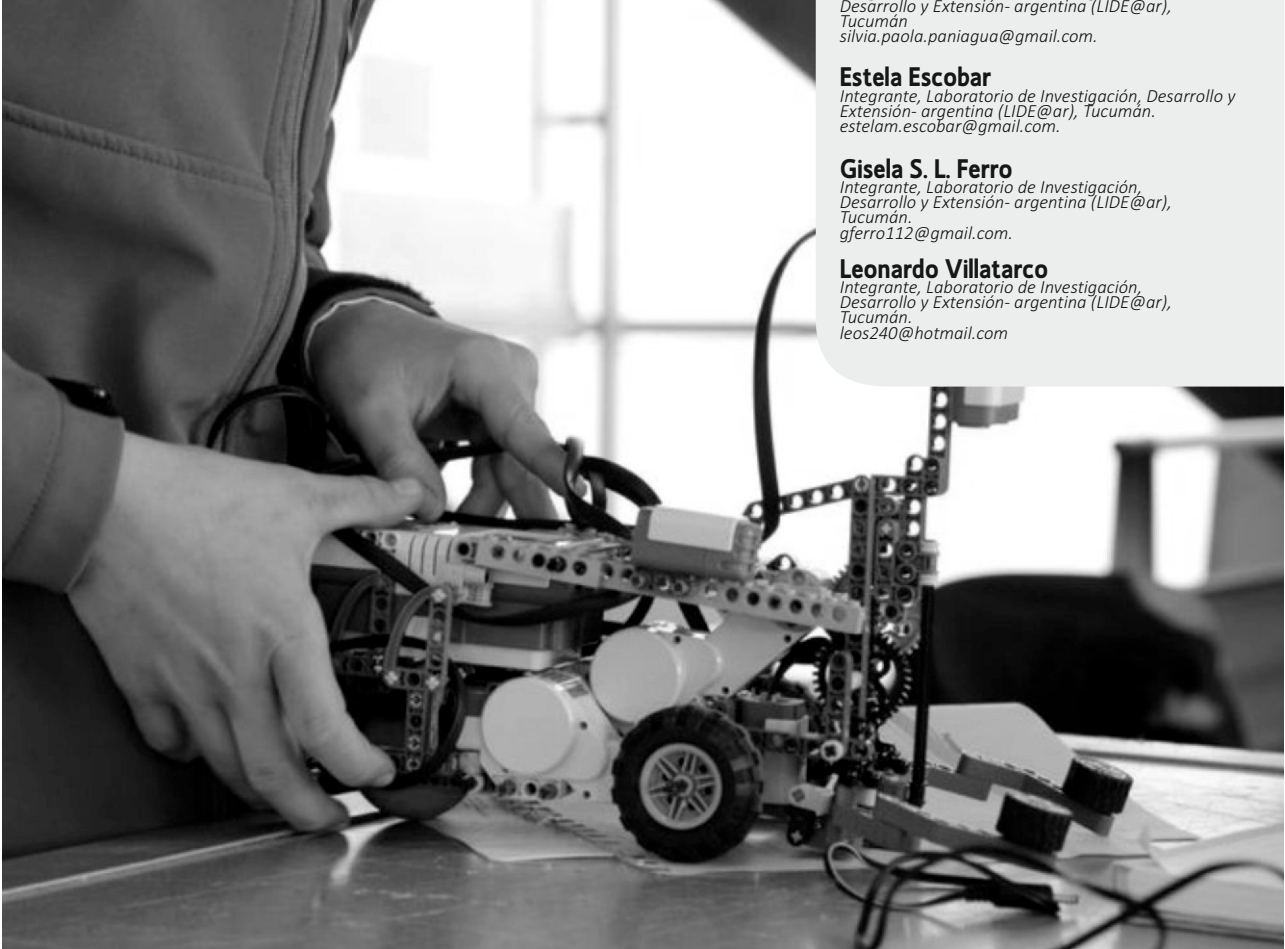


## Uso de robot como herramienta pedagógica. Entendiendo el robot seguidor de línea

Recibido: 15 Feb 2016 – Revisado: 30 Abr 2016

Aceptado: 30 May 2016 – Publicado: 30 Jul 2016



### Ana N. Rodríguez

*Coordinadora, Laboratorio de Investigación, Desarrollo y Extensión- argentina (LIDE@ar), Tucumán. arodriguez@herrera.unt.edu.ar.*

### Silvia P. Paniagua

*Integrante, Laboratorio de Investigación, Desarrollo y Extensión- argentina (LIDE@ar), Tucumán. silvia.paola.paniagua@gmail.com.*

### Estela Escobar

*Integrante, Laboratorio de Investigación, Desarrollo y Extensión- argentina (LIDE@ar), Tucumán. estelam.escobar@gmail.com.*

### Gisela S. L. Ferro

*Integrante, Laboratorio de Investigación, Desarrollo y Extensión- argentina (LIDE@ar), Tucumán. gferro112@gmail.com.*

### Leonardo Villatarco

*Integrante, Laboratorio de Investigación, Desarrollo y Extensión- argentina (LIDE@ar), Tucumán. leos240@hotmail.com*

**Resumen:** El presente trabajo muestra la manera en que los estudiantes de la carrera de Licenciatura en Informática realizan algoritmos sencillos que luego serán transferidos a alumnos de escuelas medias de población vulnerable, como una forma de despertar vocaciones en tecnologías de la información y la comunicación (TIC). En este caso, se muestra y analiza el desarrollo de un robot rastreador o seguidor de línea negra. Se realiza una descripción del software y hardware utilizado.

**Palabras clave:** Robótica, IDE, Inclusión, Proceso de enseñanza-aprendizaje, MiniBloq, Arduino

**Abstract:** The present article shows the way in which the students of Licenciatura en Informática make simple algorithms that will later be transferred to students of middle schools of vulnerable population, as a way of awakening vocations in information and communication technologies (ICT). It shows and analyzes the development of a tracking robot or black line follower. It has a description of the software and hardware used

**Keywords:** Robotics, IDE, Inclusion, teaching-learning process, MiniBloq, Arduino.

## 1. INTRODUCCIÓN

La robótica pedagógica se está destacando como un recurso eficaz para el trabajo interdisciplinario y brinda una mejora en los procesos de enseñanza-aprendizaje. Se caracteriza por el uso (o construcción) de un pequeño robot que controlado con software sencillo, permite a los alumnos aprender por ensayo-error a resolver determinadas situaciones problemáticas, por lo que se considera una herramienta pedagógica válida. Tiene como finalidad la de explotar el deseo de los educandos por interactuar con un robot para favorecer los procesos cognitivos.

Martial Vivet (1989) propone la siguiente definición de robótica pedagógica:

Es la actividad de concepción, creación y puesta en funcionamiento, con fines pedagógicos, de objetos tecnológicos que son reproducciones reducidas muy fieles y significativas de los procesos y herramientas robóticas que son usados cotidianamente, sobre todo, en el medio industrial.[1]

Las habilidades y competencias de los alumnos se pueden potenciar a través de diferentes actividades pedagógicas que permiten experimentar con diferentes principios, leyes, comportamientos en diferentes áreas del conocimiento. Por otro lado, la motivación que genera el trabajar con este tipo de artefactos crea condiciones propicias para mejorar el proceso de aprendizaje y fomentar la creatividad.

En este documento, se realiza el análisis y desarrollo de un algoritmo, ejemplo simple, con el uso robot modelo Múltiplo N6 de Robotgroup.

### 1.1. ¿Por qué usar un robot?

Cuando los alumnos comienzan a plantear algoritmos, las preguntas que surgen sobre el cómo, por qué, para qué y qué verá como resultado le ayudan a desarrollar ciertas habilidades como capacidades de abstracción

y lógicas; sin embargo, siempre queda el deseo de ver que la solución propuesta sirve y se desea materializar el conocimiento en forma inmediata. Esa solución inmediata es fácilmente demostrable con el uso de un robot como herramienta pedagógica.

Cuando se trata de estructuras de control, el uso del robot y un entorno de programación adecuado hacen más comprensible y fácil el proceso de enseñanza-aprendizaje, ya que el estudiante puede realizar las pruebas con él. Esto resulta entretenido para el alumno, alienta el trabajo en equipo, facilita la enseñanza y materializa ese pensamiento abstracto en acciones concretas que llevará a cabo el robot.

### 1.2. Robots seguidores de línea negra

En este documento, se tratará el caso de los robots seguidores de línea negra. Estos robots son muy utilizados en la industria para acarrear material pesado o peligroso para seres humanos; además, son frecuentemente usados cuando deben acarrear carga pesada.

## 2. MATERIALES Y MÉTODO

### 2.1. Del robot

El Robot Múltiplo N6 ha sido donado por la Fundación Sadosky, con el único fin de enseñar a programar a alumnos de escuelas medias de la República Argentina.

Este robot, realizado por la empresa RobotGroup, ha sido diseñado en Argentina.

En este apartado, se hará la descripción mínima y necesaria, de acuerdo a las necesidades del objeto de este documento.

- Este robot es parte del sistema constructivo Múltiplo N6, que se puede desarmar y reconfigurar de acuerdo a las necesidades del usuario.

- Posee tracción diferencial con dos cajas reductoras, con engranajes y bujes metálicos.
- Cuenta con dos sensores infrarrojos que le permiten seguir una línea negra sobre un fondo blanco.
- Tiene un sensor ultrasónico que permite detectar obstáculos u objetos con centímetros de precisión hasta una distancia de seis metros.
- Se mueve de forma independiente, utilizando tres baterías comunes (pilas) tipo AA que entregan doce voltios a los motores.

Este robot también puede moverse comandado por control remoto.

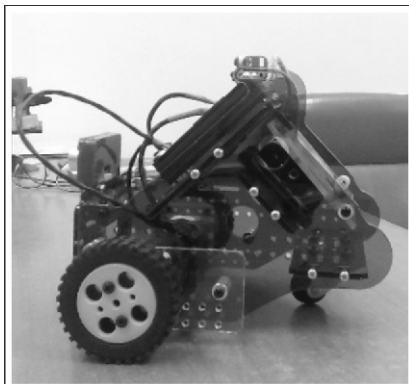
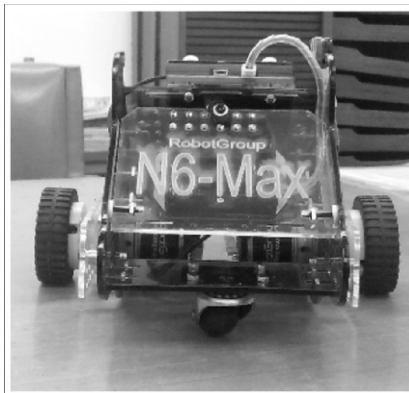


Figura 1. Robot Múltiple N6.

## 2.2. Placa Arduino

El robot posee una placa Duinobot, modelo Duino v2.3 HID.

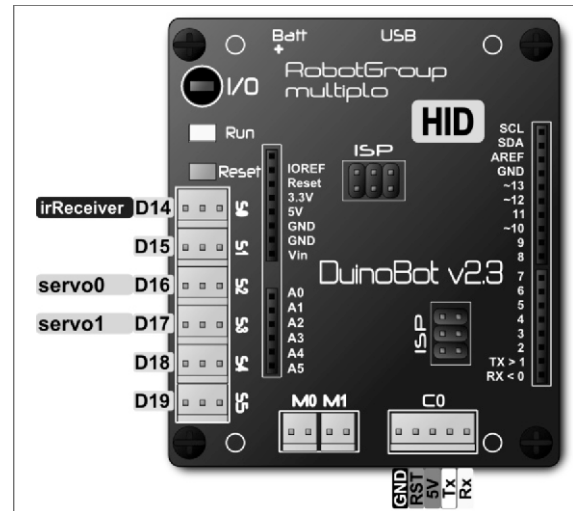


Figura 2. Placa DuinoBot v2.3.

## 2.3. Software

Para realizar este ejemplo, se utilizó el software MiniBloq V0.83. Este es un entorno de programación IDE (Integrated Development Environment) con un entorno gráfico amigable, que facilita la programación de la placa Arduino.

Creado por el argentino Julián da Silva Gillig, el IDE para programar robots ha traspasado las fronteras del país y ha encontrado receptores en lugares tan remotos como las escuelas africanas o la Universidad de Harvard. Lo que lo hace interesante es su universalidad, ya que sirve para casi cualquier tipo de proyecto y está programado en GNU C/C++; usa librerías de código abierto wxWidgets.

El lenguaje de programación de Arduino es una variante del lenguaje C, de modo que a los estudiantes de la Licenciatura en Informática, les resulta fácil elaborar ejemplos para programar la tarjeta. LIDE@ar trabaja en la introducción a la enseñanza de

programación en escuelas medias, en donde los alumnos no poseen este conocimiento; por ello, la herramienta gráfica resulta una manera muy fácil de introducir conceptos complejos.

Esta interfaz gráfica cuenta con librerías de funciones embebidas en bloques gráficos que se pueden ir ensamblando en un flujo de trabajo que representará el algoritmo de la aplicación.

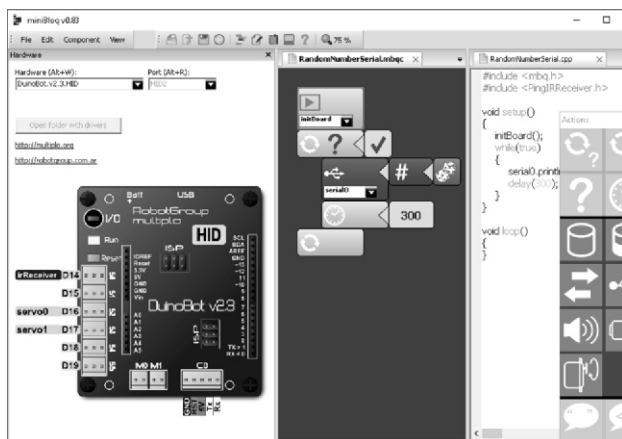


Figura 3. Entorno de programación.

### 3. DESARROLLO DEL EJEMPLO

#### 3.1. Estructura de entrada/salida del hardware

Es importante reconocer en la placa Arduino los pines de entrada y de salida a utilizar, de esta manera, se podrá tener claro las variables que se usarán durante la programación del algoritmo, ya que se encuentran directamente vinculadas (predeterminado por las librerías) con los pines de entrada y salida.

Para el seguidor de línea, es necesario (a priori) tener control sobre las ruedas y los dos sensores IR (infrarrojos).

Se utilizarán las variables S0 para el sensor izquierdo y S1, para el sensor derecho; M0 para el motor izquierdo y M1 para el motor derecho. En el siguiente diagrama, se describen dichas conexiones y variables asociadas:

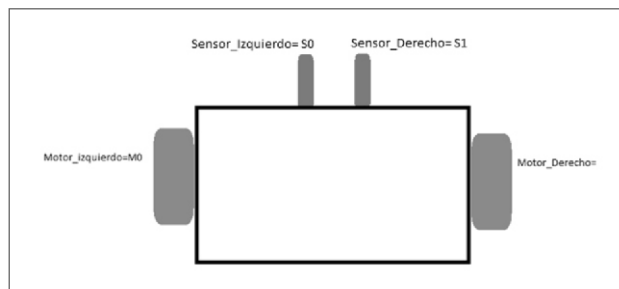


Figura 4. Modo de conexión y determinación de variables.

#### 3.2. Sensores

En el software Minibloq, todos los valores relacionados con periféricos con cierto tipo de entrada/salida analógica están normalizados entre 0 y 100; salvo el control de motores, donde la velocidad se puede establecer entre -100,0 y +100,0; el signo es el que determina el sentido de giro del motor. Esto se ha hecho así por varias razones, algunas de ellas son:

- Mayor independencia del hardware: El usuario no se ve obligado a lidiar con cuestiones tales como que el conversor analógico del microcontrolador esté configurado en 8 o 10 bits.
- Se logra código más compacto cuando se trata de leer un sensor y realimentar directamente un motor o una salida de PWM (Pulse-width modulation).
- Hace al código más claro para niños y principiantes. Por ejemplo, creemos que el 0 a 100 resulta bastante más natural que el 0 a 255.

Para poder hacer que el robot logre seguir una línea, es necesario que de alguna manera pueda distinguir entre al menos dos tipos de contrastes (el color de la línea y el fondo).

Se utiliza el sensor CNY70, que es un emisor y receptor de rayos infrarrojos.

El sensor CNY70 es capaz de diferenciar colores claros de oscuros, mediante el fenómeno físico de reflexión de los rayos infrarrojos y de acuerdo a esto, es capaz de retornar un valor entre 0-100.

Los colores claros tienden a reflejar con mayor facilidad la luz que los oscuros, que la absorben. Por ejemplo, si suponemos que el led infrarrojo despidió unos 100 rayos infrarrojos sobre una superficie y el fotodiodo receptor percibe 94, se puede concluir que la superficie donde rebotaron es clara. En cambio, si el fotodiodo percibe solo 20, se puede concluir que la superficie donde rebotaron es oscura.

### 3.3. Planteo del algoritmo

El algoritmo debe contemplar tres casos concretos, respecto de la línea a seguir:

(a) Línea recta.

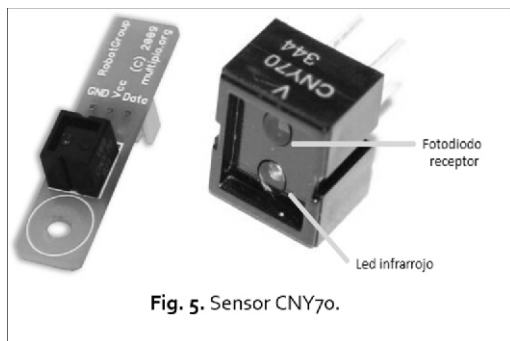


Fig. 5. Sensor CNY70.

- (b) Curva a la izquierda.
- (c) Curva a la derecha.
- (d) Otras consideraciones.

#### 3.3.1. Caso de línea recta

En una línea recta (cruces al lado de la línea) los sensores perciben al mismo tiempo que la superficie es clara. Por lo tanto, ambas ruedas giran en el mismo sentido y a igual velocidad.



Figura 6. Siguiendo una línea recta

#### 3.3.2. Curva a la izquierda

Cuando la línea se curva hacia la izquierda, el sensor izquierdo detecta una superficie oscura, mientras que el de la derecha sigue detectando una superficie clara. Entonces, el robot debe girar hacia la izquierda para volver a posicionar los sensores en la superficie clara.

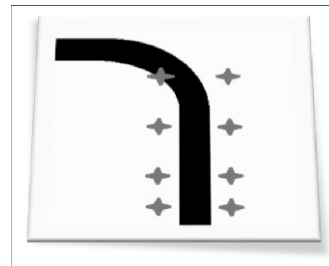


Figura 7. Curva a izquierda.

#### 3.3.3. Curva a la derecha

Del mismo modo que en la curva hacia la izquierda, el sensor de la derecha percibe un cambio de color y gira el robot hacia la derecha hasta detectar una superficie clara en ambos sensores.



Figura 8. Curva a la derecha.

#### 3.3.4. Otras consideraciones

Es importante tener presente que el sensor infrarrojo es sensible a la luz ambiente. Para evitar esta interferencia, hay dos opciones:

- (a) Recubrir con cinta aislante negra (u otro material oscuro) los sensores, de tal manera que se forme una especie de conducto que limita la interferencia de esta luz.
- (b) Reprogramar la sensibilidad de los sensores CNY70 dentro de la programación.

En este caso, el algoritmo está pensado de forma tal que es posible calibrar el punto en que la superficie se considera oscura, utilizando una variable de nombre transparencia.

### 3.4. Algoritmo

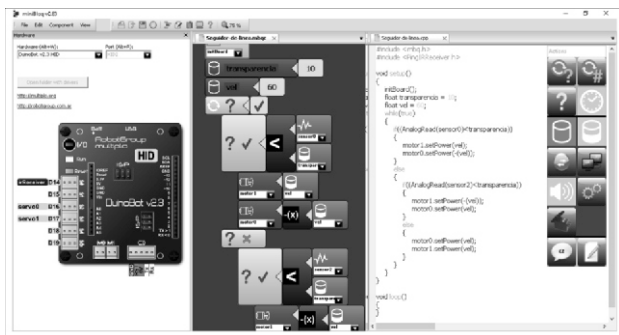


Figura 9. Vista en MiniBlok.

```
#include <mbq.h>
#include <PingIRReceiver.h>

void setup()
{
  initBoard();
  float transparencia = 10;
  float vel = 60;
  while(true)
  {
    if((AnalogRead(sensor0)<transparencia))
    {
      motor1.setPower(vel);
      motor0.setPower(-vel);
    }
    else
    {
      if((AnalogRead(sensor2)<transparencia))
      {
        motor1.setPower(-vel);
        motor0.setPower(vel);
      }
      else
      {
        motor0.setPower(vel);
        motor1.setPower(vel);
      }
    }
  }
}

void loop()
{
}
```

Figura 10. Código generado.

### 3.5. Cuestiones a tener en cuenta

- (a) Velocidad implementada:
  - Es crucial utilizar una velocidad no muy elevada, debido al tiempo que le lleva al procesador realizar un cambio en el giro de las ruedas.
  - También hay que tener en cuenta la inercia que esta provoca, ya que en una curva demasiada cerrada, puede ocasionar descarrilamiento.
- (b) Superficies lustradas o brillantes:
  - Las superficies de este tipo pueden llegar a afectar las lecturas realizadas por los sensores infrarrojos, dando una información errónea de la superficie, por más que el color sobre el que se encuentran sea oscuro.
- (c) Ancho de la línea:
  - Si la línea es muy angosta, puede que no dé tiempo al Robot para cambiar de dirección en las curvas y por inercia, se salga del camino.
  - Al aumentar el ancho de la línea, le damos más tiempo al procesador para detectar las curvas y efectuar el cambio de dirección.
- (d) Distancia de los sensores respecto al suelo:
  - Si la distancia es mayor a 1 cm, podemos tener mayor interferencia de la luz ambiente, como así también una mayor atenuación de los rayos infrarrojos, lo que provocaría una mala lectura.
- (e) Baja batería:
  - La falta de energía eléctrica puede ocasionar el mal funcionamiento de los sensores, puesto que reduce la intensidad con la que se emiten los rayos infrarrojos.

## 4. TRANSFERENCIA

Las clases, en las escuelas medias, comienzan mostrando al robot en movimiento. Esto despierta curiosidad en los estudiantes de las instituciones. Una vez que se capta la atención, se comienza explicando lo que es un robot y sus componentes esenciales. Es una breve explicación teórica acerca de los componentes de un robot y de la cotidianeidad de su presencia en nuestras vidas.



Posteriormente, se muestran las instrucciones básicas que facilita el programa: avanzar, girar a la derecha, girar a la izquierda.

#### 4.1. Primera interacción con el alumno-código-robot

Una vez explicada la forma en la que se escriben las instrucciones al robot, se propone como primera tarea dar vida, es decir, darle algún tipo de movimiento. Esto favorece la familiaridad con el entorno de programación y sus instrucciones.

Se propone como problema: detectar un obstáculo con el sensor. Se presenta alguna situación donde sea necesario usar un sensor de proximidad como esquivar un objeto, activar una alarma, etc.

##### 4.1.1. Problema del seguidor de línea

Se muestra el robot funcionando con el algoritmo del seguidor de línea y se propone entender lo que el robot hace. Como guía, se puede invitar a que los chicos imaginen que estamos en una habitación con un pedazo de cuerda sobre el piso a modo de circuito y nos encontramos parados sobre ella con un pie a cada lado, y que además llevamos los ojos vendados con la consigna de que caminemos siguiendo el circuito. De esta manera, se pueden comparar los sensores infrarrojos que detectan la línea negra con los pies que siguen la cuerda, cambiando de dirección según lo que estos sientan.

Los alumnos realizan su código y se les invita a probarlo en el robot.

## 5. RESULTADO

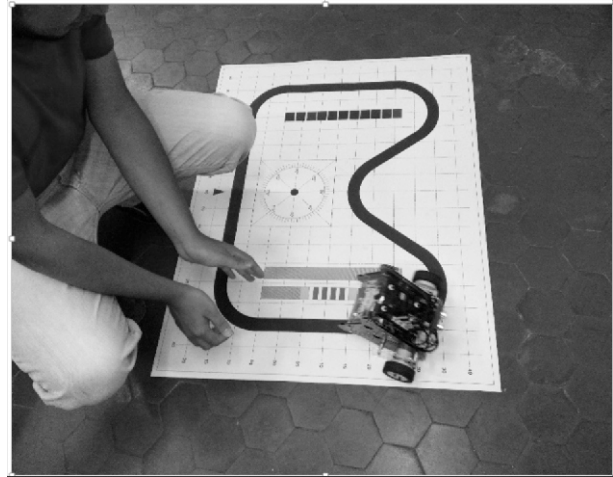


Fig. 11. Robot seguidor de línea negra.

Una vez cargado el programa en el robot, analizados el funcionamiento de los sensores y la velocidad de los motores, se pueden ver los resultados en una hoja de papel en la que se realizó el dibujo de una línea gruesa para que siga el robot. Luego, se hicieron pruebas en una hoja plastificada, que, en un principio, dio problemas, porque se reflejaba la luz; esto fue solucionado tapando la amplitud de barrido de los sensores con cinta aisladora.



Figura 12. Enseñanza en escuelas medias. En esta foto, están preparados para probar su código.

## 6. CONCLUSIONES

La experiencia fue realmente enriquecedora para el grupo.

La realización de pruebas y errores, materiales con la herramienta Robot”, brindó a los alumnos solidez, capacidad de abstracción, necesidad de investigación, búsqueda de bibliografía, entusiasmo por aprender y probar cosas nuevas.

Se puede decir que en este proceso, los alumnos lograron el aprendizaje significativo: «El aprendizaje significativo es, según el teórico norteamericano David Ausubel, el tipo de aprendizaje en que un estudiante relaciona la información nueva con la que ya posee, reajustando y reconstruyendo ambas informaciones en este proceso» (Ausubel, 1963, p. 53) Citado por Moreira[5]

Se alentó el trabajo en equipo y se planificaron estrategias de enseñanza para los alumnos de las escuelas de enseñanza media con las que se está trabajando.

**AGRADECIMIENTOS** a la Fundación Sadosky de la República Argentina, que nos facilitó los robots de prueba.

## REFERENCIAS

[1] Vivet M., (1989) "Actes du Premier Congrès Francophone de Robotique Pédagogique"- 206 pages - Laboratoire Informatique, Université du Maine, BP 535, 72017 Le Mans Cedex. (150,00 FF)

[2] AUSUBEL-NOVAK-HANESIAN (1983). Psicología Educativa: Un punto de vista cognoscitivo .2° Ed. TRILLAS México

[3] Da Silva Gillig, J. U. (2011). Minibloq v 0.81 Beta Manual del Usuario. Disponible en: <http://minibloq.tder.nl/Minibloq.v0.81.Beta.UserManual.sp.20120522.pdf>

[4] Ruiz Gutiérrez, J. M. (2012). Herramientas Gráficas para la programación de Arduino. Disponible en: <http://josemanuelruizgutierrez.blogspot.com/> <https://drive.google.com/file/d/0B4qM1H91FErNWl1MmZhNjctODkxOS00ZDQzLTljNjYtMTVjZDI4Zjc2NzJk/view>

[5] Moreira, M. "APRENDIZAJE SIGNIFICATIVO: UN CONCEPTO SUBYACENTE" – Disponible en: <https://www.if.ufrgs.br/~moreira/apsigsubesp.pdf>

[6] Ausubel, D. "Significado y aprendizaje significativo" – Disponible en: [http://www.arnaldomartinez.net/docencia\\_universitaria/ausubel02.pdf](http://www.arnaldomartinez.net/docencia_universitaria/ausubel02.pdf)