

IMPLEMENTACIÓN DE LA ROBÓTICA EDUCATIVA EN LA ESCUELA: UN ENFOQUE DIDÁCTICO PARA EL DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE ROBOTS CON ALUMNOS DE PRIMARIA

Rubén Darío Maza⁽¹⁾, Gustavo Adolfo Mamani⁽²⁾

Sede Regional Orán – Universidad Nacional de Salta (UNSa)

Laboratorio de Robótica Aplicada (LABRA)

⁽¹⁾rdmaza@yahoo.com

⁽²⁾gamamani@yahoo.com

Resumen

En este Trabajo se presenta un relato de la metodología utilizada por el Laboratorio de Robótica Aplicada (LABRA) perteneciente a la Sede Regional Orán de la Universidad Nacional de Salta (UNSa) en la enseñanza de la Robótica Educativa (RE) a alumnos de los últimos grados del nivel primario de la ciudad de Orán (Salta), utilizando para ello un kit robótico desarrollado bajo la modalidad “open source”, que posee elementos accesibles y de bajo costo, y que permite una fácil programación mediante el uso de un entorno visual basado en el lenguaje LOGO. De esta manera se pretende facilitar la posibilidad, de manera concreta y efectiva, de incorporar la RE en la currícula escolar, y presentarla además, como un ejemplo válido de la utilización correcta, adecuada y pertinente de la Tecnología en la Educación.

Palabras clave: ROBÓTICA EDUCATIVA – PROGRAMACIÓN VISUAL – LOGO – BLOCKLY – MICROCONTROLADORES – JAL

Introducción

En la actualidad existe, tanto a nivel nacional como internacional, una creciente

demanda insatisfecha de profesionales en las áreas de las ciencias exactas y las ingenierías [1]. Es por eso que, con el fin de generar vocaciones por carreras relacionadas con las denominadas STEM (acrónimo en inglés de Ciencia, Tecnología, Ingeniería y Matemática), el LABRA viene realizando una serie de acciones, materializadas en proyectos de investigación y de extensión, junto con la publicación de libros [2][3] y presentación de trabajos en congresos [4], tendientes a la difusión de la RE en las escuelas del nivel primario y secundario.

Como un ejemplo de ello, el LABRA ha realizado Talleres de Robótica para alumnos de 6to y 7mo grado en diversas escuelas públicas de la ciudad de Orán (Salta), en el marco del Proyecto de Extensión “Aprendiendo a Pensar con Robótica Educativa”, y para la construcción de los robots se utilizó un kit de robótica, diseñado y construido en el Proyecto de Investigación “Desarrollo de un Kit para la Construcción de Robots Autónomos Móviles Orientado a la Enseñanza y la Investigación”. Cabe acotar que uno de los talleres fue impartido en la Escuela N° 4.763 “Hugo Alberto Luna” de la ciudad de Orán, y como resultante de ello, su proyecto “La Robótica en la Escuela” recibió una mención especial en la instancia nacional de la “Feria de Ciencia y Tecnología”, que se

desarrolló del 14 al 20 de Noviembre de 2017 en Tecnópolis (Bs. As) [5].

Nosotros consideramos que en la currícula de la Nueva Escuela, que ineludiblemente deberá definirse, la RE tendría que estar presente, debiéndose comenzar con el nivel primario. Por lo tanto, resulta sumamente importante su difusión como una herramienta pedagógica, pues el aporte que la misma realiza al proceso de enseñanza–aprendizaje (Fig. 1) está ampliamente documentado [6][7]. Entre algunas de sus múltiples bondades, ella fomenta el desarrollo de la capacidad de abstracción y el pensamiento lógico formal de los alumnos, es decir el aprender a pensar, aptitudes que en estos momentos la escuela no las está brindando.

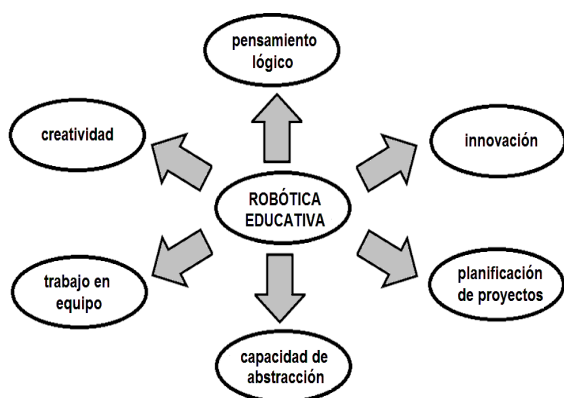


Figura 1: Bondades de la RE

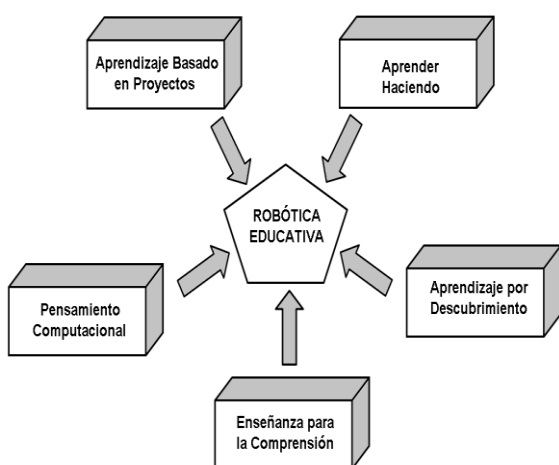


Figura 2: Marco Metodológico de la RE

Además, desde una perspectiva pedagógica, la RE presenta un sólido marco teórico,

basado en los trabajos de Piaget (constructivismo), Papert (construccionismo), Vigotsky (zona de desarrollo próximo) y Ausubel (aprendizaje significativo), siendo su basamento metodológico el que se muestra en la Fig. 2.

Y con la finalidad de hacer viable y accesible este objetivo, y como una alternativa al uso de los kits robóticos comerciales, este equipo de trabajo desarrolló bajo la modalidad hardware y software libre, un kit de bajo costo que tiene las siguientes características: modular, polimórfico, reconfigurable, de fácil programación y que utiliza materiales y componentes accesibles y de bajo costo, y que puede ser empleado en la enseñanza de la RE en las escuelas de nivel primario y secundario, e incluso ser utilizado en cursos de Introducción a la Programación a nivel universitario, posibilitándose de esta manera la implementación y puesta en práctica de los conceptos teóricos de la RE, que permite que los alumnos aprendan a:

- Construir su propio conocimiento, en el sentido piagetiano del término, pues los alumnos desempeñan un papel activo durante la construcción del robot.
- Acercarse a la ciencia y la tecnología de una manera práctica, lúdica, motivadora y significativa.
- Trabajar en equipo, es decir, realizar un trabajo colaborativo y asumiendo responsabilidades.
- Compartir sus conocimientos con el resto de sus compañeros.
- Planificar y administrar los recursos materiales que se tienen y del tiempo que se dispone, para la resolución de los problemas planteados.
- Integrar diversas áreas del conocimiento, como ser: Matemática, Lengua, Programación, etc. Es decir que la RE tiene un carácter eminentemente inter,

multi y pluridisciplinario (es decir, una transversalidad horizontal y vertical).

- Desarrollar el pensamiento lógico formal, y la capacidad de abstracción, elementos fundamentales para la resolución de problemas de todo tipo (incluidos los de la vida diaria).
- Desarrollar la creatividad y el interés por la investigación y la innovación, al fomentar la observación, la medición y la comparación sistemática.
- Crear dispositivos, experimentar con ellos, modificarlos y ver cómo funcionan.
- Utilizar el error como un elemento que permite corregir y mejorar lo realizado.
- Desarrollar la expresión oral y escrita, para explicar lo realizado por su equipo, al resto de sus compañeros.

La lista anterior no es taxativa, pues podríamos seguir agregando las innumerables ventajas de la RE. Además, en el listado anterior se observa que la RE fomenta el desarrollo de las habilidades y competencias esenciales para que los alumnos puedan desempeñarse en este nuevo siglo [8][9].

El Kit de Robótica MARGOT

El kit desarrollado se denomina **MARGOT** (acrónimo de *Mobile Autonomous Robot General Operative Tasks*), y su arquitectura física es la que se muestra en la Figura 3.

La plataforma mecánica consiste de una plancha de aluminio de 1 mm de grosor, que puede tener diversas formas. Como sistema de control posee la placa electrónica **MARGOT_28**, desarrollada ad-hoc (Figura 4), una serie de sensores (de contacto, de luz, de reflexión y de distancia), y como actuadores dos

motorreductores y tres ruedas (una de ellas libre).

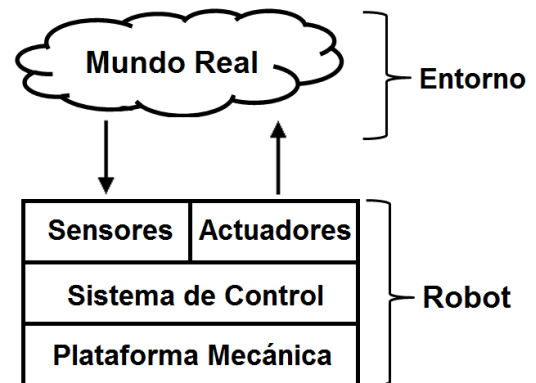


Figura 3: Arquitectura Física del Robot

Con todos esos elementos los alumnos puede construir robots autónomos móviles de tracción diferencial [10][11], comúnmente conocidos como del tipo “tortuga” (Fig. 5), designación que proviene del nombre que el neurofisiólogo Grey Walter les dio a unos pequeños robots electromecánicos que construyó a principios de 1950.



Fig. 4: Placa de Control MARGOT_28

Puesto que con el kit desarrollado se pueden construir robots del tipo “tortuga”, y debido a que su repertorio de instrucciones (Tabla 1) son iguales a las que posee la tortuga del lenguaje LOGO, entonces resulta natural (y lógico) que se utilice éste

lenguaje para la programación de los robots construidos por los alumnos. En este caso utilizamos xLOGO [12], un intérprete de LOGO escrito en JAVA de distribución libre.



Figura 5: La Robot MARGARITA

M1	M2	MOVIMIENTO
↑	↑	avanza
↓	↓	retrocede
↑	↓	giraderecha
↓	↑	giraizquierda
—	—	alto

Tabla 1: Repertorio de Movimientos

El Lenguaje LOGO

Desde la década de los años 60 del siglo pasado, mucho antes de que la frase “pensamiento computacional” [13] se pusiera de moda, Seymour Papert [14][15] del MIT ya utilizaba el lenguaje LOGO para enseñar a los niños a pensar computacionalmente, mediante el uso de instrucciones que hacían que una tortuga realice dibujos de distintas formas geométricas [16][17]. Por ejemplo, en el Listado 1 se muestra un programa en LOGO que utiliza un robot para seguir una línea negra sobre un fondo blanco, uno de los cinco problemas clásicos de la RE.

Así también, con este lenguaje se implementó un simulador, que permite a los alumnos verificar la corrección de sus

programas, antes de proceder a grabarlo en la placa de control del robot.

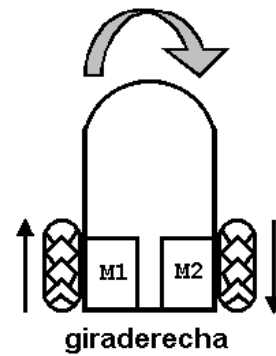


Figura 6: Movimiento Gira Derecha

```

para seguidor.de.líneas
# Blanco = 0 – Negro = 1
repitesiempre [
  mientras [sensor.reflex.1 = 0]
    [giraderecha 1]
  mientras [sensor.reflex.2 = 0]
    [giraizquierda 1]
  avanza 15
  espera 1
]
fin
  
```

Listado 1: Programa en LOGO

La Programación Visual

Como ya dijimos anteriormente, existe un consenso generalizado de la importancia de la enseñanza de la Programación en la escuela [18]. Con el objetivo de difundir el pensamiento computacional mediante la enseñanza de la Programación, y facilitar su aprendizaje, tanto en los niños como en usuarios adultos sin conocimientos previos de lenguajes de programación, recientemente ha surgido un creciente interés por los lenguajes de programación visual (LPV).

Esta clase de programación [19][20][21] se refiere al desarrollo del software que

emplea y manipula dinámicamente elementos visuales (como bloques, íconos, gráficos, etc.) de manera parcial o total. De esta manera, los programas son creados con objetos visuales que representan distintas instrucciones del programa, mediante la combinación de los mismos como si fueran piezas de un rompecabezas.

TIPO	EDADES	SOPORTE
Programación Tangible	3 a 7 años	Figuras
		Fichas
		Cubos
Programación Visual	8 a 12 años	TortugArte
		Scratch
		Blockly
Programación Textual	13 a 17 años	Pascal
		C
		Python

Tabla 2: Tipos de Programación

Por lo tanto, se evita la frustración que produce en los alumnos novicios (que no saben programar), los errores de sintaxis que se presentan en los lenguajes de programación textuales, porque cualquier pequeño error en el código (como la falta de una coma, por ejemplo), puede hacer que el programa realizado no funcione correctamente. También permite, debido a que los bloques mencionados se encuentran agrupados por categorías de acuerdo a su función, que los programadores puedan encontrar el bloque que necesitan en lugar de tener que recordar el nombre de la instrucción, eliminando de esa manera otro problema que se presenta con la programación textual

A pesar de lo expuesto anteriormente, en muchos casos los LPV no son suficientes para aquellas aplicaciones con un elevado nivel de complejidad, teniendo que optar por lenguajes de programación textuales (los cuáles suelen ser más potentes). Con respecto a esto, en la Tabla 2 se muestra

una clasificación de los tipos de lenguajes, según las edades de los alumnos.

Blockly

Blockly es un framework de código abierto basado en JavaScript que se ejecuta en un navegador web, que fue creado por un equipo liderado por Neil Fraser de “Google for Education”, que permite crear editores de programación visual [22][23].

Un programa en Blockly se construye mediante la conexión entre los distintos tipos de bloques, que representan las diferentes instrucciones del programa. Ellos están agrupados por categorías en el menú de bloques (*toolbox*) de acuerdo a su función, y deben ser arrastrados y depositados en el área de trabajo (*workspace*), tal como si fueran las piezas de un rompecabezas.

Una vez que se terminó de construir el programa, éste es convertido a código textual en cinco lenguajes (JavaScript, Python, PHP, Lua y Dart), pero uno puede definir la conversión a cualquier otro lenguaje. En nuestro caso, nosotros necesitábamos generar código textual en lenguaje JAL, un lenguaje de programación de alto nivel libre, que genera código para los microcontroladores PIC) por lo que tuvimos que implementar la correspondiente traducción.

Para la definición de los bloques, esto es, cuál será su forma (cómo será mostrado), si tendrá entradas o salidas, y cómo interactuará con los otros bloques en el área de trabajo, se utiliza el formato de intercambio de datos JSON (acrónimo en inglés de *JavaScript Object Notation*). En el Listado 2 se muestra la definición de un bloque, y en la Figura 7 el bloque resultante.

Y para generar el código en un determinado lenguaje textual, que estará asociado al bloque previamente definido, deberemos escribir el código a generar en el lenguaje

elegido. Esta tarea es relativamente complicada, porque la traducción debe ser siempre sintácticamente correcta, por lo que requiere de un buen conocimiento del lenguaje usado para dicha traducción. Como ejemplo del bloque definido anteriormente, en el Listado 3 se muestra el código que genera dicha traducción.

```
{
  "type": "mov_gd",
  "message0": "giraderecha %1 %2",
  "args0": [{
    "type": "input_dummy",
    "align": "CENTRE"
    { "type": "field_angle",
      "name": "ANGULO",
      "angle": 0
    }
  }],
  "inputsInline": true,
  "previousStatement": null,
  "nextStatement": null,
  "colour": 120,
  "tooltip": "Robot gira a la derecha",
  "helpUrl": ""
},
```

Listado 2: Definición de Bloque



Figura 7: Bloque Obtenido

```
Blockly.JavaScript ['mov_gd'] =
function(block) {
  var angulo =
  block.getFieldValue('ANGULO');
  var codigo =
  'giraderecha(' + angulo + ')\n';
  return codigo;
};
```

Listado 3: Código del Bloque

El Editor MARblock

Para poder obtener a partir del programa fuente desarrollado por los alumnos en LOGO, un programa escrito en el lenguaje JAL [24], que una vez compilado permitirá obtener el archivo en el lenguaje del microcontrolador PIC para su grabación en el mismo, se desarrolló un editor de programación visual basado en Blockly que denominamos **MARblock**, cuya interfaz se muestra en la Fig. 8. En ella se observa el clásico programa “Hola mundo”, que en este caso enciende y apaga un diodo LED de manera intermitente.

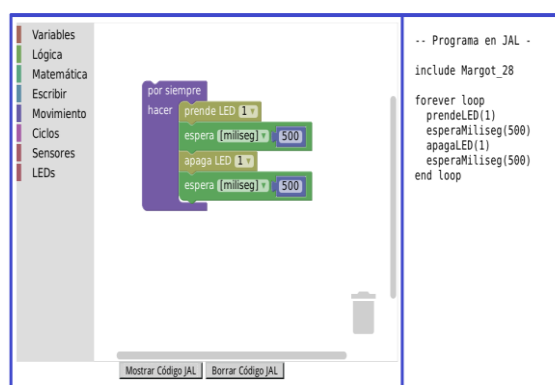


Figura 8: El Editor Visual MARblock

Este editor permite encapsular (y de esta manera ocultar) los detalles de bajo nivel de la placa de control, permitiendo que los alumnos trabajen solamente con elementos que poseen un alto nivel de abstracción (los bloques), permitiéndoles de esta manera concentrarse únicamente en la solución del problema planteado. Por ejemplo, la orden **giraderecha** involucra ordenar al motor izquierdo (M1) que gire en un determinado sentido (horario por ejemplo), al derecho (M2) girar en el sentido contrario (antihorario) y durante un determinado tiempo, de acuerdo al valor del ángulo a girar (Fig. 6). Pues bien, todos esos detalles están definidos en la librería **Margot_28**, y son transparentes a los alumnos.

Por ejemplo, a partir del fragmento de código en LOGO que se muestra en el Listado 4, obtenemos la correspondiente combinación de bloques que se muestran en la Fig. 9.

repite 4 [giraderecha 90]

Listado 4: Código en LOGO



Figura 9: Programa en Bloques

```
for 4 loop
  giraderecha(90)
end loop
```

Listado 5: Código en JAL

Resumiendo: una vez terminado el programa en bloques, el editor MARblock lo traduce a código en JAL. Finalmente, este código es copiado y pegado en la IDE de dicho lenguaje, para su compilación final. De esta manera obtenemos un archivo binario (cuya extensión es **.hex**), el cual podrá ser grabado en el microcontrolador de la placa de control.

En la Fig. 10 se muestra el programa obtenido con MARblock, que es equivalente al programa en LOGO del robot seguidor de líneas del Listado 1. Y una vez construido el programa en bloques, el editor se encarga de generar el código textual del mismo, pero en lenguaje JAL (Listado 6).

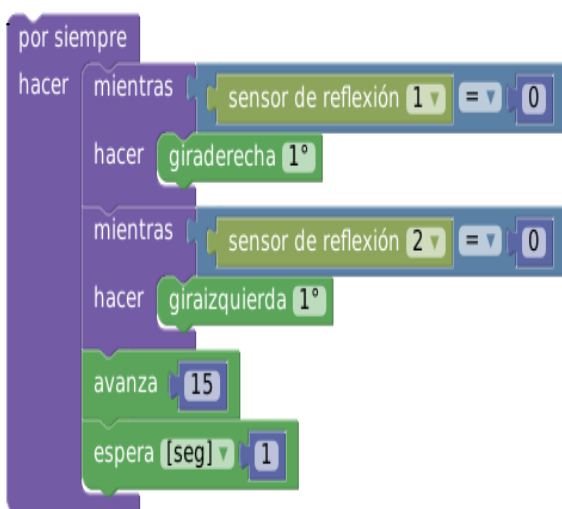


Figura 10: Seguidor de Líneas (Bloques)

```
-- Programa en JAL --
include Margot_28

forever loop
  while (sensorReflexion(1) == 0) loop
    giraderecha(1)
  end loop
  while (sensorReflexion(2) == 0) loop
    giraizquierda(1)
  end loop
  avanza(15)
  esperaSeg(1)
end loop
```

Listado 6: Seguidor de Líneas (JAL)

Conclusiones

En este trabajo se presenta la metodología utilizada por el LABRA en los Talleres de Robótica Educativa realizados en diversas instituciones educativas del nivel primario de la ciudad de Orán (Salta), desde el año 2013 hasta el 2017.

Consideramos que la RE puede transformarse en un motor de innovación en la escuela, un elemento superador y disruptivo, pues sus características intrínsecas permiten insertar cambios relevantes en las formas de enseñar y aprender de los alumnos, contando con la posibilidad concreta de consolidarse e incorporarse como una práctica regular y cotidiana en los procesos de enseñanza.

Referencias

- [1] *STEM Education Coalition*
[http:// www.stemedcoalition.org](http://www.stemedcoalition.org)
- [2] Maza R. D., *Robótica Educativa para Aprender a Pensar*, Editorial EUNSa., Salta, 2014.-
- [3] Maza R. D., *Robótica Educativa para una Nueva Escuela*, Editorial EUNSa., Salta, 2018.-
- [4] Maza R. D., Méndez E. A., Torres J. A., Mamani G. A. *Taller de Robótica*

- en la Escuela*, 1ras. Jornadas Nacionales de TIC e Innovación en el Aula, La Plata, 2011.-
- [5] *Alumnos de Orán, entre los mejores en Tecnópolis*
<http://informatosalta.com.ar/noticia/135994/alumnos-de-oran-entre-los-mejores-en-tecnopolis>
- [6] Pinto Salamanca M. L., Barrera Lombana N., Pérez Holguín W. J., *Uso de la Robótica Educativa como herramienta en los procesos de enseñanza*, Ingeniería, Investigación y Desarrollo, Vol 10, Nro 1, Universidad Pedagógica de Colombia, 2010.-
- [7] Nourbakhsh I. R., Crowley K., Wilkinson K., Hammer E., *The educational impact of the Robotic Autonomy mobile robotic course*, Technical Report CMU-RI-TR-03-18, Carnegie Mellon University, Pittsburgh, 2002.-
- [8] Perkins D., Tishman S., Jay E., *Un Aula para Pensar*, Aique Grupo Editor, Buenos Aires, 2006.-
- [9] Pogré P., Lombardi G., *Escuelas que Enseñan a Pensar*, Papers Editores, Buenos Aires, 2004.-
- [10] Kuo B. C., *Sistemas de Control Automático*, Pearson – Prentice Hall, 7ma. Edición, México, 1996.-
- [11] Ogata K. *Ingeniería de Control Moderno*. Prentice Hall, 3ra. Edición, Madrid, 1998.-
- [12] *xLOGO*
<http://xlogo.tuxfamily.org>
- [13] Wing J., *Computational Thinking*, Comm. of the ACM, Vol. 49, No. 3, Marzo 2006.-
- [14] Papert S., *Desafío a la Mente*, Ediciones Galápagos, Buenos Aires, 1987.-
- [15] Papert S., *Mindstorm: Children, Computers, and Powerful Ideas*, Basic Books, 1980.-
- [16] Abelson H., diSessa A., *Turtle Geometry: the Computer as a Medium for Exploring Mathematics*, MIT Press, 1981.-
- [17] Reggini H. C., *Alas para la Mente*, Ediciones Galápagos, 2da. Edición, Buenos Aires, 1983.-
- [18] *Code.org: Anybody can learn*
<https://code.org>
- [19] Begel A., *LogoBlocks: A graphical programming language for interacting with the world*, Electrical Engineering and Computer Science Department, MIT, Boston, MA, 1996.-
- [20] Bau D., Bau A., Dawson M, Pickens C. S., *Pencil Code: Block code for a text world*, 14th International Conference on Interaction Design and Children, 2015.-
- [21] Monig J., Ohshima Y., Maloney J., *Blocks at your fingertips: Blurring the line between blocks and text in GP*, IEEE Blocks and Beyond Workshop, Oct. 2015.-
- [22] *Google Blockly*
<https://developers.google.com/blockly>
- [23] Fraser, N., *Ten things we've learned from Blockly*. Blocks and Beyond Workshop, IEEE, 2015.-
- [24] *Just Another Language (JAL)*
<http://justanotherlanguage.org>