



Desarrollo de una aplicación basada en App Inventor para el manejo de bases de datos meteorológicas mediante técnicas API y scraping

Development of an application based on App Inventor for managing meteorological databases using API and scraping techniques

Juan Carlos Gómez-Sánchez y Francisco José Martínez-López

Universidad de Murcia, España

Resumen

Siendo evidente la preocupación social por la sostenibilidad medioambiental, los organismos europeos se han percatado de que, actuando en edades tempranas, se pueden paliar las consecuencias medioambientales a largo plazo y por ello se estableció en la 57 sesión de la ONU, al periodo 2004-2014 como la década en la que la educación sería la base de concienciación medioambiental. El presente artículo tiene por objetivo responder a los referentes marcados por la Orden ECD/65/2015 relativos al fomento de competencias transversales. Para ello, se planteará una propuesta académica en la materia de Tecnologías de la Información y Comunicación del primer curso de bachillerato en la que el alumnado deberá implementar una app basada en App Inventor con la finalidad de trabajar con variables climáticas de distintas bases de datos mediante técnicas basadas en interfaz de programación (API) y Web Scraping. Finalmente, se planteará una aplicación práctica con un microcontrolador y se comprobará la disparidad de resultados existente entre las bases de datos nacionales y las bases de datos internacionales debido al desfase de sincronización de sus sistemas de información.

Palabras clave: bases de datos; App Inventor; Arduino; API; Web Scraping.

Abstract

The social concern for environmental sustainability being evident, the European organizations have realized that, acting at an early age, the long-term environmental consequences can be alleviated and therefore the ONU established at the 57th session the 2004-2014 period as the decade in which education would be the basis of environmental awareness. The aim of this article is to respond to the Order ECD/65/2015 related to the promotion of transversal competences. For this reason, an academic proposal will be proposed in the subject of Information and Communication Technologies in the first high school course in which students must implement an app based on App Inventor in order to work with climatic variables of different databases using techniques based on programming interface (API) and Web Scraping. Finally, a practical application with a microcontroller will be considered and the disparity of results between national databases and international databases will be checked due to the synchronization lag of their information systems.

Keywords: Databases; App Inventor; Arduino; API; Web Scraping.

Fecha de recepción: 21/04/2021

Fecha de aceptación: 23/07/2021

Correspondencia: Francisco José Martínez López, Universidad de Murcia, España
Email: fjmartinez@um.es

Introducción

Según Freire (1995), la educación da lugar a una conciencia crítica e integral de nuestra situación en el planeta y que depende del grado de desarrollo económico del país. Sin embargo, para que esta conciencia pueda desarrollarse plenamente, la acción educativa ha de estar vinculada a la legislación vigente, las políticas y las medidas de control implementadas por los gobiernos (UNESCO, 2004). A nivel estatal, se puede mencionar que en la enseñanza secundaria el artículo 6 del Real Decreto 1105/2014 hace referencia a la necesidad de fomentar los elementos transversales en educación en valores.

Aunque se han llevado a cabo experiencias con alumnado de secundaria donde se ha podido demostrar que la evolución de sus perspectivas medioambientales no varía si no se utilizan tecnologías adecuadas (Jaén y Barbudo, 2010), lleva a pensar que, según García, Bassilotta y López (2014), es necesario fomentar el desarrollo de actividades centradas en el desarrollo de competencias garantizando unos escenarios de aprendizaje motivadores para el alumnado basándonos en las TIC.

Según Lozano et al. (2012), el uso de actividades novedosas que se alejen de las lecciones magistrales, despierta en el alumnado una notable curiosidad que consigue aumentar su atención en el desarrollo de tareas. Siendo conscientes de las numerosas alternativas existentes para introducir el control y la programación en el aula, se puede comprobar que las metodologías basadas en un lenguaje gráfico son aquellas preferidas por los docentes ya que permiten al alumnado abstraerse de los minuciosos detalles que componen la programación y su sintaxis. En consonancia a la elevada dificultad de la adquisición de conocimientos en programación, el *Massachusetts Institute of Technology* (MIT) desarrolló la aplicación App Inventor, la cual ha experimentado un gran auge debido a que se basa en la idea de programación por bloques orientados a objetos.

Si bien, la percepción de los docentes sobre el uso educativo de las TIC en el aula de tecnología muestra una tendencia positiva a su uso, ya que mejora los resultados del proceso de enseñanza-aprendizaje (Vallespín, 2014. Badia et al. 2013). Atendiendo a la aplicación de App Inventor en las aulas, se puede observar cómo docentes de la especialidad de matemáticas aplican en clase esta plataforma como recurso didáctico observando que un 60% del alumnado se siente cómodo en el uso de aplicaciones en el aula (Almaraz et al. 2015). En cuanto a la robótica en el aula, Vargas et al. (2015), indican que es un área muy atractiva debido a su carácter práctico, la cual se ha visto muy aceptada entre los jóvenes estudiantes de secundaria. En consecuencia, cabe destacar la presencia de numerosos centros que aplican las técnicas de gamificación y *flipped classroom* en el aula a la hora de utilizar App Inventor o la combinación de App Inventor junto con elementos robóticos, siendo el método preferido por muchos docentes a la hora de abordar los conocimientos del bloque de contenidos de robótica para así fomentar el aprendizaje constructivista (Cortés et al. 2009). En la Región de Murcia, se encuentra el proyecto de El Cable Amarillo, programa con el que se pretende fomentar el aprendizaje de la robótica en los centros docentes abasteciéndolos con kits electrónicos y cuya finalidad es potenciar los procesos cognitivos asociados al razonamiento lógico-matemático mediante la programación.

La clave del éxito para que el alumnado adquiera los conocimientos referentes al control y programación, han de basarse en unos referentes legislativos comunes para todos los docentes, siendo relevante mencionar el planteamiento de una correcta estrategia metodológica afín a esta temática. Atendiendo al artículo 6 bis de la Ley Orgánica 2/2006, de educación, coincidente con el artículo 6 bis de la Ley Orgánica 8/2013, para la mejora de la calidad educativa, la metodología docente deberá fomentar la reflexión e investigación, así como la realización de retos relacionados con la asignatura. Teniendo en cuenta las premisas anteriores, el manejo de información extraído de bases de datos y su posterior aplicación en la sociedad, tiene una directa ubicación en las materias de Tecnologías de la Información y la Comunicación I.

En la Región de Murcia, y teniendo en cuenta que el manejo de bases de datos resulta irrelevante si no se le da una aplicación práctica, los contenidos relacionados con el tratamiento de datos se pueden ubicar dentro de los bloques de programación y control (bloque 5) y de programas informáticos (bloque 3), ubicándose ambos en el Anexo III del Decreto 221/2015, de 2 de septiembre. En el artículo 5.5 de la misma orden se recoge que los criterios de evaluación se concretan en estándares de aprendizaje

evaluables. Por lo que, para estudiar el tratamiento de los datos, se han de dar respuesta a los siguientes estándares:

- E.A.E. 3.1.1. Diseña bases de datos sencillas y/o extrae información, realizando consultas, formularios e informes.
- E.A.E. 5.1.1. Desarrolla algoritmos que permitan resolver problemas aritméticos sencillos elaborando sus diagramas de flujo correspondientes.
- E.A.E. 5.5.1. Realiza programas de aplicación sencillos en un lenguaje determinado que solucionen problemas de la vida real.

Para poder manejar grandes cantidades de información, los servidores de páginas web cuentan con bases de datos a los que el público puede acceder fácilmente con determinadas credenciales. Pese a que la mayoría de las bases son de índole privada, existen otras bases que son totalmente accesibles y gratuitas, sin embargo, no se pueden acceder a ellas sin utilizar herramientas de extracción de datos. Si se hace hincapié en datos con notable relevancia social (por ejemplo, tráfico o condiciones meteorológicas), se puede observar cómo las administraciones han intervenido mejorando y facilitando su acceso, por ello, se puede observar que desde el año 2010, siguiendo directrices europeas, la Agencia Estatal de Meteorología (AEMET) adoptó una nueva política de datos que permitió acceder libremente a sus bases de datos de forma progresiva, siendo en marzo de 2019 cuando habilitó una plataforma destinada a desarrolladores mejorando así la transferencia de conocimiento.

Según AEMET (2019), la obtención de datos referentes al clima ha sido relevante en agricultura, aviación o actividades marítimas y por ello, desde el siglo XVIII, han sido guardados y organizados. Aunque a principios del año 1900 la red contaba con pocos puntos de observación, es en el año 1911 cuando se amplía esta red con colaboración de personal voluntario que anotaba temperatura y precipitaciones. Casi una década después el Servicio Meteorológico se reorganizó y aparecieron las primeras legislaciones acerca de la observación y la incorporación de nuevas estaciones, siendo en la década de los 80 cuando se llega al número máximo de estaciones sobrepasando las 800 ya ubicadas en zonas de déficit de observatorios o áreas de difícil acceso, vertiendo todas ellas los datos al Banco Nacional de Datos Climáticos (BNDC). La importancia de este banco de datos radica en que sirve como elemento fundamental para la vigilancia del clima, así como para el desarrollo de investigaciones referentes a esta temática.

Ya que AEMET tiene responsabilidades asignadas en el tratamiento de la información meteorológica como consecuencia de la aplicación del Real Decreto 186/2008, de 8 de febrero, creó en el 2015 una nueva base de datos, paralela a la existente y que funciona a día de hoy, recogiendo datos de los distintos observatorios principales y que son volcados en el BNDC.

La implantación de la nueva base de datos ha repercutido en una mejora del servicio disminuyendo radicalmente el tiempo de disponibilidad de los datos. En consecuencia, se ha ampliado el catálogo de productos que ofrece AEMET alcanzando los 1.800 millones de datos diarios y 2.400 millones de datos diezminutales. Dentro de este catálogo, para promover la difusión de información a administración, usuarios o empresas, parte de dichos datos son de libre acceso y totalmente gratuitos disponibles en la pestaña “Datos abiertos” en la web de AEMET. Además, todos los datos recogidos en las estaciones están conectados, en estrella, a nodos de comunicación de alta velocidad, con nexo común en los servicios centrales, siendo las predicciones y fenomenologías adversas realizadas a partir de los datos suministrados por el modelo HARMONIE del Centro Europeo de Predicción a Plazo Medio (AEMET, 2016).

A nivel internacional, según la web de datosclima (2019), se puede acceder a los datos meteorológicos de casi 100.000 estaciones repartidas en todo el mundo gracias a la Administración Nacional Oceánica y Atmosférica (NOAA). Esta agencia dependiente de los Estados Unidos elabora la “Global Historical Climatology Network” (GHCN), una fuente de datos donde la mayoría de los países vuelcan sus resultados y predicciones, convirtiéndose así en la mayor fuente de datos para estudios climatológicos. Sin embargo, dicha base tiene un problema de consistencia de datos debido a la normalización que deben cumplir todos los países en el volcado de datos. Es relevante destacar que numerosas aplicaciones y webs meteorológicas hacen uso de estas bases de datos y no de las estatales.

Analizando las principales aplicaciones meteorológicas disponibles en la web, podemos mencionar que muchas de ellas basan sus datos en el proyecto de Índice Mundial de Calidad del Aire (AQICN) que recoge datos de 11.000 estaciones mundiales de más de 90 países. En dicha web se recogen, de manera totalmente gratuita y accesible, las condiciones básicas meteorológicas (temperatura, presión y humedad) junto con partículas contaminantes (NO₂, SO₂, CO, O₃ y PM10), relevantes para evaluar la contaminación en las ciudades y los efectos del cambio climático. Dichos datos son publicados en tiempo real, por lo que no están validados, aunque, para garantizar la consistencia de las mediciones dichos valores se comparan con las estaciones vecinas, pudiendo detectar así las estaciones que funcionen de manera errónea.

Se conoce como extracción de datos Web, al conjunto de metodologías que permiten obtener información de diferentes orígenes de datos accesibles (Grigalis & Cenys, 2013). Dentro de las técnicas existentes, se encuentran procesos automáticos de recolección de información de múltiples páginas web de manera sistemática, también conocido como Web Scraping, siendo esta una de las técnicas de extracción más antiguas utilizadas en el mundo de la informática (González-Peña et al., 2014), ya que permite obtener datos de páginas web sin requerir credenciales. La extracción de datos Web, se lleva a cabo habitualmente en documentos estructurados como HTML, siendo necesario analizarlos para mostrar su estructura interna (Gultom et al. 2010). Este proceso de extracción de datos orientado a una página Web se divide en 5 pasos (Baumgartner, 2009): interacción con el sitio Web mediante una dirección URL, soporte para la generación y ejecución del contenedor, planificación, refinamiento o filtrado de datos y extracción de datos para la aplicación deseada.

Sin embargo, para evitar tener acceso a información privada y no depender de que el programador cambie el código fuente del sitio web a analizar, los desarrolladores crearon las interfaces de programación de aplicaciones (API), las cuales definen un conjunto de reglas y especificaciones que permiten interactuar con aplicaciones software, mejoran la reutilización del código y promueven la productividad de los programadores (Qiu et al. 2016). Una vez ejecutadas las técnicas enumeradas, todas estas tienen como finalidad común obtener resultados en forma de archivo de texto con notación de objeto de JavaScript (JSON) que debe ser interpretado por cualquier código de programación que separe su contenido en las distintas variables solicitadas por el usuario. Aplicando estas técnicas en bases de datos meteorológicas se observa que el nivel de dificultad para el acceso a los datos no es muy complicado, pero sí se requieren nociones básicas de programación que son estudiadas por el alumnado en el bloque de contenidos de programación de TIC.

Según AEMET (2019), la AEMET OpenData es una API que permite acceder a los datos recogidos en el BNDC históricos desde el año 1981. Para poder descargar los datos de la API, el usuario debe obtener una API key personalizada aportando su correo electrónico. Una vez que se ha obtenido la clave y se tenga acceso al listado de estaciones, junto con su indicador de éxito o fracaso, el programador deberá comprobar si dicho indicador es afirmativo, en caso contrario no se podrán disponer de datos y habrá que esperar hasta la próxima actualización. En dicha descarga ya están contenidos los diferentes datos meteorológicos que pueden ser descargados y vienen codificados como se muestra en la figura 1.

Figura 1

Extracto de datos generados en la API.

```
[ {  
  "idema" : "7237E",  
  "lon" : -1.158615,  
  "fint" : "2019-06-13T19:00:00",  
  "prec" : 0.0,  
  "alt" : 170.0,  
  "vmax" : 6.5,  
  "vv" : 2.4,  
  "dv" : 141.0,
```

Debido a que el proyecto AQICN no cuenta con plataforma API, se debe analizar el código fuente de la página web con el fin de obtener el nombre que recibe cada variable. Atendiendo a la declaración de variables, según esta web, se observa que las principales están recogidas en los siguientes contenedores de información:

- 'nu' temp=: temperatura actual (°C)
- 'cur_p' class='tdcur': presión actual (hPa)
- 'cur_w' class='tdcur': velocidad del viento (km/h)
- 'cur_h' class='tdcur': humedad actual (%)
- ('cur_no2', 'cur_so2', 'cur_pm10', 'cur_o3', 'cur_co') class='tdcur': concentraciones de contaminantes en µg/m³.

Una vez obtenidas las declaraciones de las variables, se debe recorrer todo el código fuente hasta encontrar estos referentes y extraer el dato que se encuentra a continuación de dicho código. Es por ello que, adicionalmente, se deberán usar técnicas de extracción, recorte y composición de cadenas de texto.

El objetivo del presente artículo pretende dar respuesta a esa necesidad de concienciación ambiental amparándonos en herramientas informáticas y recursos TIC, ofreciendo una propuesta didáctica que permita fomentar competencias transversales. Para ello, se plantea como objetivo general el diseño de una actividad en la que el alumnado usará técnicas de extracción de datos de bases climatológicas, los cuales usará para darles un uso posterior con un microcontrolador. Para poder alcanzar el objetivo general planteado, se debe dar respuesta, de manera progresiva, a los siguientes objetivos específicos: análisis del código fuente de páginas web que contengan bases de datos meteorológicas, elaboración de rutinas de programación personalizadas a estas bases de datos, diseño de una aplicación móvil basada en el software libre App Inventor, implementación de las rutinas de programación en App Inventor, conexión de bluetooth entre APP Inventor y Arduino, implementación de un circuito led controlado por Arduino para mostrar los resultados extraídos.

Material y Método

Diseño del Estudio

El éxito de esta propuesta versa en conseguir que el alumnado domine los principales conceptos de la programación y algoritmia para obtener datos de texto y consiga separarlos en otras variables de interés. Teniendo en cuenta la legislación vigente en la Comunidad Autónoma de la Región de Murcia, la actividad propuesta se ubica en la materia de Tecnología de la Información y Comunicación en el primer curso de bachillerato, según el Anexo III del Decreto 221/2015. Analizando dicho anexo, encontramos los contenidos que el alumnado debe adquirir y que son los siguientes:

- Programación. Proceso y metodología. Diseño de algoritmos para la resolución de problemas. Diagramas de flujo. Diseño y seguimiento de un diagrama.
- Descomposición modular de un problema. Bloques funcionales.
- Paradigmas y tipos de lenguajes de programación. Lenguajes estructurados. Sintaxis de un lenguaje de programación estructurado determinado. Elementos y construcciones básicas. Estructuras de datos sencillas.
- Procedimientos y funciones. Entrada y salida. Interacción con el usuario.
- Creación de programas sencillos funcionales. Característica del diseño de estudio e información relevante a la muestra, criterios de inclusión, exclusiones, etc. La estructura del método se debe de adaptar a cada diseño de estudio. Iniciar con mayúscula la palabra con más de cuatro letras.

Material

Para poder desarrollar los contenidos implicados, se deberá seleccionar una variedad de recursos didácticos de entre todos los existentes, ya que en el Anexo II de la Orden ECD/65/2015, se menciona que la selección de estos constituye un aspecto esencial de la metodología. Los recursos didácticos que se emplearán serán los siguientes:

- Ordenador con acceso a Internet y plataforma App Inventor.
- IDE Arduino vs. 1.8.9 y cable USB.
- Móvil o tablet con conexión Bluetooth.
- Kit básico de Arduino Uno con componentes electrónicos elementales.
- Display LCD 16x2 o LCD de creación casera.
- Placa protoboard, conectores y cables de distintos colores.
- Módulo bluetooth HC-05 o HC-06.
- Programa Fritzing vs. 0.9.3b para la elaboración de planos esquemáticos.

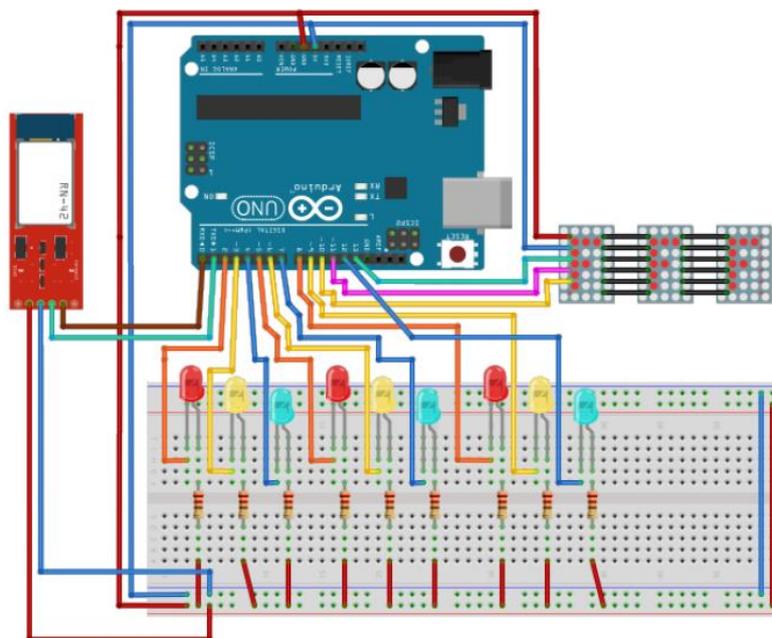
Desarrollo del Hardware

La propuesta planteada deberá mostrar todas las propiedades climatológicas expuestas en apartados anteriores, así como los contaminantes, en caso de tener datos en las bases de datos analizadas. Para ello se empleará un display que mostrará con un barrido horizontal las distintas propiedades, siendo estas actualizadas manualmente por el alumnado. Si bien, como profundización se puede plantear que se introduzcan unas rutinas de programación que actualicen el panel en cortos periodos de tiempo.

El panel display seleccionado cuenta con 3 módulos led 8x8 controlados con una tarjeta controladora MAX7219. El montaje puede observarse en la figura 2:

Figura 2.

Montaje de los componentes en la placa protoboard.



Dichos paneles están conectados con cables Dupont en serie y son alimentados directamente desde la placa Arduino. Cada uno de los paneles cuenta con cinco terminales de conexión, siendo dos de ellos tierra y alimentación. Los tres restantes son el selector de fila, el selector de columna y la señal de sincronización (conectados a los pines 10, 11 y 13, respectivamente). El control de los leds se hace mediante software con dos librerías, por lo que no se necesitan componentes adicionales en el montaje del panel. Una vez montados los tres paneles, se monta el circuito indicador led de temperatura, humedad y velocidad de viento. Para ello, se utilizará una placa protoboard y resistencias limitadoras de 220 Ω . La placa de Arduino será la encargada de recibir la información de la app, por lo que el conexionado del módulo bluetooth y los pines leds se hará en los 28 terminales no ocupados por el panel.

Finalmente, se conecta el dispositivo bluetooth a la placa. De todos los módulos inalámbricos disponibles en el mercado, se ha optado por elegir el módulo bluetooth HC-05 que puede funcionar en modo maestro o esclavo, en consecuencia, se puede conectar con un móvil y recibir información de él. Debido a su facilidad de programación, con comandos AT, estos módulos son los más utilizados por los programadores de Arduino, además de ser los más económicos del mercado. En cuanto a requerimientos, estos módulos se alimentan con una tensión entre 3,3 y 6 V, por lo se podrá utilizar el pin de alimentación de Arduino de 5 V. Si bien, los terminales TX y RX donde se transmite y recibe información, han de estar conectados a una tensión de 3,3 V, por lo que, si se analiza a fondo la distribución de pines de la placa de Arduino, se puede observar que hay dos pines marcados con esta codificación y, por tanto, serán los idóneos para el conexionado.

Desarrollo del Software

En primer lugar, se ha de programar la app denominada AirQApp con la plataforma de App Inventor. La app mostrará una primera pantalla donde se seleccionará la base de datos nacional de AEMET o la internacional de AQICN. Tal y como se ha mencionado en apartados anteriores, la API de AEMET aporta datos haciendo llamadas web siguiendo una serie de rutinas. Para ello se comienza inicializando la lista de variables a utilizar, que serán: claveAEMET, estado (mostrará si AEMET tiene o no datos disponibles), estación, ciudad y provincia. Estas tres últimas variables serán las que, a continuación, serán completadas. Para ello, se inicializarán como listas vacías y, tras introducir la API key y pulsar en Obt. Estaciones, se rellenarán con los valores de todas las estaciones de las distintas comunidades autónomas.

Una vez seleccionada la estación de la que se quiere obtener datos, la app extrae el enlace proporcionado por la API junto con un código de éxito o fracaso (200 es que hay datos y 401 que no) que se guardará en la variable estado. En caso de que estado valga 200, se ejecutará el enlace de la estación obteniendo las distintas variables en un archivo de texto JSON. Finalmente, se lee este archivo de texto llamando a las variables que interesen con marcadores, por ejemplo, para obtener la temperatura, se deberá buscar dentro del texto JSON la variable "ta". Tras obtener las variables deseadas se compone el archivo, en formato string, que se enviará a Arduino por bluetooth y que irán separados por ";".

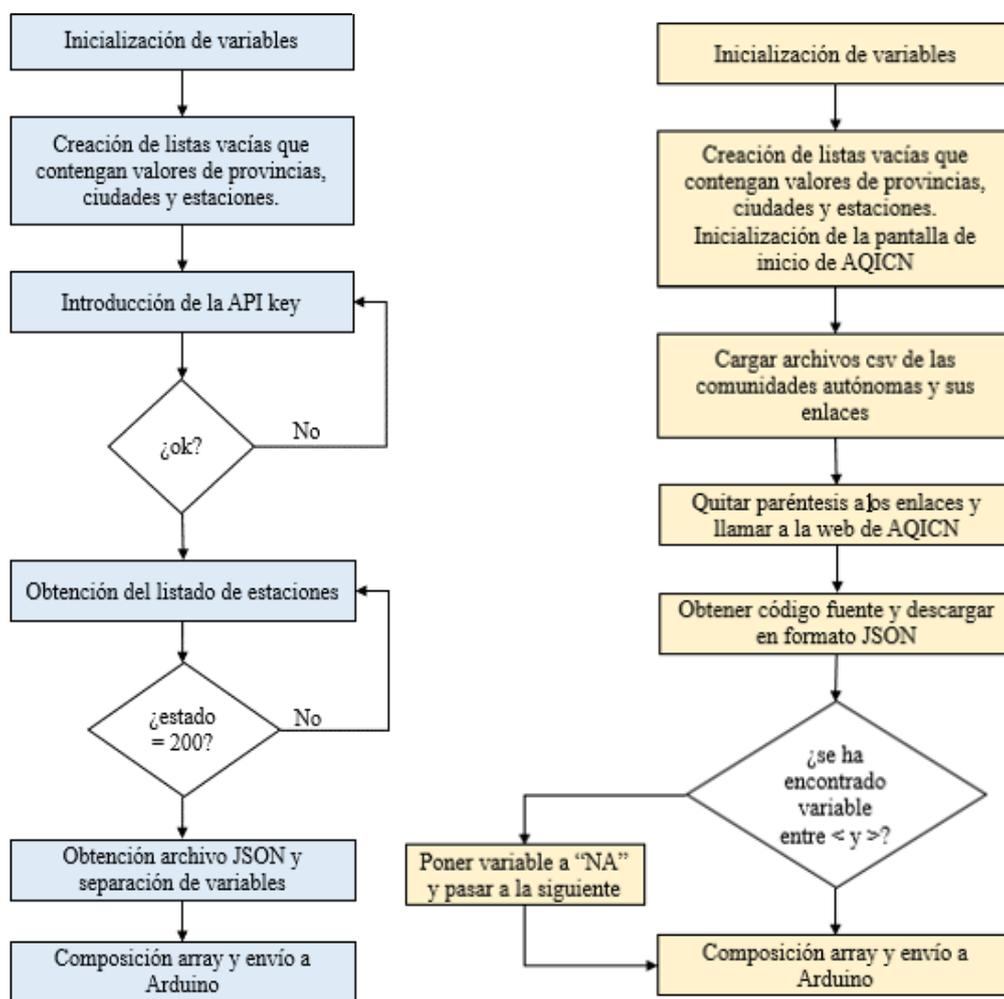
Si además de conocer las principales variables meteorológicas se quieren conocer los contaminantes medidos en las estaciones, se debe programar una rutina que entre a la web de AQICN y extraiga de sus bases de datos lo que se necesite. En primer lugar, el equipo de trabajo ha elaborado una hoja de cálculo donde se han anotado todas las estaciones meteorológicas de España y se han asignado a cada una de ellas, manualmente, el enlace web de AQICN asociado. Dicho archivo se guarda en formato csv y se importa a la aplicación en App Inventor. Las rutinas de programación en este caso son más sencillas que las de la API de AEMET, sin embargo, el proceso de obtención de datos es más complejo. Una vez inicializadas las variables "quitarparentesis", "ccaa", "enlace", t, p, h, v, NO₂, PM10, O₃, SO₂ y CO. Se carga en la variable "ccaa" las distintas estaciones de todas las comunidades españolas y en "enlace" las webs asociadas a cada una de estas. Como el archivo csv vierte las webs con paréntesis a App Inventor, es necesario ejecutar la función "quitarparentesis" que suprima el paréntesis inicial y final. Una vez ejecutada esta parte del código, se llama a la web y se descarga el código fuente HTML

en formato JSON. Finalmente, se lee el documento de texto JSON aplicando técnicas de comparación de caracteres, siendo evidente que el marcador de posición parará cuando se cumpla dicha iteración. Por ejemplo: la variable de temperatura es reconocida en el código fuente de AQICN como “'nu' temp=”.

Una vez que el marcador alcanza la primera variable, este devuelve un string con dos valores formado por todos los caracteres anteriores y todos los caracteres posteriores. Como nuestra medición se encuentra entre dos marcadores “>” y “<”, es necesario volver a recortar el string. Repitiendo este proceso para las demás variables se obtiene una nueva cadena de texto separada por “;” que se enviará por bluetooth a Arduino. Se muestra en la figura 3 a la izquierda el flujograma de funcionamiento de app Inventor para la obtención de datos AEMET y en la derecha de AQICN:

Figura 3.

Flujograma de funcionamiento de App Inventor. En la izquierda con API y en la derecha con scraping.



Por último, se ha de programar Arduino para que reconozca las cadenas de texto recibidas por bluetooth y las guarde en variables que serán enviadas al display. Para poder hacer funcionar el display se ejecutarán las bibliotecas aconsejadas por el fabricante: MD_Parola.h, MD_MAX72xx.h, string.h y SPI.h. A continuación, se definen los pines de control de la pantalla led (pin de selección de fila, columna y sincronización) y se inicializan los vectores que contendrán los resultados de las mediciones recibidas. Estos vectores son: “rxInfoDouble” que contendrá los resultados en formato de número, “rxInfoChar”

que contendrá los valores en formato texto, “rxInfo” que es la variable donde se volcarán directamente los resultados de App Inventor y un vector para la selección de datos que llamaremos “rawData”. A continuación, se configura la pantalla led modificando la velocidad, la dirección y el tiempo de espera final del texto. Como estas pantallas muestran los resultados que reciben en la memoria interna del procesador que tiene, se deberán inicializar y reasignar tamaños a sus buzzer.

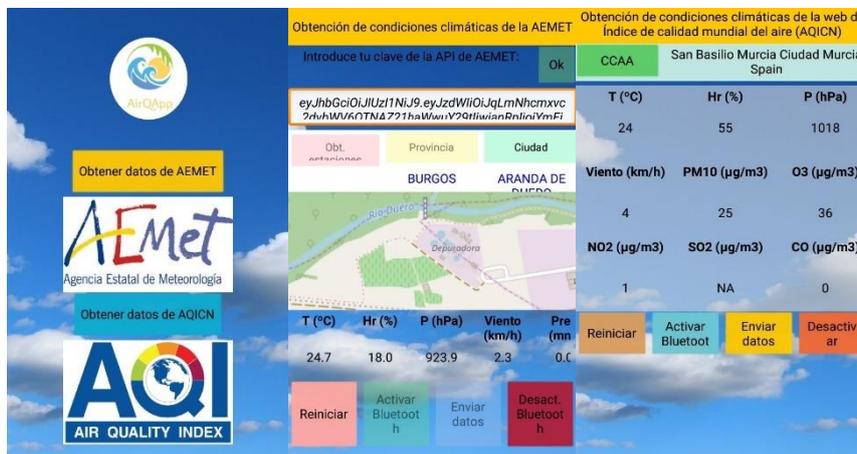
Una vez que se han recibido los datos de App Inventor se ejecuta la función “procesaDatos” que rompe la cadena de texto recibida separada por “;” guardándola en los vectores “rxInfo” y “rxInfoDouble”, anteriormente mencionados. Estos vectores serán los que se llevarán a la función “actualizaDisplay” que compondrá la cadena de caracteres a mostrar en el display.

Resultados

Una vez se han implementado en Arduino y App Inventor los códigos de programación anteriormente analizados, se compilan las aplicaciones y realiza el montaje del prototipo de panel indicador para comprobar su funcionamiento. Para obtener datos de la API de la AEMET se ha generado un código personal que ha sido recibido por correo electrónico y es el que se ha usado para obtener los datos que se han ido recogiendo a lo largo de las pruebas realizadas. Sin embargo, para acceder a la base de datos de AQICN se ha tenido, en primer lugar, que obtener el enlace web de todas las estaciones meteorológicas de España de las que esta base de datos tiene acceso y después generar un archivo csv delimitado por comas. Se ha generado, antes de comenzar toda la parte de programación, una lista de 153 estaciones repartidas en todo el territorio nacional. En la figura 4 se muestra la aplicación AirQApp diseñada con la plataforma App Inventor donde se puede observar, en la pantalla central, los datos obtenidos de la AEMET de Murcia y, en la pantalla de la derecha, los datos obtenidos con la base internacional AQICN.

Figura 4.

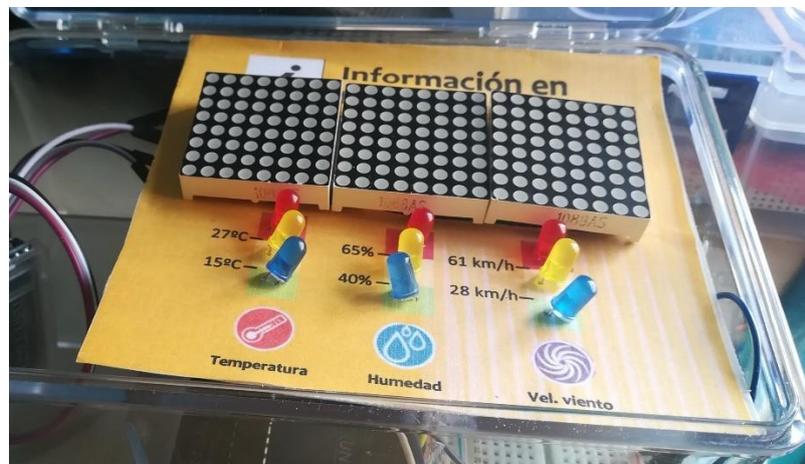
Resultados de la estación de Murcia (San Basilio) obtenidos de AirQApp.



En cuanto al funcionamiento del prototipo, se observa que este cumple la función para la que ha sido programado. Sin embargo, pueden surgir problemas a la hora de los conexiones que dependen, básicamente, de los materiales de los que se disponga. Para una mejor asimilación de los contenidos, se ha empleado una placa protoboard así como elementos electrónicos de elevadas dimensiones. Si bien, en el aula es recomendable utilizar este tipo de elementos con el fin de detectar rápidamente los problemas que le pueda surgir al alumnado. Asimismo, para una aplicación real en el ámbito público se podrían utilizar placas impresas con componentes y pantallas led más eficientes con mayor resolución. En la figura 5 se muestra un montaje similar al que se ha planteado como propuesta de actividad:

Figura 5.

Montaje final propuesto.



Discusión

Analizando los fundamentos de las técnicas API o Scraping surge la problemática de la accesibilidad puesto que, si el código fuente cambia o se modifica, dichas técnicas de programación deberán de modificarse y adaptarse a los cambios, suponiendo esto que nuestro panel informativo no responda a los objetivos planteados. Se observa en la experiencia realizada en el aula que los resultados obtenidos al aplicar distintas estrategias de extracción de datos difieren. Esto no se debe a las sentencias usadas en la programación del código fuente, sino que se debe a la existencia de un problema de fondo asociado al desfase de sincronización entre las mediciones de las estaciones de la AEMET y las mediciones existentes en las bases de datos meteorológicas internacionales. Además, mientras se ejecuta la aplicación surgen varios errores, en ocasiones puntuales, debido al retraso en la conexión bluetooth o que no carga los datos de las estaciones y hay que reiniciar la app. En consecuencia, considerando lo anterior, si lo que se quiere obtener es una temperatura precisa del momento estudiado, se deberá usar una técnica API. Si por el contrario se quisiera conocer la contaminación de una determinada forma, se debe, inexcusablemente acudir a una técnica scraping ya que la API de AEMET no proporciona esos datos.

Si se analiza el comportamiento individual de cada uno de los elementos empleados en el prototipo realizado, se observa que el módulo bluetooth ha de ser programado tanto en App Inventor (para el envío de datos) como en Arduino (para la recepción de datos). Esto conlleva una dificultad extra en el proceso de enseñanza-aprendizaje del alumnado puesto que ha de conocer la declaración de varios tipos de variables, así como la creación de arrays, recorte y unión de estos a partir de un código fuente html base. Es por ello, que se han de profundizar este tipo de contenidos en la materia previamente. En el caso que nos compete, es la cadena de texto la única que se adapta a la necesidad planteada y considerando la tecnología y programación de los componentes. Si se tiene en cuenta la facilidad de programar Arduino, se puede afirmar que cumple las expectativas ya que no se requieren datos precisos.

La obtención de las listas de estaciones meteorológicas en la web AQICN también tiene un trabajo extra que no se puede automatizar, siendo necesaria una mejora de propuesta para este hecho en estudios sucesivos. En cuanto a la API de AEMET, la obtención del listado de estaciones es inmediato, aunque estas son sólo a nivel nacional. En este caso, se deberá facilitar al alumnado una lista de hipervínculos en un archivo csv que será cargado en la app creada para el móvil.

Conclusiones

En el presente proyecto se ha planteado una propuesta didáctica a impartir en el bloque de programación de TIC I como respuesta a una necesidad de concienciación medioambiental que se recoge en los principios de la UNESCO (2004). Asimismo, se responde a la necesidad de fomentar competencias transversales según la Orden ECD/65/2015.

Se ha podido comprobar cómo las técnicas API o scraping son efectivas a la hora de extraer datos de las páginas web. Sin embargo, surge la problemática de la accesibilidad, puesto que, si el código fuente cambia o se modifica, dichas técnicas de programación deberán de modificarse y adaptarse a los cambios, suponiendo esto que nuestro panel informativo no responda a los objetivos planteados. Adicionalmente se encuentra como limitación la diferencia de resultados que ofrecen las distintas alternativas. La razón se debe, tal y como se ha comentado anteriormente, a que la sincronización de las bases de datos nacionales e internacionales tienen desfases de más de 15 minutos. Esto hace que, si en la práctica se tuviese que montar dicho dispositivo, se acudiese únicamente a la base de datos de la AEMET. Por el contrario, si lo que se quieren son datos de contaminantes, deberá de acudir a la web de AQICN, ya que es la única que los tiene accesibles, aunque no sincronizados a tiempo real.

Además, se ha diseñado una aplicación en App Inventor con varias pestañas de acceso a las diferentes bases de datos. Aunque el funcionamiento es correcto, se pueden remarcar algunas limitaciones que se presentan en App Inventor y que nos afectan negativamente. Por ejemplo, la recodificación en archivos de texto separados por “;”, ya que es más simple de reconocer a Arduino y se manda en una sola secuencia de bluetooth, la obligación de depurar el código para evitar errores de funcionamiento o la necesidad de exportación manual de archivos a la app, ya que las rutinas de programación para buscar el archivo en el móvil personal aumentan en muchas líneas el código ejecutable. En cuanto al conexionado del módulo bluetooth, se hace necesaria una programación tanto en App Inventor como en Arduino. La limitación que se presenta es el formato en el que se envían los datos, observando que el texto es el que mejor se ajusta a nuestras expectativas. Sin embargo, la programación en Arduino se hace más tediosa ya que se tiene que romper el string recibido y esto tiene como consecuencia una pérdida de precisión en los resultados.

Finalmente, mencionar que, para optimizar el código de programación en la IDE de Arduino, se han de usar librerías externas y, en consecuencia, existe la obligatoriedad de explicar al alumnado la programación de éstas. Se hace evidente que el uso de librerías simplifica el código, aunque para algunos procedimientos como la separación de datos de archivos JSON, se necesita dominar la composición y rotura de cadenas de caracteres. En cuanto a la implementación del prototipo, el alumnado deberá dominar el conexionado del panel led teniendo en cuenta el conexionado de sus pines, así como el ensamblaje de los marcadores led de temperatura, humedad y velocidad del viento, dificultando esto el montaje y requiriendo de numerosas resistencias y conductores que complican el resultado final.

Contribución de cada Autor: el diseño de la investigación, los métodos, los procedimientos y la escritura del documento ha sido elaborado por JCGS y FJML. La validación del prototipo, supervisión y depuración del código ha sido realizada por JCGS y FJML.

Conflicto de Intereses: Los autores declaran que no tienen conflictos de intereses.

Referencias

- AEMET. (2016). *La observación en AEMET*. Recuperado el 14 de junio de 2019, de AEMET Blog: <https://aemetblog.es/2016/11/28/la-observacion-en-aemet/>
- AEMET. (2019). *Registros climáticos: AEMET*. Recuperado el 14 de junio de 2019, de http://www.aemet.es/es/idi/clima/registros_climaticos
- Almaraz, F., Maz, A., & López, C. (2015). Tecnología móvil y enseñanza de las matemáticas: una experiencia de aplicación de App Inventor. *Epsilon*, 32(3), 77-86.

- Badia, A., Meneses, J. & Sigalés, C. (2013). Percepción de los docentes sobre los factores que afectan el uso educativo de las TIC en el aula equipada de tecnología. *Electronic Journal of Research in Educational Psychology*, 11(31), 787-808. <http://dx.doi.org/10.14204/ejrep.31.13053>
- Baumgartner, R., Gatterbauer, W. & Gottlob, G. (2009). *Web data extraction system*, *Encyclopedia of Database Systems*. Springer US, 3465-3471.
- Cortés, J., Arbeláez, O., & Mendoza, J. (2009). El entorno lego mindstorms en la introducción a la robótica y la. *Redalyc*, 15(41), 42-45.
- Datosclima. (2019). Recuperado el 14 de junio de 2019, de <https://datosclima.es/>
- Decreto 221/2015, de 2 de septiembre, por el que se establece el currículo de Bachillerato en la Comunidad Autónoma de la Región de Murcia. Boletín Oficial de la Región de Murcia. Murcia, 3 de septiembre de 2015, núm. 203, 31594-32545.
- Freire, P. (1995). *La educación como una acción cultural*. EUNED.
- García, A., Bassilotta, V., & López, C. (2014). Las TIC en el aprendizaje colaborativo en el aula de Primaria y Secundaria. *Comunicar*, 21(42).
- González-Peña, D., Lourenco, A., López, H., Reboiro, M. & Fernández, F. (2014). Web scraping technologies in an API World. *Brief in Bioinformatics*, 15, 788.
- Grigalis, T. & Cenys, A. (2013). State of-the-art web data extraction systems for online business intelligence. *Informacijos Mokslai*, 64, 145-155.
- Gultom, R.A., Sari, R.F. & Budiardjo, B. (2010). Implementing web data extraction and making Mashup with Xtractorz, *IEEE 2nd International Advance Computing Conference*, 385-393.
- Jaén, M., & Barbudo, P. (2010). Evolución de las perspectivas medioambientales de los alumnos de secundaria de un curso académico. *Eureka*, 7(1), 247-259.
- Ley Orgánica 2/2006, de 3 de mayo, de Educación. Boletín Oficial del Estado. Madrid, de 4 de mayo de 2006, núm. 106, 17158-17207.
- Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa. Boletín Oficial del Estado. Madrid, de 10 de diciembre de 2013, núm. 295, 97858-97921.
- Lozano, J., Alcaraz, S. & Bernabeu, M. (2012). Competencias emocionales del alumnado con trastornos del espectro autista en un aula abierta específica de Educación Secundaria. *Aula Abierta*, 40(1), 15-26.
- Orden ECD/65/2015, de 21 de enero, por la que se describen las relaciones entre las competencias, los contenidos y los criterios de evaluación de la educación primaria, la educación secundaria obligatoria y el bachillerato. Boletín Oficial del Estado. Madrid, 29 de enero de 2015, núm. 25, 6986-7003.
- Qiu, D., Li, B. & Leung, H. (2016). Understanding the API usage in Java. *Information and Software Technology*, 73, 81-100.
- Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato. Boletín Oficial del Estado. Madrid, 3 de enero de 2015, núm. 3, 169-546.
- UNESCO. (2004). *Education for a Sustainable Development*. Recuperado el 3 de junio de 2019, de <https://en.unesco.org/themes/education-sustainable-development>
- Vallespín, I. (2014). Las 'apps' del insti. *La Verdad*. Recuperado el 10 de octubre de 2019, de https://elpais.com/tecnologia/2014/02/14/actualidad/1392406482_481618.html
- Vargas, M., Castillo, G., Sandoval, J., & Brambila, A. (2015). Arduino. Una herramienta accesible para el aprendizaje de programación. *Revista de Tecnología e Innovación*, 2(4), 810-815.