App as a learning tool of dynamic systems and automatic control," in Engineering Mechatronics and Automation

(CIIMA), 2014 III International Congress of , vol., no., pp.1-5, 22-24 Oct. 2014.

- [Flynn, 2015] Flynn, D., "Challenges for Power Supplies in Medical Equipment: Ensuring patient and operator safety," in IEEE Power Electronics Magazine, vol. 2, no. 2, pp. 32-37, June 2015. doi: 10.1109/MPEL.2015.2420231
- [Forms, 2018] Google Forms. [Online]. Disponible en: <https://www.google.com/forms/about/>, visitada el 6 de julio de 2018
- [Förster, 2018] Förster, E. C., Förster, K. T. & Löwe, T., "Teaching programming skills in primary school mathematics classes: An evaluation using game programming," 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, pp. 1504-1513. doi: 10.1109/EDUCON.2018.8363411
- [FP, 2018] TODO FP. [Online]. Disponible: <http://www.todofp.es/>, visitada el 6 de julio de 2018
- [Fraanje, 2016] Fraanje, R., Koreneef, T., Le Mair, A. & de Jong, S., "Python in robotics and mechatronics education", 2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM), Compiègne, 2016, pp. 014-019. doi: 10.1109/MECATRONICS.2016.7547108
- [Friedman, 1990] Friedman, M., "Teoría de los precios", Alianza Editorial. 1990. ISBN 9788420680545.
- [Funke, 2017] Funke, A., Geldreich, K. & Hubwieser, P., "Analysis of scratch projects of an introductory programming course for primary school students," 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 1229-1236. doi: 10.1109/EDUCON.2017.7943005
- [Gale, 2007] Gale, R., Karp, T., Lowe, L. & Medina, V., "Generation NXT," 2007 IEEE Meeting the Growing Demand for Engineers and Their Educators 2010-2020 International Summit, Munich, 2007, pp. 1-13. doi: 10.1109/MGDETE.2007.4760354
- [Garrigós, 2017] Garrigós, A., Marroquí, D., Blanes, J. M., Gutiérrez, R., Blanquer, I. & Cantó, M., "Designing Arduino electronic shields: Experiences from secondary and university courses," 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 934-937. doi: 10.1109/EDUCON.2017.7942960
- [Gatward, 2012] Gatward, A. J., "An internet-attached robot system supporting teaching in higher education". Tesis doctoral. University of Reading. Berkshire, Reino Unido, 2012.
- [Genuino, 2018a] Genuino Yún Shield shop [Online]. Disponible en: <https://store.arduino.cc/product/GSX00102>, visitada el 6 de julio de 2018
- [Genuino, 2018b] Genuino MKR1000 shop [Online]. Disponible en: <https://store.arduino.cc/product/GBX00011>, visitada el 6 de julio de 2018

- [GerbV, 2018] Página web de GerbV. [Online]. Disponible en: <http://gerbv.geda-project.org>, visitada el 6 de julio de 2018
- [Ghosh, 2017] Ghosh, S., Panigrahi, P. K. & Parhi, D. R., "Analysis of FPA and BA meta-heuristic controllers for optimal path planning of mobile robot in cluttered environment," in IET Science, Measurement & Technology, vol. 11, no. 7, pp. 817-828, 10 2017. doi: 10.1049/iet-smt.2016.0273
- [Gil, 2014] Gil, J. J., Diaz, I., Justo, X. & Ciäurriz, P., "Educational haptic controller based on Arduino platform," 2014 XI Tecnologías Aplicadas a la Enseñanza de la Electrónica (Technologies Applied to Electronics Teaching) (TAE), Bilbao, 2014, pp. 1-7. doi: 10.1109/TAE.2014.6900140
- [Gillet, 2017] Gillet, D., Rodríguez-Triana, M. J., de Jong, T., Bollen, L. & Dikke, D., "Cloud ecosystem for supporting inquiry learning with online labs: Creation, personalization, and exploitation," 2017 4th Experiment@International Conference (exp.at'17), Faro, 2017, pp. 208-213. doi: 10.1109/EXPAT.2017.7984406
- [Github, 2018] Github. [Online]. Disponible en: <https://github.com>, visitada el 6 de julio de 2018
- [Gomez-de-Gabriel, 2011] Gomez-de-Gabriel, J. M., Mandow, A., Fernandez-Lozano, J. & Garcia-Cerezo, A. J., "Using LEGO NXT Mobile Robots With LabVIEW for Undergraduate Courses on Mechatronics", in IEEE Transactions on Education, vol. 54, no. 1, pp. 41-47, Feb. 2011. doi: 10.1109/TE.2010.2043359
- [Gómez-de-Gabriel, 2015] Gómez-de-Gabriel, J. M., Mandow, A., Fernández-Lozano, J. & García-Cerezo, A., "Mobile Robot Lab Project to Introduce Engineering Students to Fault Diagnosis in Mechatronic Systems," in IEEE Transactions on Education, vol. 58, no. 3, pp. 187-193, Aug. 2015. doi: 10.1109/TE.2014.2358551
- [González, 2017] González, N. (2017) "Robótica Educativa en la Educación Formal," (Trabajo Fin de Máster no publicado), Universidad Nebrija, Madrid, España.
- [Graven, 2016] Graven, O. H. & Bjørk, J., "The use of an Arduino pocket lab to increase motivation in Electrical engineering students for programming," 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Bangkok, 2016, pp. 239-243. doi: 10.1109/TALE.2016.7851800
- [Grover, 2014] Grover, R., Krishnan, S., Shoup, T. & Khanbaghi, M., "A competition-based approach for undergraduate mechatronics education using the arduino platform," Fourth Interdisciplinary Engineering Design Education Conference, Santa Clara, CA, 2014, pp. 78-83. doi: 10.1109/IEDEC.2014.6784685
- [Guerra, 2015] Guerra, H., Cardoso, A., Sousa, V., Leitão, J., Graveto, V. & Gomes, L. M., "Demonstration of programming in Python using a remote

- lab with Raspberry Pi," 2015 3rd Experiment International Conference (exp.at'15), Ponta Delgada, 2015, pp. 101-102. doi: 10.1109/EXPAT.2015.7463226
- [Guo, 2017] Guo, J. et al., "Novel Radiation-Hardened-by-Design (RHBD) 12T Memory Cell for Aerospace Applications in Nanoscale CMOS Technology," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1593-1600, May 2017. doi: 10.1109/TVLSI.2016.2645282
- [Hajdarevic, 2014] Hajdarevic, K., Konjicija, S. & Subasi, A., "Svmlink VOIP implementation using raspberry Pi in education and disaster relief situations," 2014 X International Symposium on Telecommunications (BIHTEL), Sarajevo, 2014, pp. 1-6. doi: 10.1109/BIHTEL.2014.6987647
- [Haro, 2014] Haro, L. F. D., Cordoba, R., Rojo Rivero, J. I., Diez de la Fuente, J., Avendano Peces, D. & Bermudo Mera, J. M., "Low-Cost Speaker and Language Recognition Systems Running on a Raspberry Pi," Latin America Transactions, IEEE (Revista IEEE America Latina) , vol.12, no.4, pp.755,763, June 2014. doi: 10.1109/TLA.2014.6868880
- [Hashem Eiza, 2017] Hashem Eiza, M. & Ni, Q., "Driving with Sharks: Rethinking Connected Vehicles with Vehicle Cybersecurity," in IEEE Vehicular Technology Magazine, vol. 12, no. 2, pp. 45-51, June 2017.. doi: 10.1109/MVT.2017.2669348
- [Hatzigianni, 2016] Hatzigianni, M., Gregoriadis, A. & Fler, M. (2016), Computer use at schools and associations with social-emotional outcomes – A holistic approach. Findings from the longitudinal study of Australian Children, Computers & Education, Volume 95, April 2016, Pages 134-150. doi: 10.1016/j.compedu.2016.01.003
- [Hawley, 2013] Hawley, C. E., Cardoso, E. & McMahan, B. T., "Adolescence to adulthood in STEM education and career development: The experience of students at the intersection of underrepresented minority status and disability". Journal of Vocational Rehabilitation, 2013, 39(3), pp.193-204. doi: 10.3233/JVR-130655
- [Hernandez, 2012] Hernandez, U., "Metodología de control independiente de Instrumentos y experimentos para su despliegue en laboratorios remotos". Tesis doctoral. Facultad de Ingeniería, Universidad de Deusto. Bilbao, 2012.
- [Hertzog, 2016] Hertzog, P. E. & Swart, A. J., "Arduino — Enabling engineering students to obtain academic success in a design-based module," 2016 IEEE Global Engineering Education Conference (EDUCON), Abu Dhabi, 2016, pp. 66-73. doi: 10.1109/EDUCON.2016.7474533
- [Hew, 2016] Hew, K. F., Huang, B., Chu, K. W. S. & Chiu, D. K.W. (2015), Engaging Asian students through game mechanics: Findings from two experiment studies, Computers & Education, Volumes 92–93, January–February 2016, Pages 221-236. doi: 10.1016/j.compedu.2015.10.010

- [Hijón-Neira, 2017] Hijón-Neira, R., Santacruz-Valencia, L., Pérez-Marín, D. & Gómez-Gómez, M., "An analysis of the current situation of teaching programming in Primary Education," 2017 International Symposium on Computers in Education (SIIE), Lisbon, 2017, pp. 1-6. doi: 10.1109/SIIE.2017.8259650
- [Hirabayashi, 2015] Hirabayashi, T., Yoshie, M., Takahashi, K. & Shirai, K., "Considering methods of executing work in undersea resources development," 2015 IEEE Underwater Technology (UT), Chennai, 2015, pp. 1-8. doi: 10.1109/UT.2015.7108247
- [i90, 2018] Wireless Networked Autonomous Mobile Robot with High Resolution Pan-Tilt-Zoom Camera i90. [Online]. Disponible: http://www.drrobot.com/products_item.asp?itemNumber=i90, visitada el 6 de julio de 2018
- [Ibarra, 2010] Ibarra Esquer, J., López-Morteo, G., Flores Rios, B. & Castro Peña-loza, U., "Utilizando Scratch en el Desarrollo de Exhibiciones para un Museo Interactivo". XXXVI Conferencia Latinoamericana de Informática - CLEI 2010.
- [ICL 2017, 2018] ICL 2017. [Online]. Disponible en: <http://www.icl-conference.org/icl2017/>, visitada el 6 de julio de 2018
- [IEEE, 2018] IEEE. [Online]. Disponible: <https://www.ieee.org/>, visitada el 6 de julio de 2018
- [Incomaz, 2018] Incomaz. [Online]. Disponible: <http://www.incomaz.com/>, visitada el 6 de julio de 2018
- [Indeed, 2018] Salarios Indeed [Online]. Disponible en: <https://www.indeed.es/salaries/>, visitada el 6 de julio de 2018
- [Indra, 2018] La Ciudad 2020 de Indra [Online]. Disponible: <http://www.indra-company.com/en/noticia/2020-city-ends-30-technological-solutions-smart-cities>, visitada el 6 de julio de 2018
- [Intel, 2018] Intel. [Online]. Disponible en: <http://www.intel.com>, visitada el 6 de julio de 2018
- [Irawati, 2016] Irawati, I. & Munajat, M. D. E., "Evaluation of Internet-Based Long Distance Education at Universitas Padjadjaran," 2016 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, 2016, pp. 1-5. doi: 10.1109/ICITSI.2016.7858206
- [Islam, 2017] Islam, M. M., Razzaque, M. A., Hassan, M. M., Ismail, W. N. & Song, B., "Mobile Cloud-Based Big Healthcare Data Processing in Smart Cities," in IEEE Access, vol. 5, no., pp. 11887-11899, 2017.. doi: 10.1109/ACCESS.2017.2707439
- [Jacobson, 2000] Jacobson, I., Booch, G. & Rumbaugh, J., "El Proceso Unificado de Desarrollo de software". 2000. Addison-Wesley.
- [Jamieson, 2015] Jamieson, P. & Herdtner, J., "More missing the Boat - Arduino, Raspberry Pi, and small prototyping boards and engineering education needs them". IEEE Frontiers in Education Conference (FIE), El Paso, TX, 2015, pp. 1-6. doi: 10.1109/FIE.2015.7344259

- [Jiménez, 2016] Jiménez López, A. F., Prieto Pelayo, M. C. & Ramírez Forero, Á., "Teaching Image Processing in Engineering Using Python", in IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 11, no. 3, pp. 129-136, Aug. 2016. doi: 10.1109/RITA.2016.2589479
- [Joardar, 2015] Joardar, S., Chatterjee, A. & Rakshit, A., "A Real-Time Palm Dorsa Subcutaneous Vein Pattern Recognition System Using Collaborative Representation-Based Classification," Instrumentation and Measurement, IEEE Transactions on , vol.64, no.4, pp.959,966, April 2015. doi: 10.1109/TIM.2014.2374713
- [JobTonic, 2018] Salarios JobTonic [Online]. Disponible en: <http://espana.jobtonic.es/salary/>, visitada el 6 de julio de 2018
- [Johnson, 2010] Johnson, J. & Dyer, J., "Developing an Educational Model of Internet-Based Learning in a Transatlantic Context," in Computing in the Global Information Technology (ICCGI), 2010 Fifth International Multi-Conference on, vol., no., pp.199-204, 20-25 Sept. 2010. doi: 10.1109/ICCGI.2010.50
- [Johnson, 2013] Johnson, B. D. (2013), Jimmy: The Robot That Changed It All, in Computer, vol. 46, no. 3, pp. 104-107, March 2013. doi: 10.1109/MC.2013.105
- [Jordan, 2014] Jordan, S. & Lande, M., "Might young makers be the engineers of the future?," in Frontiers in Education Conference (FIE), 2014 IEEE , vol., no., pp.1-4, 22-25 Oct. 2014
- [Jung, 2012] Jung, S., "Experiences in Developing an Experimental Robotics Course Program for Undergraduate Education", in IEEE Transactions on Education, vol. 56, no. 1, pp. 129-136, Feb. 2013. doi: 10.1109/TE.2012.2213601
- [Karp, 2011] Karp, T. & Schneider, A., "Evaluation of a K-8 LEGO robotics program". Frontiers in Education Conference (FIE), Rapid City, SD, 2011, pp. T1D-1-T1D-6. doi: 10.1109/FIE.2011.6142977
- [Kaynov, 2008] Kaynov, D., "Open motion control architecture for humanoid robots". Tesis doctoral. Departamento de Sistemas y Automática, Universidad Carlos III. Madrid, 2008.
- [Kendall, 2014] Kendall, K. E. & Kendall, J. E., "Systems Analysis and Design", 9th Edition, Pearson, 2014. ISBN13: 9780133023442. ISBN10: 0133023443.
- [Kim, 2015] Kim, C. M., Kim, D., Yuan, J., Hill, R. B., Doshi, P. & Thai, C. N., "Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching". Computers & Education, Volume 91, 15 December 2015, Pages 14-31. doi: 10.1016/j.compedu.2015.08.005
- [Kintsakis, 2017] Kintsakis, D. & Rangoussi, M., "An early introduction to STEM education: Teaching computer programming principles to 5th graders through an e-learning platform: A game-based approach", IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 17-23. doi: 10.1109/EDUCON.2017.7942816

- [Klassner, 2003] Klassner, F. & Anderson, S. D., "LEGO MindStorms: not just for K-12 anymore", in *IEEE Robotics & Automation Magazine*, vol. 10, no. 2, pp. 12-18, June 2003. doi: 10.1109/MRA.2003.1213611
- [Knipprath, 2018] Knipprath, H. et al., "STEM Education in Flanders: How STEM@school Aims to Foster STEM Literacy and a Positive Attitude towards STEM," in *IEEE Instrumentation & Measurement Magazine*, vol. 21, no. 3, pp. 36-40, June 2018. doi: 10.1109/MIM.2018.8360917
- [Królak, 2017] Królak, A., Chen, W., Sanderson, N. C. & Kessel, S., "The Accessibility of MOOCs for Blind Learners". In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 401-402). 2017. ACM. doi: 10.1145/3132525.3134796
- [Kumar, 2017] Kumar, N., Singh, M., Zeadally, S., Rodrigues, J. J. P. C. & Rho, S., "Cloud-Assisted Context-Aware Vehicular Cyber-Physical System for PHEVs in Smart Grid," in *IEEE Systems Journal*, vol. 11, no. 1, pp. 140-151, March 2017. doi: 10.1109/JSYST.2015.2437201
- [La Estera, 2018] La Estera. [Online]. Disponible: <http://laestera.blogspot.com.es/>, visitada el 6 de julio de 2018
- [Lade, 2017] Lade, P., Ghosh, R. & Srinivasan, S., "Manufacturing Analytics and Industrial Internet of Things," in *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 74-79, May-June 2017. doi: 10.1109/MIS.2017.49
- [Lee, 2014] Lee, W. C., Matusovich, H. M. & Brown, P. R., "Measuring underrepresented student perceptions of inclusion within engineering departments and universities". *International Journal of Engineering Education*, 2014, 30(1), pp.150-165.
- [LEGO, 2018] LEGO MINDSTORM Support. [Online]. Disponible: <http://www.lego.com/en-us/mindstorms/support>, visitada el 6 de julio de 2018
- [Leonard, 2016] Leonard, J., Buss, A., Gamboa, R. et al., "Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills", *J Sci Educ Technol* 25: 860. doi: <https://doi.org/10.1007/s10956-016-9628-2>
- [Lin, 2012] Lin C., Fung P. T., Di Y. & Weijia J., "WeFiLab: A Web-Based WiFi Laboratory Platform for Wireless Networking Education," in *Learning Technologies*, *IEEE Transactions on*, vol.5, no.4, pp.291-303, Oct.-Dec. 2012
- [Lin, 2017] Lin, Y., Conners, J. & Livengood, S., "Assessing the use of open source microcontroller board for teaching engine sensing and communication in automotive laboratory," 2017 *IEEE Frontiers in Education Conference (FIE)*, Indianapolis, IN, 2017, pp. 1-5. doi: 10.1109/FIE.2017.8190738
- [Lind, 2014] Lind, M., "Open Real-Time Control and Emulation of Robots and Production Systems". Tesis doctoral. Department of Productions and Quality Engineering, Norwegian University of Science and Technology. Trondheim, Noruega, 2012.

- [Liou, 2018] Liou, W. K. & Chang, C. Y., "Virtual reality classroom applied to science education," 2018 23rd International Scientific-Professional Conference on Information Technology (IT), Zabljak, 2018, pp. 1-4. doi: 10.1109/SPIT.2018.8350861
- [Liu, 2017a] Liu, G. P., "Predictive Control of Networked Multiagent Systems via Cloud Computing," in IEEE Transactions on Cybernetics, vol. 47, no. 8, pp. 1852-1859, Aug. 2017.. doi: 10.1109/TCYB.2017.2647820
- [Liu, 2017b] Liu, J., Xiong, K., Fan, P. & Zhong, Z., "RF Energy Harvesting Wireless Powered Sensor Networks for Smart Cities," in IEEE Access, vol. 5, no., pp. 9348-9358, 2017.. doi: 10.1109/ACCESS.2017.2703847
- [LOE, 2006] Ley Orgánica 2/2006, 3 de mayo, de Educación (LOE).
- [LOGSE, 1990] Ley Orgánica 1/1990, 3 de octubre, General del Sistema Educativo (LOGSE).
- [LOMCE, 2013] Ley Orgánica 8/2013, 9 de diciembre, para la Mejora de la Calidad de la Enseñanza (LOMCE).
- [Longo, 2015] Longo, A., Zappatore, M. & Bochicchio, M. A., "Collaborative learning from Mobile Crowd Sensing: A case study in electromagnetic monitoring," in Global Engineering Education Conference (EDUCON), 2015 IEEE , vol., no., pp.742-750, 18-20 March 2015
- [Lopes, 2016] Lopes, M., Gomes, I., Trindade, R., Silva, A. & Lima, A. C., "Web environment for programming and control of mobile robot in a remote laboratory," in *IEEE Transactions on Learning Technologies*, vol. PP, no. 99, pp. 1-1. doi: 10.1109/TLT.2016.2627565
- [LOU, 2001] Ley Orgánica 6/2001, 21 de diciembre, de Universidades (LOU).
- [LWMOOCS 2018, 2018] LWMOOCS 2018. [Online]. Disponible en: <http://learningwithmoocs2018.org/>, visitada el 6 de julio de 2018
- [Ma, 2015] Ma, J., Zhang, T. & Dong, M., "A Novel ECG Data Compression Method Using Adaptive Fourier Decomposition With Security Guarantee in e-Health Applications," in IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 3, pp. 986-994, May 2015.. doi: 10.1109/JBHI.2014.2357841
- [Macías, 2014] Macías, M. M., Agudo, J. E., Orellana, C. J. G., Velasco, H. M. G. & Manso, A. G., "The "mbed" platform for teaching electronics applied to product design," 2014 XI Tecnologías Aplicadas a la Enseñanza de la Electrónica (Technologies Applied to Electronics Teaching) (TAEE), Bilbao, 2014, pp. 1-6. doi: 10.1109/TAEE.2014.6900156
- [Makeblock, 2018] Starter Robot Kit from Makeblock. [Online]. Disponible: <http://www.makeblock.cc/starter-robot-kit>, visitada el 6 de julio de 2018
- [Manic, 2016] Manic, M., Wijayasekara, D., Amarasinghe, K. & Rodriguez-Andina, J. J., "Building Energy Management Systems: The Age of Intelligent and Adaptive Buildings," in IEEE Industrial Electronics Magazine, vol. 10, no. 1, pp. 25-39, March 2016.. doi: 10.1109/MIE.2015.2513749

- [Mare, 2016] Mare, R. M., Luiz Marte, C. & Cugnasca, C. E., "Visible Light Communication Applied to Intelligent Transport Systems: an Overview," in IEEE Latin America Transactions, vol. 14, no. 7, pp. 3199-3207, July 2016.. doi: 10.1109/TLA.2016.7587621
- [Marot, 2017] Marot, J. & Bourennane, S., "Raspberry Pi for image processing education," 2017 25th European Signal Processing Conference (EUSIPCO), Kos, 2017, pp. 2364-2366. doi: 10.23919/EUSIPCO.2017.8081633
- [Martin, 1991] Martin, J., "Rapid Application Development," Macmillan, New York, 1991.
- [Martin, 2010] Martin, S., "M2learn: framework for development of mobile and ubiquitous learning applications". Tesis doctoral. Escuela Técnica Superior de Ingenieros Industriales Universidad Nacional de Educación a Distancia (UNED). Madrid, 2010.
- [Martínez-Mediano, 2016] Martínez-Mediano, C. & Riopérez Losada, N., "Internet-Based Performance Support Systems in Engineering Education," in IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 12, no. 2, pp. 86-93, May 2017. doi: 10.1109/RITA.2017.2697778
- [Martínez-Santos, 2017] Martínez-Santos, J. C., Acevedo-Patino, O. & Contreras-Ortiz, S. H., "Influence of Arduino on the Development of Advanced Microcontrollers Courses," in IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 12, no. 4, pp. 208-217, Nov. 2017. doi: 10.1109/RITA.2017.2776444
- [Mateu, 2014] Mateu, J., Lasala, M. J. & Alamán, X., "VirtualTouch: a tool for developing mixed reality educational applications and an example of use for inclusive education". *International Journal of Human-Computer Interaction*, 2014, 30(10), pp.815-828. doi: 10.1080/10447318.2014.927278
- [Matsumoto, 2016] Matsumoto, S., Sato, T., Muneo, Y., Abukawa, K. & Katakura, K., "Prototype of underwater acoustic video camera," OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, 2016, pp. 1-4. doi: 10.1109/OCEANS.2016.7761454
- [Mavroudi, 2017] Mavroudi, A., Economides, A. A., Fragkou, O., et al. "Motivating students with Mobiles, Ubiquitous applications and the Internet of Things for STEM (MUMI4STEM)", IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 37-38. doi: 10.1109/EDUCON.2017.7942820
- [Mazumder, 2016] Mazumder, O., Kundu, A., Lenka, P., et al. "Robotic AFO to enhance walking capacity: initial development", Electronics Letters, vol. 52, no. 22, pp. 1840-1841, 10 27 2016. doi: 10.1049/el.2016.3056
- [mBot, 2018] mBot - STEM Educational Robot Kit for Kids from Makeblock. [Online]. Disponible: <http://www.makeblock.cc/mbot/>, visitada el 6 de julio de 2018
- [McGee, 2017] McGee, E. O. & Bentley, L., "The Troubled Success of Black Women in STEM". *Cognition and Instruction*, 2017, 35(4), pp.265. doi: 10.1080/07370008.2017.1355211

- [McKenna, 1998] McKenna, A. & Agogino, A., "A Web-Based Instructional Module for Teaching Middle School Students Engineering Design with Simple Machines*". *Journal of Engineering Education*, 87: 437–444. doi:10.1002/j.2168-9830.1998.tb00376.x
- [Menendez, 2017] Menendez, O., Auat Cheein, F. A., Perez, M. & Kouro, S., "Robotics in Power Systems: Enabling a More Reliable and Safe Grid," in *IEEE Industrial Electronics Magazine*, vol. 11, no. 2, pp. 22-34, June 2017.. doi: 10.1109/MIE.2017.2686458
- [Metodología, 2018a] Definición de Metodología RAE. [Online]. Disponible en: <http://dle.rae.es/?id=P7eTCPD>, visitada el 6 de julio de 2018
- [Metodología, 2018b] Definición de Metodología Wikipedia. [Online]. Disponible en: <https://es.wikipedia.org/wiki/Metodolog%C3%ADa>, visitada el 6 de julio de 2018
- [Metodología, 2018c] Definición de Metodología Tecnología e Informática. [Online]. Disponible en: <https://tecnologiaeinformaticapjg.jimdo.com/tecnologia/cuarto-periodo-1/3-periodo-1/>, visitada el 6 de julio de 2018
- [Microchip, 2018] Microchip Development Tools. [Online]. Disponible en: <http://www.terasic.com.tw/en>, visitada el 6 de julio de 2018
- [Milanovic, 2018] Milanovic, J. Z., Milanovic, P., Kragic, R. & Kostic, M., "'Do-It-Yourself' reliable pH-stat device by using open-source software, inexpensive hardware and available laboratory equipment". 2018. *PLoS ONE* 13(3): e0193744. doi: 10.1371/journal.pone.0193744
- [Mironova, 2017] Mironova, O., Amitan, I. & Vilipõld, J., "Programming basics for beginners: Experience of the institute of informatics at Tallinn University of Technology," 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 735-739. doi: 10.1109/EDUCON.2017.7942929
- [MIT Education, 2018a] Página web de MIT Education. [Online]. Disponible: <http://web.mit.edu/>, visitada el 6 de julio de 2018
- [MIT Education, 2018b] Página web de MIT Education Scratch. [Online]. Disponible: <https://scratch.mit.edu/>, visitada el 6 de julio de 2018
- [Mobility, 2018] The Mobility Division of Siemens in Spain [Online]. Disponible: <https://www.siemens.com/businesses/es/es/mobility.htm>, visitada el 6 de julio de 2018
- [Modelsim, 2018] Página web de Altera Modelsim. Disponible en: <https://www.altera.com/products/design-software/model---simulation/modelsim-altera-software.html>, visitada el 3 de agosto de 2017.
- [Mondada, 2017] Mondada, F. et al. (2017), Bringing Robotics to Formal Education: The Thymio Open-Source Hardware Robot, in *IEEE Robotics & Automation Magazine*, vol. 24, no. 1, pp. 77-85, March 2017. doi: 10.1109/MRA.2016.2636372
- [Mostafa, 2014] Mostafa, R., Hassan, R., Khadija Jahan, S.M., Chandan, S. & Ali, F., "Proposed system design for deployment of an e-learning system in rural Bangladesh," in *Electrical Engineering and Information &*

Communication Technology (ICEEICT), 2014 International Conference on, vol., no., pp.1-5, 10-12 April 2014. doi: 10.1109/ICEEICT.2014.6919095

- [Mumtaz, 2017] Mumtaz, S., Alshaily, A., Pang, Z., Rayes, A., Tsang, K. F. & Rodriguez, J., "Massive Internet of Things for Industrial Applications: Addressing Wireless IIoT Connectivity Challenges and Ecosystem Fragmentation," in IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 28-33, March 2017.. doi: 0.1109/MIE.2016.2618724
- [Muntone, 2013] Muntone, S., "Second Industrial Revolution". Education.com. The McGraw-Hill Companies. Retrieved 14 octubre 2013.
- [myRIO, 2018] NI myRIO. [Online]. Disponible: <http://www.ni.com/myrio/what-is/>, visitada el 6 de julio de 2018
- [Nastic, 2013] Nastic, S., Sehic, S., Vogler, M., Truong, H.L. & Dustdar, S., "PatRICIA -- A Novel Programming Model for IoT Applications on Cloud Platforms," in Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on, vol., no., pp.53-60, 16-18 Dec. 2013. doi: 10.1109/SOCA.2013.48
- [Natanael, 2017] Natanael Gallo, W., Heimfarth, T., Diego Ferreira, D. & Martins Mendes, T., "Real-time Detection System of Electrical Disturbances for Remote Communication Stations and Smart Grid," in IEEE Latin America Transactions, vol. 15, no. 10, pp. 1894-1900, Oct. 2017. doi: 10.1109/TLA.2017.8071233
- [Navarro, 2013] Navarro, J., Zaballos, A., Sancho-Asensio, A., Ravera, G. & Armendáriz-Lñigo, J. E., "The Information System of INTEGRIS: INTElligent Electrical GRId Sensor Communications," in IEEE Transactions on Industrial Informatics, vol. 9, no. 3, pp. 1548-1560, Aug. 2013. doi: 10.1109/TII.2012.2228869
- [Newberry, 2015] Newberry, S. & Gupta, G.S., "Wireless sensor based home automation as an educational springboard," in Sensors Applications Symposium (SAS), 2015 IEEE, vol., no., pp.1-6, 13-15 April 2015. doi: 10.1109/SAS.2015.7133610
- [Niemelä, 2017] Niemelä, P., "All rosy in scratch lessons: No bugs but guts with visual programming," 2017 IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, 2017, pp. 1-9. doi: 10.1109/FIE.2017.8190612
- [Nikou, 2014] Nikou, S. A. & Economides, A. A., "Transition in student motivation during a scratch and an app inventor course," 2014 IEEE Global Engineering Education Conference (EDUCON), Istanbul, 2014, pp. 1042-1045. doi: 10.1109/EDUCON.2014.6826234
- [NodeMcu, 2018a] NodeMcu ESP8266 [Online]. Disponible en: <http://www.nodemcu.com/>, visitada el 6 de julio de 2018
- [NodeMCU, 2018b] NodeMCU v2 - Lua based ESP8266 shop [Online]. Disponible en: <http://www.exp-tech.de/nodemcu-v2-lua-based-esp8266>, visitada el 6 de julio de 2018

- [NodeMCU, 2018c] NodeMCU forum [Online]. Disponible en: <https://www.hackster.io/nodemcu>, visitada el 6 de julio de 2018
- [Ogu, 2017] Ogu, R. E. & Chukwudebe, G. A., "Development of a cost-effective electricity theft detection and prevention system based on IoT technology," 2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON), Owerri, 2017, pp. 756-760. doi: 10.1109/NIGERCON.2017.8281943
- [Onion, 2018a] Onion Omega [Online]. Disponible en: <https://wiki.onion.io/Get-Started>, visitada el 6 de julio de 2018
- [Onion, 2018b] Onion Omega shop [Online]. Disponible en: <https://onion.io/store/>, visitada el 6 de julio de 2018
- [Onion, 2018c] Onion community [Online]. Disponible en: <https://community.onion.io/>, visited on 9 October 2017.
- [Orduña, 2014] Orduña, P., "Transitive and scalable federation model for remote laboratories". Tesis doctoral. Facultad de Ingeniería, Universidad de Deusto. Bilbao, 2013. Disponible en http://www.weblab.deusto.es/pub/dissertation_pablo.pdf. Accesible el 15 de abril de 2014.
- [Orha, 2014] Orha, I. & Oniga, S., "Study regarding the optimal sensors placement on the body for human activity recognition," Design and Technology in Electronic Packaging (SIITME), 2014 IEEE 20th International Symposium for, Bucharest, 2014, pp. 203-206. doi: 10.1109/SIITME.2014.6967028
- [Papadimitriou, 2007] Papadimitriou, V. & Papadopoulos, E., "Putting low-cost commercial robotics components to the test - Development of an educational mechatronics/robotics platform using LEGO components". IEEE Robotics & Automation Magazine, vol. 14, no. 3, pp. 99-110, Sept. 2007. doi: 10.1109/MRA.2007.901322
- [Papavlasopoulou, 2018] Papavlasopoulou, S., Giannakos, M. N. & Jaccheri, L., "Discovering children's competences in coding through the analysis of Scratch projects," 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, pp. 1127-1133. doi: 10.1109/EDUCON.2018.8363356
- [Parkhomenko, 2018] Parkhomenko, A., Parkhomenko, A., Tabunshchyk, G., Henke, K. & Wuttke, H. D., "The remote labs as an effective tool of inclusive engineering education," 2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, 2018, pp. 209-214.
- [Particle, 2018a] Particle Photon [Online]. Disponible en: <https://www.particle.io/products/hardware/photon-wifi-dev-kit>, visitada el 6 de julio de 2018
- [Particle, 2018b] Particle Photon shop [Online]. Disponible en: <https://store.particle.io/collections/photon>, visitada el 6 de julio de 2018
- [Particle, 2018c] Particle community [Online]. Disponible en: <https://community.particle.io/>, visitada el 6 de julio de 2018

- [Pastor, 2006] Pastor, R., "Especificación formal de laboratorios virtuales y remotos: aplicación a la ingeniería de control". Tesis doctoral. Escuela Técnica Superior de Ingeniería Informática Universidad Nacional de Educación a Distancia (UNED). Madrid, 2006.
- [Pattichis, 2017] Pattichis, M. S., Celedon-Pattichis, S. & LopezLeiva, C., "Teaching image and video processing using middle-school mathematics and the Raspberry Pi," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 6349-6353. doi: 10.1109/ICASSP.2017.7953378
- [Paull, 2017] Paull, L. et al. (2017), Duckietown: An open, inexpensive and flexible platform for autonomy education and research, 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 1497-1504. doi: 10.1109/ICRA.2017.7989179
- [Peixoto, 2018a] A. Peixoto et al., "Diversity and inclusion in engineering education: Looking through the gender question," 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, pp. 2071-2075. doi: 10.1109/EDUCON.2018.8363494
- [Peixoto, 2018b] A. Peixoto et al., "Robotics tips and tricks for inclusion and integration of students," 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, pp. 2037-2041. doi: 10.1109/EDUCON.2018.8363487
- [Peppler, 2013] Peppler, K., "STEAM-Powered Computing Education: Using E-Textiles to Integrate the Arts and STEM," in *Computer*, vol. 46, no. 9, pp. 38-43, September 2013. doi: 10.1109/MC.2013.257
- [Perini, 2017] Perini, S., Margoudi, M., Oliveira, M. F. & Taisch, M., "Increasing middle school students' awareness and interest in manufacturing through digital game-based learning (DGBL)". *Comput Appl Eng Educ.* 2017;25:785-799. doi: 10.1002/cae.21836
- [Perva, 2016] Perva, C., Perva, I., Rusu, D. D., Andreescu, N. & Puiu, M., "Web based application for improving the education quality of young medical genetics healthcare professionals," 2017 E-Health and Bioengineering Conference (EHB), Sinaia, 2017, pp. 161-164. doi: 10.1109/EHB.2017.7995386
- [Petry, 2016] Petry, C. A., Pacheco, F. S., Lohmann, D., Correa, G. A. & Moura, P., "Project teaching beyond Physics: Integrating Arduino to the laboratory," 2016 Technologies Applied to Electronics Teaching (TAEE), Seville, 2016, pp. 1-6. doi: 10.1109/TAEE.2016.7528376
- [Phanomchoeng, 2016] Phanomchoeng, G., Chanchaen, R., Kwanmuang, S. & Maneeratana, K., "Successive build up lab for learning mechatronics," 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Bangkok, 2016, pp. 93-100. doi: 10.1109/TALE.2016.7851777
- [Pickering, 2016] Pickering, T. A., Yuen, T. T., and Wang, T. "STEM conversations in social media: Implications on STEM education", 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engi-

neering (TALE), Bangkok, 2016, pp. 296-302. doi: 10.1109/TALE.2016.7851810

- [Pinto, 2012] Pinto, M., Moreira, A. P. & Matos, A., "Localization of Mobile Robots Using an Extended Kalman Filter in a LEGO NXT". IEEE Transactions on Education, vol. 55, no. 1, pp. 135-144, Feb. 2012. doi: 10.1109/TE.2011.2155066
- [PIXEL, 2018] Academia PIXEL [Online]. Disponible: <https://www.academiapixel.com/>, visitada el 6 de julio de 2018
- [Plaza Robotica, 2018a] Plaza Robotica. [Online]. Disponible en: <http://www.plazarobotica.es/>, visitada el 6 de julio de 2018
- [Plaza Robotica, 2018b] Página de Facebook de Plaza Robotica. [Online]. Disponible en: <https://www.facebook.com/plazarobotica.es/>, visitada el 6 de julio de 2018
- [Plaza Robotica, 2018c] Página de Twitter de Plaza Robotica. [Online]. Disponible en: <https://twitter.com/plazarobotica>, visitada el 6 de julio de 2018
- [Plaza Robotica, 2018d] Página de Instagram de Plaza Robotica. [Online]. Disponible en: <https://www.instagram.com/wei3h/?hl=es>, visitada el 6 de julio de 2018
- [Plaza, 2016a] Plaza, P., Sancristobal, E., Carro, G. & Castro, M., "A Wireless robotic educational platform approach". 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), Madrid, Spain, 2016, pp. 145-152. doi: 10.1109/REV.2016.7444455
- [Plaza, 2016b] Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Pérez, C., "Collaborative robotic educational tool based on programmable logic and Arduino," 2016 Technologies Applied to Electronics Teaching (TAEE), Seville, 2016, pp. 1-8. doi: 10.1109/TAEE.2016.7528380
- [Plaza, 2016c] Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Pérez, C., "Herramienta Colaborativa Robótica Educativa basada en Lógica Programable y Arduino," Libro de actas del TAEE 2016. Disponible en: <http://www.taee2016.org/index.php/es/actas>, visitada el 6 de julio de 2018
- [Plaza, 2016d] Plaza, P., Sancristobal, E., Carro, G. & Castro, M., "Robotic Educational Tool to engage students on Engineering," 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 2016, pp. 1-4. doi: 10.1109/FIE.2016.7757417
- [Plaza, 2017a] Plaza, P., Sancristobal, E., Carro, G. & Castro, M., "Home-made robotic education, a new way to explore". 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp.132-136. doi: 10.1109/EDUCON.2017.7942837
- [Plaza, 2017a] Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Ruiz, E., "Wireless Development Boards to Connect the World," In: Auer M., Zutin D. (eds) Online Engineering & Internet of Things. Lecture Notes in Networks and Systems, vol 22. Springer, Cham. doi: 10.1007/978-3-319-64352-6_2

- [Plaza, 2017b] Plaza, P., Sancristobal, E., Carro, G. & Castro, M., "Home-made robotic education, a new way to explore". 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp.132-136. doi: 10.1109/EDUCON.2017.7942837
- [Plaza, 2017c] Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Blázquez, M., "From 4 Wheeled Remote Robot to Serious Collaborative Remote Laboratory". *Int Rob Auto J.* 2017. 3(4): 00065. doi: 10.15406/iratj.2017.03.00065
- [Plaza, 2017d] Plaza, P., Sancristobal, E., Carro, G., Castro, M., Blázquez, M., Muñoz, J. & Álvarez, M., "Scratch as Educational Tool to Introduce Robotics". In: Auer M., Guralnick D., Simonics I. (eds) *Teaching and Learning in a Digital World. ICL 2017. Advances in Intelligent Systems and Computing*, vol 715. Springer, Cham, 2017 doi: 10.1007/978-3-319-73210-7_1
- [Plaza, 2018a] Plaza, P., Sancristobal, E., Carro, G., Castro, M., Blazquez, M. & Peixoto, A., "Traffic lights through multiple robotic educational tools," 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, pp. 2015-2020. doi: 10.1109/EDUCON.2018.8363483
- [Plaza, 2018b] Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Blazquez, M., "Scratch day to introduce robotics," 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, 2018, pp. 208-216. doi: 10.1109/EDUCON.2018.8363230
- [Plaza, 2018c] Plaza P., Sancristobal E., Carro G., Garcia-Loro F., Blazquez M. & Castro M., "European Robotics Week to introduce robotics and promote engineering," *Comput Appl Eng Educ.* 2018;1–13. doi: 10.1002/cae.21966
- [Plaza, 2018d] Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blazquez, M., Garcia-Loro F. & Muñoz, J., "Crumble como herramienta educativa para introducir en robótica," *Libro de actas del TAAE 2018*. Disponible en: <http://www.taee2016.org/index.php/es/actas>, visitada el 6 de julio de 2018
- [Plaza, 2018e] Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blazquez, M., Garcia-Loro F., "Iluminando a través de la robótica educativa," *Libro de actas del TAAE 2018*. Disponible en: <http://www.taee2016.org/index.php/es/actas>, visitada el 6 de julio de 2018
- [Plaza, 2018f] Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blazquez, M., Garcia-Loro F. & Muñoz, J., "Crumble as an educational tool to introduce robotics," 2018 *Technologies Applied to Electronics Teaching (TAAE)*, Tenerife, 2018, pendiente de publicar.
- [Plaza, 2018g] Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blazquez, M. & Garcia-Loro F., "Lighting through educational robotics," 2018 *Technologies Applied to Electronics Teaching (TAAE)*, Tenerife, 2018, pendiente de publicar.

- [Plaza, 2018h] Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blazquez, M. & Garcia-Loro F., "Crumble Day to introduce robotics," 2018 IEEE Frontiers in Education Conference (FIE), San Jose, California, USA, 2018, pendiente de publicar.
- [Plaza, 2018i] Plaza, P., Sancristobal, E., Carro, G., Castro, M. Blazquez, M. & Garcia-Loro F., "Multiplatform educational robotics course to introduce children in robotics," 2018 IEEE Frontiers in Education Conference (FIE), San Jose, California, USA, 2018, pendiente de publicar.
- [Plaza, 2018j] Plaza, P., Sancristobal, E., Carro, G., Blazquez, M., Menacho, A., Castro, M., Perez, C. & Garcia-Loro F., "Local MOOC Solution For Tight Budgets Or Limited Internet Access," LWMOOCs V Learning with MOOCs 2018 , Madrid, Spain, 2018, pendiente de publicar.
- [Plaza, 2018k] Plaza, P., Sancristobal, E., Carro, G., Blazquez, M., Garcia-Loro F., Martin, S., Perez, C., & Castro, M., "Arduino as educational tool to introduce robotics," 7th International Conference on Teaching, Assessment and Learning for Engineering (TALE) 2018, Wollongong, NSW, Australia, 2018.
- [Plaza, 2018l] Plaza, P., Sancristobal, E., Carro, G., Castro, M. & Ruiz, E., "Internet of Things development boards for Robotic Education," IEEE Consumer Electronics Magazine, en proceso de revisión.
- [Plaza, 2018m] Plaza, P., Sancristobal, E., Carro, G., Blazquez, M., Garcia-Loro F. & Castro, M., "Combining Programmable Logic and Arduino for Robotic STEM Education," Journal of Science Education and Technology, en proceso de revisión.
- [PMBOK®, 2018] PMBOK®. [Online]. Disponible en: <https://www.pmi.org/pmbok-guide-standards>, visitada el 6 de julio de 2018
- [Prechelt, 2013] Prechelt, L., "Agile offshoring: Using pair work to overcome nearshoring difficulties," 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), San Francisco, CA, 2013, pp. 157-158.. doi: 10.1109/CHASE.2013.6614755
- [Premkumar, 2015] Premkumar, K. & Nigel, K. G. J., "Smart phone based robotic arm control using raspberry pi, android and Wi-Fi," in Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on , vol., no., pp.1-3, 19-20 March 2015
- [Pruet, 2015] Pruet, P., Farzin, D., Ang, C.S. & Chaiwut, N., "Exploring the Internet of "Educational Things" (IoET) in rural underprivileged areas," in Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015 12th International Conference on, vol., no., pp.1-5, 24-27 June 2015. doi: 10.1109/ECTICon.2015.7207125

- [Pumipuntu, 2008] Pumipuntu, S. & Phromchanthuek, S., ""Collaborative Learning", the Learning Method through Internet," in Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on, vol.1, no., pp.3-7, 11-13 Nov. 2008. doi: 10.1109/ICCIT.2008.293
- [Puritty, 2017] Puritty, C., Strickland, L. R., Alia, E., Blonder, B., Klein, E., Kohl, M. T., McGee, E., Quintana, M., Ridley, R. E., Tellman, B. & Gerber, L. R., "Without inclusion, diversity initiatives may not be enough". Science, 357(6356), pp.1101-1102. doi: 10.1126/science.aai9054
- [PWC, 2018] PWC Press Room [Online]. Disponible: <http://press.pwc.com/>, visitada el 6 de julio de 2018
- [Quartus II, 2018a] Página web de Quartus II. Disponible en: <http://dl.altera.com/?edition=lite>, visitada el 3 de agosto de 2017.
- [Quartus II, 2018b] Quartus II Handbook Version 13.0.0, Volume 1: Design and Synthesis. Altera Corporation. May 2013.
- [Quartus II, 2018c] Quartus II Handbook Version 13.0.0, Volume 2: Design Implementation and Optimization. Altera Corporation. May 2013.
- [Quartus II, 2018d] Quartus II Handbook Version 13.0.0, Volume 3: Verification. Altera Corporation. May 2013.
- [Rama IEEE, 2018] Página de Facebook de la Rama de Estudiantes del IEEE de la UNED. [Online]. Disponible en: <https://www.facebook.com/IEEEsbUNED/>, visitada el 6 de julio de 2018
- [Raspberry Pi, 2018] Página Web de Raspberry Pi. [Online]. Disponible en: <https://www.raspberrypi.org>, visitada el 6 de julio de 2018
- [Raspberry, 2018] Raspberry Pi. [Online]. Disponible en: <https://www.raspberrypi.org>, visitada el 6 de julio de 2018
- [Raspberry, 2018] Raspberry Pi. [Online]. Disponible en: <https://www.raspberrypi.org>, visitada el 6 de julio de 2018
- [Raspberry, 2018] Raspberry Pi. [Online]. Disponible en: <https://www.raspberrypi.org>, visitada el 6 de julio de 2018
- [Raspberry, 2018] Raspberry Pi. [Online]. Disponible en: <https://www.raspberrypi.org>, visitada el 6 de julio de 2018
- [Rational, 1998] Rational Software White Paper, "Rational Unified Process. Best Practices for Software Development Teams" TP026B, Rev 11/01. 1998.
- [Reasonwhy, 2018] Reasonwhy [Online]. Disponible en: <https://www.reasonwhy.es/actualidad/sector/sueldos-salarios-marketing-espana-2017-02-27>, visitada el 6 de julio de 2018
- [Redfern, 2018] Redfern electronics [Online]. Disponible en: <http://redfernelectronics.co.uk/>, visitada el 6 de julio de 2018
- [Redondi, 2014] Redondi, A., Buranapanichkit, D., Cesana, M., Tagliasacchi, M. & Andreopoulos, Y., "Energy Consumption of Visual Sensor Networks: Impact of Spatio-Temporal Coverage," Circuits and Systems for Video Technology, IEEE Transactions on , vol.24, no.12, pp.2117,2131, Dec. 2014. doi: 10.1109/TCSVT.2014.2329378

- [Repositorio, 2018] Educational-Robotic-Platform en Github. [Online]. Disponible en: <https://github.com/Vertick/Educational-Robotic-Platform.git>, visitada el 6 de julio de 2018
- [Reutilización, 2018] Reutilización de clases. [Online]. Disponible en: <http://piensaenjavadesdecero.blogspot.com/2011/06/reutilizacion-de-clases.html>, visitada el 6 de julio de 2018
- [REV 2016, 2018] REV 2016. [Online]. Disponible en: <http://rev-conference.org/REV2016/>, visitada el 6 de julio de 2018
- [REV 2017, 2018] REV 2017. [Online]. Disponible en: <http://www.rev-conference.org/REV2017/>, visitada el 6 de julio de 2018
- [REV 2018, 2018] REV 2018. [Online]. Disponible en: <http://www.rev-conference.org/REV2018/>, visitada el 6 de julio de 2018
- [Rifkin, 2008] Rifkin, J., "The third industrial revolution," in *Engineering & Technology*, vol. 3, no. 7, pp. 26-27, April 26 2008. doi: 10.1049/et:20080718
- [Rising, 2000] Rising, L. & Janoff, N. S., "The Scrum Software Development Process for Small Teams," *Focus process diversity, IEEE SOFTWARE* July / August 2000
- [Rivera, 2017] Rivera, L. F. Z., Larrondo-Petrie, M. M. & Ribeiro Da Silva, L., "Implementation of cloud-based smart adaptive remote laboratories for education," 2017 IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, 2017, pp. 1-5. doi: 10.1109/FIE.2017.8190473
- [Rodríguez, 2014] Rodríguez, R., Blázquez, M., López, B., Castro, M., Sancristobal, E. & Martín, S. (2014), Educational games for improving the teaching-learning process of a CLIL subject: Physics and chemistry in secondary education, 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, 2014, pp. 1-8. doi: 10.1109/FIE.2014.7044064
- [Rodriguez-Diaz, 2016] Rodriguez-Diaz, E., Vasquez, J. C. & Guerrero, J. M., "Intelligent DC Homes in Future Sustainable Energy Systems: When efficiency and intelligence work together," in *IEEE Consumer Electronics Magazine*, vol. 5, no. 1, pp. 74-80, Jan. 2016.. doi: 10.1109/MCE.2015.2484699
- [Rojas, 2014] Rojas, D., Kapralos, B. & Dubrowski, A., "Gamification for Internet Based Learning in Health Professions Education," in *Advanced Learning Technologies (ICALT)*, 2014 IEEE 14th International Conference on, vol., no., pp.281-282, 7-10 July 2014. doi: 10.1109/ICALT.2014.87
- [Ross, 2017] Ross, R., Whittington, J. & Huynh, P., "LaserTag for STEM Engagement and Education," in *IEEE Access*, vol. 5, pp. 19305-19310, 2017. doi: 10.1109/ACCESS.2017.2753218
- [Ruberg, 2015] Ruberg, P., Guitar, A. & Ellervee, P., "Flexible controller for educational robot kit," in *Microelectronics Systems Education (MSE)*, 2015 IEEE International Conference on, vol., no., pp.17-20, 20-21 May 2015. doi: 10.1109/MSE.2015.7160007

- [Sáenz, 2015] Sáenz, J., Chacon, J., De La Torre, L., Visioli, A. & Dormido, S., "Open and Low-Cost Virtual and Remote Labs on Control Engineering," Access, IEEE, vol.3, no., pp.805,814, 2015. doi: 10.1109/ACCESS.2015.2442613
- [Safety, 2010] Functional safety of electrical/electronic/programmable electronic safety-related systems: Part 3 Software requirements, IEC Standard ISO/IEC 61508, 2010
- [Saleh, 2015] Saleh, N., Sharawi, A. A., Elwahed, M. A., Petti, A., Puppato, D. & Balestra, G., "Preventive Maintenance Prioritization Index of Medical Equipment Using Quality Function Deployment," in IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 3, pp. 1029-1035, May 2015. doi: 10.1109/JBHI.2014.2337895
- [San Cristobal, 2010] San Cristobal, E., "Metodología, estructura y desarrollo de interfaces intermedias para la conexión de laboratorios remotos y virtuales a plataformas educativas". Tesis doctoral. Escuela Técnica Superior de Ingenieros Industriales Universidad Nacional de Educación a Distancia (UNED). Madrid, 2010.
- [Saraladevi, 2015] Saraladevi, B. & Sedhumadhavan, S., "Video streaming in autonomous Mobile Robot using Wi-Fi," Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on, Coimbatore, 2015, pp. 1-5. doi: 10.1109/ICIIECS.2015.7192954
- [Satija, 2017] Satija, U., Ramkumar, B. & Sabarimalai Manikandan, M., "Real-Time Signal Quality-Aware ECG Telemetry System for IoT-Based Health Care Monitoring," in *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 815-823, June 2017. doi: 10.1109/JIOT.2017.2670022
- [SCE, 2018] Siemens Automation Cooperates with Education (SCE) [Online]. Disponible: <http://w3.siemens.com/mcms/sce/en/pages/default.aspx>, visitada el 6 de julio de 2018
- [Schilling, 2005] Schilling, K., Roth, H. & Spilca, C., "A tele-experiment on rover motor control via internet". *J. Robotic Syst.*, 22: 123-130. 2005. doi:10.1002/rob.20053
- [Schreuder, 2011] Schreuder, H. W. R., "Advances in training for laparoscopic and robotic surgery". Tesis doctoral. Utrecht University. Utrecht, Países Bajos, 2011.
- [Scratch, 2018a] Página web de Scratch version 1.4. [Online]. Disponible: https://scratch.mit.edu/scratch_1.4/, visitada el 6 de julio de 2018
- [Scratch, 2018b] Página web de Scratch version 2.0. [Online]. Disponible: <https://scratch.mit.edu/scratch2download/>, visitada el 6 de julio de 2018
- [Scratch, 2018c] Página web de Scratch resources. [Online]. Disponible: <https://scratch.mit.edu/explore/projects/all>, visitada el 6 de julio de 2018
- [Scratch, 2018d] Página web de Scratch for parents. [Online]. Disponible: <https://scratch.mit.edu/parents/>, visitada el 6 de julio de 2018

- [Scratch, 2018e] Página web de Scratch for educators. [Online]. Disponible: <https://scratch.mit.edu/educators/>, visitada el 6 de julio de 2018
- [Scratch, 2018f] Scratch Day [Online]. Disponible: <https://day.scratch.mit.edu/>, visitada el 6 de julio de 2018
- [ScratchEd, 2018a] Página web de ScratchEd community. [Online]. Disponible: <http://scratched.gse.harvard.edu/>, visitada el 6 de julio de 2018
- [ScratchEd, 2018b] Página web de ScratchEd Meetups. [Online]. Disponible: <https://www.meetup.com/pro/scratched/>, visitada el 6 de julio de 2018
- [SEIDI, 2018] Página web de SEIDI [Online]. Disponible: <http://www.mineco.gob.es/portal/site/mineco/idi>, visitada el 6 de julio de 2018
- [Seiler, 2012] Seiler, S., "Laboratory as a Service – A Holistic Framework for Remote and Virtual Labs. Tesis doctoral. Faculty of Mechanical Engineering Department of Mechatronics", Tallinn University of Technology. Tallin, Estonia 2012. ISBN 9789949233601, ISBN 9789949233618, ISSN 14064758.
- [Settembre, 2012] Settembre, M., "Towards a hyper-connected world," 2012 15th International Telecommunications Network Strategy and Planning Symposium (NETWORKS), Rome, 2012, pp. 1-5 doi: 10.1109/NETWKS.2012.6381667
- [Shapiro, 2017] Shapiro, J., "Your Five Year Old Can Learn To Code With An iPad App". Forbes. doi: 10.3389/fpsyg.2017.00660
- [Shen, 2014] Shen, Q., Liang, X., Shen, X., Lin, X. & Luo, H. Y., "Exploiting Geo-Distributed Clouds for a E-Health Monitoring System With Minimum Service Delay and Privacy Preservation," in IEEE Journal of Biomedical and Health Informatics, vol. 18, no. 2, pp. 430-439, March 2014.. doi: 10.1109/JBHI.2013.2292829
- [Shim, 2016] Shim, J., Kwon, D. & Lee, W. (2016), The Effects of a Robot Game Environment on Computer Programming Education for Elementary School Students, in IEEE Transactions on Education, vol. PP, no.99, pp.1-9, Nov. 2016. doi: 10.1109/TE.2016.2622227
- [Siemens, 2018] Siemens Vision 2020 [Online]. Disponible: <https://www.siemens.com/about/en/strategic-overview.htm>, visitada el 6 de julio de 2018
- [Sjoberg, 2017] Sjoberg, K., Andres, P., Buburuzan, T. & Brakemeier, A., "Cooperative Intelligent Transport Systems in Europe: Current Deployment Status and Outlook," in IEEE Vehicular Technology Magazine, vol. 12, no. 2, pp. 89-97, June 2017.. doi: 10.1109/MVT.2017.2670018
- [Škraba, 2016] Škraba, A., Koložvari, A., Kofjač, D., Stojanović, R., Stanovov, V. & Semenkin, E., "Streaming pulse data to the cloud with bluetooth LE or NODEMCU ESP8266," 2016 5th Mediterranean Conference on Embedded Computing (MECO), Bar, 2016, pp. 428-431. doi: 10.1109/MECO.2016.7525798
- [Skype, 2018] Skype [Online]. Disponible: <https://www.skype.com/en/>, visitada el 6 de julio de 2018

- [Sochacka, 2016] Sochacka, N. W., Guyotte, Kelly. W. & Walther, J. (2016), Learning Together: A Collaborative Autoethnographic Exploration of STEAM (STEM + the Arts) Education. *Journal of Engineering Education*, 105: 15–42. doi: 10.1002/jee.20112
- [Sripakagorn, 2014] Sripakagorn, A., Chanchaen, R., Maneeratana, K. & Panyajirakul, K., "An implementation of CDIO/design thinking in mechatronics projects," 2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), Wellington, 2014, pp. 516-521. doi: 10.1109/TALE.2014.7062627
- [ST, 2018] St Microelectronics Development Tools. [Online]. Disponible en: <https://www.st.com/en/evaluation-tools/stm32-mcu-eval-tools.html?querycriteria=productId=SS1532>, visitada el 6 de julio de 2018
- [Stair, 2011] Stair, R., Stair, M., Rohan, F., Rohan, G. & Reynolds, G. (2011) "Principles of information systems", 2nd ed. Cengage Learning Australia, Melbourne, Australia. ISBN 9780170188456
- [Stankovic, 2014] Stankovic, J. A., "Research Directions for the Internet of Things," in *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3-9, Feb. 2014. doi: 10.1109/JIOT.2014.2312291
- [Stubbs, 2009] Stubbs, K. N. & Yanco, H. A., "STREAM: A workshop on the use of Robotics in K–12 STEM education [Education]," in *IEEE Robotics & Automation Magazine*, vol. 16, no. 4, pp. 17-19, December 2009. doi: 10.1109/MRA.2009.934830
- [Susilo, 2016] Susilo, E. et al. (2016), STORMLab for STEM Education: An Affordable Modular Robotic Kit for Integrated Science, Technology, Engineering, and Math Education, in *IEEE Robotics & Automation Magazine*, vol. 23, no. 2, pp. 47-55, June 2016. doi: 10.1109/MRA.2016.2546703
- [Tabunshchyk, 2018] Tabunshchyk, G., Parkhomenko, A., Morshchavka, S. & Luengo, D., "Engineering education for HealthCare purposes: A Ukrainian perspective," 2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, 2018, pp. 245-249. doi: 10.1109/MEMSTECH.2018.8365743
- [TAAE 2016, 2018] TAAE 2016. [Online]. Disponible en: <http://www.taee2016.org/>, visitada el 6 de julio de 2018
- [TAAE 2018, 2018] TAAE 2018. [Online]. Disponible en: <http://taee2018.org/>, visitada el 6 de julio de 2018
- [TALE 2018, 2018] TALE 2018. [Online]. Disponible en: <http://tale-conference.org/>, visitada el 6 de julio de 2018
- [Tang, 2010] Tang, H. & Jiao, L., "Analysis on the application of sketchup software in architectural design teaching", *The 2nd International Conference on Information Science and Engineering*, Hangzhou, China, 2010, pp. 5683-5686. doi: 10.1109/ICISE.2010.5689844

- [Tawfik, 2013] Tawfik, M., Monteso, S., Garcia-Loro, F., Sancristobal, E., Mur, F., Diaz, G. & Castro, M., "Design of electronics circuits practices for an online master degree program using VISIR," in Global Engineering Education Conference (EDUCON), 2013 IEEE, vol., no., pp.1222-1227, 13-15 March 2013. doi: 10.1109/EduCon.2013.6530262
- [Tedder, 2004] Tedder, M., Chamulak, D., Chen, L., Nair, S., Shvartsman, A., Tseng, I. & Chung, C., "An affordable modular mobile robotic platform with fuzzy logic control and evolutionary artificial neural networks". J. Robotic Syst., 21: 419-428. 2004. doi:10.1002/rob.20023
- [Telefonica, 2018] Telefonica Press Office [Online]. Disponible: <https://www.telefonica.com/en/web/press-office>, visitada el 6 de julio de 2018
- [Teoría, 2018] Teoría de la Programación Orientada a Objetos. [Online]. Disponible en: <https://desarrolloweb.com/manuales/teoria-programacion-orientada-objetos.html>, visitada el 6 de julio de 2018
- [TerasIC, 2018] TerasIC. [Online]. Disponible en: <http://www.terasic.com.tw/en>, visitada el 6 de julio de 2018
- [Texas, 2018] Texas Instruments Development Tools. [Online]. Disponible en: <http://www.ti.com/tools-software/hw-kits-boards.html#>, visitada el 6 de julio de 2018
- [Thaker, 2016] Thaker, T., "ESP8266 based implementation of wireless sensor network with Linux based web-server," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, 2016, pp. 1-5. doi: 10.1109/CDAN.2016.7570919
- [Tien, 2012] Tien, J. M., "Towards the next industrial revolution," 2012 4th IEEE International Symposium on Logistics and Industrial Informatics, Smolenice, 2012, pp. 13-14. doi: 10.1109/LINDI.2012.6319480
- [Torre, 2016] Torre, I. & Torsani, S., "A Recommender System as a Support and Training Tool," 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Naples, 2016, pp. 773-780. doi: 10.1109/SITIS.2016.127
- [Trevino, 2008] Trevino, S. A. & Anantatmula, V. S., "Capitalizing from past projects: the value of lessons learned". Paper presented at PMI® Research Conference: Defining the Future of Project Management, Warsaw, Poland. Newtown Square, PA: Project Management Institute. Disponible en: <https://www.pmi.org/learning/library/business-benefits-value-lessons-learned-7116>, visitada el 6 de julio de 2018
- [Trowbridge, 2017] Trowbridge, B., Calas, F., Weingarten, Z., Husinger, J., Fomitchev, V. & Integlia, R., "Protomesh, a wireless solution and platform for embedded education and Raspberry Pi workshops," 2017 IEEE World Engineering Education Conference (EDUNINE), Santos, 2017, pp. 90-94. doi: 10.1109/EDUNINE.2017.7918189

- [Tsai, 2018] Tsai, F. H., "Designing a Computer Detective Game for Science Education," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 2018, pp. 797-798. doi: 10.1109/COMPSAC.2018.00125
- [Tsukamoto, 2016] Tsukamoto, H. et al., "Textual vs. visual programming languages in programming education for primary schoolchildren," 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 2016, pp. 1-7. doi: 10.1109/FIE.2016.7757571
- [Vaidya, 2017] Vaidya, B., Patel, A., Panchal, A., Mehta, R., Mehta, K. & Vaghasiya, P., "Smart home automation with a unique door monitoring system for old age people using Python, OpenCV, Android and Raspberry pi," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, 2017, pp. 82-86. doi: 10.1109/ICCONS.2017.8250582
- [Valdez, 2016] Valdez, M. T., Ferreira, C. M. & Barbosa, F. P. M., "3D virtual laboratory for teaching circuit theory — A virtual learning environment (VLE)," 2016 51st International Universities Power Engineering Conference (UPEC), Coimbra, 2016, pp. 1-4. doi: 10.1109/UPEC.2016.8114126
- [Velasquez, 2014] Velasquez, N. F. et al., "Novice Programmers Talking about Projects: What Automated Text Analysis Reveals about Online Scratch Users' Comments," 2014 47th Hawaii International Conference on System Sciences, Waikoloa, HI, 2014, pp. 1635-1644. doi: 10.1109/HICSS.2014.209
- [Volnei, 2014] Volnei A. Pedroni., "Circuit Design with VHDL". MIT Press, Cambridge, MA, USA, 2004.
- [Walker, 2008] Walker, L. W., "Learning lessons on lessons learned". Paper presented at PMI® Global Congress 2008—North America, Denver, CO. Newtown Square, PA: Project Management Institute. Disponible en: <https://www.pmi.org/learning/library/lessons-learned-project-lessons-6993>, visitada el 6 de julio de 2018
- [Wan, 2016] Wan, J., Yi, M., Li, D., Zhang, C., Wang, S. & Zhou, K., "Mobile Services for Customization Manufacturing Systems: An Example of Industry 4.0," in IEEE Access, vol. 4, no., pp. 8977-8986, 2016.. doi: 10.1109/ACCESS.2016.2631152
- [Wang, 2012] Wang, Y. M., Zhao, G. Z. & Yin, H. L., "A novel prototype architecture for equipment tele-control and simulation", Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, 2012, pp. 633-637. doi: 10.1109/WCICA.2012.6357956
- [Wang, 2017] Wang, Y., Hill, K. J. & Foley, E. C., "Computer programming with Python for industrial and systems engineers: Perspectives from an instructor and students". Computer Applications in Engineering Education. 2017;25:800–811. doi: 10.1002/cae.21837
- [Warin, 2016] Warin, B., Talbi, O., Kolski, C. & Hoogstoel, F., "Multi-Role Project (MRP): A New Project-Based Learning Method for STEM", in IEEE

Transactions on Education, vol. 59, no. 2, pp. 137-146, May 2016.
doi: 10.1109/TE.2015.2462809

- [Wei, 2016] Wei, L., "A Research of College English Self-Learning Based on Internet Supporting Factors," 2016 International Conference on Robots & Intelligent System (ICRIS), Zhangjiajie, 2016, pp. 43-46. doi: 10.1109/ICRIS.2016.9
- [WhatsApp, 2018] WhatsApp [Online]. Disponible: <https://www.whatsapp.com/>, visitada el 6 de julio de 2018
- [WIE Spain, 2018] Página de Facebook de IEEE Women in Engineering Spain. [Online]. Disponible en: <https://www.facebook.com/ieeeWiESpain/>, visitada el 6 de julio de 2018
- [Williams, 2017] Williams, K., "A Collaborative Learning Culture: Preparing K-12 Teachers for STEM Success [Pipelining: Attractive Programs for Women]," in IEEE Women in Engineering Magazine, vol. 11, no. 2, pp. 44-46, Dec. 2017. doi: 10.1109/MWIE.2017.2746365
- [Wollschlaeger, 2017] Wollschlaeger, M., Sauter, T. & Jasperneite, J., "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," in IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 17-27, March 2017.. doi: 10.1109/MIE.2017.2649104
- [Xie, 2010] Xie, X., Zheng, B. & Nan, X., "Notice of Retraction Digital design experiments of mechanical product based on fischertechnik model", 2010 2nd International Conference on Mechanical and Electronics Engineering, Kyoto, 2010, pp. V1-237-V1-240. doi: 10.1109/ICMEE.2010.5558558
- [Xie, 2017] Xie, G., Zeng, G., Li, R. & Li, K., "Energy-Aware Processor Merging Algorithms for Deadline Constrained Parallel Applications in Heterogeneous Cloud Computing," in IEEE Transactions on Sustainable Computing, vol. 2, no. 2, pp. 62-75, April-June 1 2017.. doi: 10.1109/TSUSC.2017.2705183
- [Xu, 2014] Xu, B., Xu, L. D., Cai, H., Xie, C., Hu, J. & Bu, F., "Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services," in Industrial Informatics, IEEE Transactions on , vol.10, no.2, pp.1578-1586, May 2014. doi: 10.1109/TII.2014.2306382
- [Yamanoor, 2017] Yamanoor, N. S. & Yamanoor, S., "High quality, low cost education with the Raspberry Pi," 2017 IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, 2017, pp. 1-5. doi: 10.1109/GHTC.2017.8239274
- [Yan, 2016] Yan, Z., Ding, W., Yu, X., Zhu, H. & Deng, R. H., "Deduplication on Encrypted Big Data in Cloud," in IEEE Transactions on Big Data, vol. 2, no. 2, pp. 138-150, June 1 2016.. doi: 10.1109/TBDDATA.2016.2587659

- [Yepez-Bonilla, 2017] Yepez-Bonilla, M., Martin-Gutierrez, S., Diaz-Orueta, G. & Castro-Gil, M., "RISK ANALYSIS AND RECOMMENDATIONS FOR ELECTRONIC DESIGN WITH ARDUINO". 2017. DYNA, 92(6). 607-608. doi: 10.6036/8587
- [Yoshihara, 2016] Yoshihara, K. & Watanabe, K., "Practice of Programming Education Using Scratch and NanoBoardAG for High School Students," 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), Fukuoka, 2016, pp. 567-568.
- [YP Spain, 2018] Página de Facebook de IEEE Young Professionals Spain. [Online]. Disponible en: <https://www.facebook.com/yppspain/>, visitada el 6 de julio de 2018
- [Zanella, 2014] Zanella, A., Bui, N., Castellani, A., Vangelista, L. & Zorzi, M., "Internet of Things for Smart Cities," in IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22-32, Feb. 2014. doi: 10.1109/JIOT.2014.2306328
- [Zhou, 2017] Zhou, S., Zhang, W., Zou, Y., et al. "Piezoelectric driven insect-inspired robot with flapping wings capable of skating on the water", Electronics Letters, vol. 53, no. 9, pp. 579-580, 4 27 2017. doi: 10.1049/el.2017.0186
- [Zieris, 2013] Zieris, F. & Salinger, S., "Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practices," 2013 IEEE 8th International Conference on Global Software Engineering, Bari, 2013, pp. 144-153. doi: 10.1109/ICGSE.2013.26
- [Zolyomi, 2017] Zolyomi, A., Ross, A. S., Bhattacharya, A., Milne, L. & Munson, S., "Value Sensitive Design for Neurodiverse Teams in Higher Education". In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (pp. 353-354). 2017. ACM. doi: 10.1145/3132525.3134787

ANEXO 1

ANEXO 1: CONTENIDO DEL DVD-ROM

Dada la gran cantidad de documentos y archivos que no pueden ser incluidos en formato físico se ha elaborado un DVD-ROM con el siguiente contenido:

Anexo 1.1. Diseño hardware

En la carpeta "PCB" se han incluido todos los archivos de diseño y fabricación relacionados con el diseño hardware:

- MAIN_MODULE_v1_0
- MAIN_MODULE_v1_1
- MAIN_MODULE_v1_2
- MAIN_MODULE_v1_3
- SECONDARY_MODULE_v1_0
- SECONDARY_MODULE_v1_1
- SECONDARY_MODULE_v1_2
- SECONDARY_MODULE_v1_3

Anexo 1.2. Diseño firmware

En la carpeta "FPGA" se han incluido todos los archivos de diseño, simulación e implementación relacionados con el diseño firmware:

- Firmware_v1:
 - FPGA_MAIN_MODULE
 - VHD:
 - 0_Packages
 - 1_FPGA_MAIN_MODULE:
 - Simulation
 - Source
 - Synthesis
 - 2_MAIN_PROCESSOR:
 - Simulation
 - Source

- Synthesis
- 3_UART_BLUETOOTH:
 - Simulation
 - Source
 - Synthesis
- 4_UART_SHIELD:
 - Simulation
 - Source
 - Synthesis
- 5_DISCRETE_COMM_SHIELD:
 - Simulation
 - Source
 - Synthesis
- 6_UART_ARDUINO:
 - Simulation
 - Source
 - Synthesis
- 7_PWM_MONITORING_ARDUINO:
 - Simulation
 - Source
 - Synthesis
- 8_DISCRETE_COMM_ARDUINO:
 - Simulation
 - Source
 - Synthesis
- 9_UART_RX_MACHINE:
 - Simulation
 - Source
 - Synthesis
- 10_UART_TX_MACHINE:
 - Simulation
 - Source
 - Synthesis
- Firmware_v2:
 - FPGA_MAIN_MODULE

- VHD:
 - 0_Packages
 - 1_FPGA_MAIN_MODULE:
 - Simulation
 - Source
 - Synthesis
 - 2_MAIN_PROCESSOR:
 - Simulation
 - Source
 - Synthesis
 - 3_UART_BLUETOOTH:
 - Simulation
 - Source
 - Synthesis
 - 4_UART_SHIELD:
 - Simulation
 - Source
 - Synthesis
 - 5_DISCRETE_COMM_SHIELD:
 - Simulation
 - Source
 - Synthesis
 - 6_UART_ARDUINO:
 - Simulation
 - Source
 - Synthesis
 - 7_PWM_MONITORING_ARDUINO:
 - Simulation
 - Source
 - Synthesis
 - 8_DISCRETE_COMM_ARDUINO:
 - Simulation
 - Source
 - Synthesis
 - 9_UART_RX_MACHINE:

- Simulation
- Source
- Synthesis
- 10_UART_TX_MACHINE:
 - Simulation
 - Source
 - Synthesis

Anexo 1.3. Diseño software

En la carpeta "Arduino" se han incluido todos los archivos de proyectos Arduino relacionados con el diseño software:

- Software_v1
- Integration_v1
- Configurator_HC06

Anexo 1.4. Material educativo

En la carpeta "Cursos y talleres" se han incluido todos los archivos relacionados con el material educativo:

- Cursos:
 - Curso anual de introducción a la robótica 2016_2017:
 - Scratch
 - Crumble
 - Arduino
 - Curso anual de introducción a la robótica 2017_2017:
 - Scratch básico
 - Scratch intermedio
 - Scratch avanzado
 - Crumble básico
- Talleres:
 - Scratch Day 2017
 - Crumble Day 2017
 - Robótica educativa multi-plataforma 2017

ANEXO 2

ANEXO 2: OTROS MÉRITOS

Durante todo el periodo de tiempo que ha ocupado esta investigación ha sido posible realizar una diversidad de actividades que, aunque no estaban directamente relacionadas con el núcleo de la investigación, bien han aportado en mayor o menor medida un impacto en el desarrollo de esta tesis doctoral. A lo largo de este capítulo se presentan dichas actividades.

Anexo 2.1. Colaboración en conferencias internacionales

Se colaboró en la conferencia FIE 2014 celebrada en Madrid (21-25 de octubre de 2014) con tareas relacionadas con la recepción e inscripción a la conferencia durante 50 horas.

Se ha contribuido en diversas conferencias internacionales en el proceso de revisión de diferentes artículos. La Tabla 166 contiene el nombre de la conferencia y el título del artículo para el año 2017.

Tabla 166. Revisiones en conferencias internacionales para el año 2017.

Conferencia	Título del artículo
REV 2017	How Game Design can enhance engineering higher education: focused IT study
REV 2017	Design of WEB Laboratory for Programming and Use of an FPGA device
EDUNINE 2017	Experiences using the Jigsaw learning technique to teach IFPUG function points
FIE 2017	Inclusion in Computing and Engineering Education: Perceptions and Learning in Diagram-Based e-Learning Classes with Blind and Sighted Learners
FIE 2017	Leveraging a Multi-Partner Approach to Develop Successful STEM Outreach Programs
ICL 2017	Study and development of a classroom management system - Application for programming language labs
ICL 2017	Interactive Games as Educative Strategy to Motivate Students to Communicate Inside the Classroom

La Tabla 167 contiene el nombre de la conferencia y el título del artículo para el año 2018.

Tabla 167. Revisiones en conferencias internacionales para el año 2018.

Conferencia	Título del artículo
REV 2018	An approach for supporting of navigation of blind people in public building based on hierarchical map ontology
REV 2018	REMLABNET and Virtual Reality interface
REV 2018	Online Experimenting with 3D LED cube
EDUCON 2018	Towards the Improvement of Children's ADHD through Augmented Reality Serious Games: Preliminary Results
EDUCON 2018	3D printing of microscope slides for visually impaired university students
FIE 2018	Teaching Fundamental Power Engineering Concepts using the VisPhasor App
FIE 2018	Informal STEM Experiences: Impact on Diverse Groups of Students and Teachers
FIE 2018	Work in Progress: Silicon Valley Women in Engineering Conference – Creating Community and Nurturing Engineering Identity
FIE 2018	From Explaining How Random Forest Classifier Predicts Learning of Software Engineering Teamwork to Guidance for Educators

Se contribuyó en la conferencia EDUCON 2018 celebrada en Tenerife (18-20 de abril de 2018) con la realización de los talleres “Driving lights through robotic educational” y “Inclusion and Integration of Students with Robotics”. También se moderó la sesión titulada “Learning Systems Platforms and Architectures”.

Se contribuyó en la conferencia TAEE 2018 celebrada en Tenerife (20-22 de junio de 2018) con la realización del taller “Lighting through educational robotics”. Además, se moderó la sesión titulada “Sesión III.A: Automática y Robótica 1”.

La Tabla 168 contiene el nombre de la conferencia y el título del artículo para el año 2019.

Tabla 168. Revisiones en conferencias internacionales para el año 2019.

Conferencia	Título del artículo
EDUCON 2019	How Can Computer Science Faculties Increase the Proportion of Women in Computer Science by Using Robots?
EDUCON 2019	Project Based Learning methodology to improve electrical efficiency in road lighting
EDUCON 2019	Engage Your Students via Physical Computing!
FIE 2019	Senioritis From the Student's Perspective

Conferencia	Título del artículo
FIE 2019	Student Progress in Learning Computer Programming: Insights from Association Analysis
FIE 2019	Towards the Identification of Profiles based on the Understanding of Programming Concepts: the Case of the Variable
FIE 2019	Balancing Theory and Practice in Software Engineering Education – A PBL, toolset based approach
FIE 2019	Mentoring in Lower-Level Engineering Courses
FIE 2019	A pilot study of a directed self-placement exam and a workshop designed to improve student learning outcomes in a junior level circuits and signals course
LWMOOCS 2019	Experience of Offering MOOC on Research in Technical Education for Teachers and Lessons Learnt
LWMOOCS 2019	Performance of Men and Women in Graded Team Assignments in MOOCs
LWMOOCS 2019	Analyzing Student Code Trajectories in an Introductory Programming MOOC

Se contribuyó en la conferencia EDUCON 2019 celebrada en Dubái (9-11 de abril de 2019) con la realización del taller “Driving lights through robotic educational”.

Anexo 2.2. Colaboración en revistas científicas internacionales

En el año 2016 se tuvo la oportunidad de ser parte del proceso de revisión del artículo “Open Modular Hardware and Software Kit for Creations of Remote Experiments Accessible from PC and Mobile Devices” publicado en la revista “International Journal of Online Engineering”.

Desde agosto de 2016 hasta la actualidad se ha tenido la oportunidad de ejercer como “Editorial Board Member” de la revista “International Journal of Automation and Robotic Technology”. Como parte del ejercicio del cargo se ha contribuido mediante la revisión de varias publicaciones tales como “Robust Active Vision Industrial CAD Parts Recognition System”, “Lenient computation in controlling the nonlinear system based on adaptive error optimisation in microgrid” y “Towards a Secure and Automated Platform for Fingerprint Based Electronic Voting Machine”.

En el año 2018 se tuvo la oportunidad de ser parte del proceso de revisión del artículo “A Digital Technology-based Introductory Biology Course Designed for Engineering and Other Non-life Science STEM Majors” publicado en la revista “Computer Applications in Engineering Education”.

En el año 2019 se tuvo la oportunidad de ser parte del proceso de revisión del artículo “FPGA Implementation of Error Control Codes in VHDL: An Undergraduate Research Project” publicado en la revista “Computer Applications in Engineering Education”. También se ha participado en la revisión de los artículos “Improvement in Research Quality of Higher Education in Universities: A Proposed Framework” y “STEM Education for Bangladeshi and Australian university students” que está en proceso de revisión en la revista “Computer Applications in Engineering Education”. Además, se ha participado en la revisión del artículo “Game Learning Analytics, Facilitating the Use of Serious Games in the Class” que está en proceso de revisión en la revista “IEEE Revista Iberoamericana de Tecnologías del Aprendizaje”.

Anexo 2.3. Cargos en la Rama de Estudiantes del IEEE de la UNED

Desde enero de 2016 hasta diciembre de 2017 se contribuyó con la figura de co-coordinador de la Rama de Estudiantes del IEEE de la UNED. Cabe mencionar los talleres de Scratch Day 2017 y Crumble 2017 como aportaciones más significativas.

Desde enero de 2018 hasta la fecha se contribuye como presidente y moderador de Mujer en la Ingeniería de la Rama de Estudiantes del IEEE de la UNED. Durante la conferencia EDUCON 2018 celebrada en Tenerife (18-20 de abril de 2018) se realizaron las charlas tituladas “Resources and materials for pre-university teachers” y “The Student Branch and related groups” como parte de la sesión organizada por el Capítulo Español de la Sociedad de Educación del IEEE.

Anexo 2.4. Colaboración con grupos de afinidad del IEEE

Mediante el despliegue de los talleres de robótica educativa multi-plataforma se trabajó en estrecha colaboración con los grupos de afinidad del IEEE Young Professionals Spain y Women in Engineering Spain.

Anexo 2.5. Promoción y divulgación mediante la impartición de cursos y talleres

Se ha contribuido en la promoción y difusión de la UNED, el IEEE, la Rama de Estudiantes del IEEE de la UNED y los grupos de afinidad del IEEE Young Professionals Spain y Women

in Engineering Spain a lo largo del despliegue de los talleres de robótica educativa multi-plataforma.