

# Simulador para robots basados en Arduino

Cristian González García  
Escuela de Ingeniería Informática  
Universidad de Oviedo  
gonzalezcristian@uniovi.es

Ana I. Gómez  
Escuela de Ingeniería Informática  
Universidad Rey Juan Carlos  
ana.gomez.perez@urjc.es

Domingo Gómez  
Facultad de Ciencias  
Universidad de Cantabria  
gomezd@unican.es

## Resumen

En un contexto de clase, el desarrollo de software sobre plataformas hardware, aunque es una herramienta pedagógica atractiva, presenta ciertos inconvenientes. En ocasiones el hardware puede ser defectuoso, estar mal configurado o ensamblado y/o realizarse un uso no esperado o correcto de éste. Otras veces, los problemas son externos, por ejemplo cuando se dispone de unidades limitadas de los robots de clase, o hay restricciones horarias o de espacio físico para su uso. El uso complementario de simuladores ayuda a reducir riesgos al permitir comprobar y simular la ejecución de código sin requerir el hardware. Sin embargo, no existen suficientes herramientas docentes maduras que trabajen en lenguajes de bajo nivel, o bien, pueden no adaptarse a los ejercicios propuestos para robots con diseños propios. Por todo esto, se presenta una herramienta de simulación de robots basados en Arduino, con finalidad de ser fácilmente configurable a distintos tipos de ejercicios y configuraciones de sensores basados en Arduino. Esta herramienta ha sido inicialmente propuesta para la asignatura "Software para Robots", por lo que permite la simulación y programación de los robots utilizados en dicha asignatura. Se ha realizado una experiencia docente conjunta entre tres universidades en los que los estudiantes han realizado distintas tareas. Los comentarios de los estudiantes indican un alto grado de satisfacción en su uso. Además, presenta otras ventajas como el trabajo autónomo en casa, la detección rápida de errores hardware mediante la comprobación del funcionamiento del código en el simulador, y el beneficio de su uso a los estudiantes con evaluación diferenciada.

## Abstract

In a classroom context, the development of software on hardware platforms, although it is an attractive pedagogical tool, presents certain drawbacks. Sometimes the hardware may be defective, be poorly configured or assembled and/or be used unexpectedly or incorrectly. Other times, the problems are external, for example when there are limited units of class robots,

or there are time restrictions or physical space for their use. The complementary use of simulators helps reduce risks by allowing code execution to be tested and simulated without requiring the hardware. However, there are not enough mature teaching tools that work in low-level languages, or they may not adapt to the exercises proposed for self-designed robots. For all this, an Arduino-based robot simulation tool is presented, in order to be easily configurable to different types of exercises and sensor configurations. This tool has been initially proposed for the subject "Software for Robots", so it allows the simulation and programming of the robots used in said subject. A joint teaching experience has been carried out between three universities in which students have carried out different tasks. Student feedback indicates a high degree of satisfaction with its use. In addition, it presents other advantages such as autonomous work at home, rapid detection of hardware errors by checking the operation of the code in the simulator, and the benefit of its use for students with differentiated evaluation.

## Palabras clave

Arduino, simulador, software, robótica, recurso docente, prácticas

## 1. Introducción

El uso de simuladores para la formación de los estudiantes presenta numerosas ventajas dado que permite mostrar un modelo computerizado de un sistema que permite experimentar y simular situaciones reales en un entorno controlado para asimilar los conceptos impartidos. No solo tienen un impacto en el aprendizaje de los estudiantes según lo observado por Dale [4], sino que debido a que permite el uso fuera de clase los estudiantes pueden utilizarlos de forma autónoma para reforzar su aprendizaje o en contextos en los que no puede acudir a clase por condiciones como evaluación diferenciada o dispensa académica. Un estudio reciente [9] muestra además que para la adquisición de la

habilidad para la descomposición de problemas, es indiferente utilizar robots o simuladores. Además, el uso de simuladores se ha demostrado exitoso para simular redes [2] o en sistemas operativos[5].

Por otra parte, el uso de la robótica educativa para la enseñanza de la programación proporciona más diversión y esto influye positivamente en la adquisición de las habilidades de pensamiento computacional de los estudiantes [9]. Las conclusiones del anterior artículo han sido corroboradas en nuestras experiencias docentes anteriores y sabemos que la programación en dispositivos basados en Arduino despierta un gran interés en estudiantes de grados de informática cuyos planes de estudio no incluyen una asignatura obligatoria sobre programación de robots en aula. Aunque el hardware robótico real presenta costes y riesgos en el aula, se ha ido incrementando su presencia para la enseñanza de la robótica en distintos niveles educativos [1, 7].

Actualmente, existen distintos programas en marcha que tratan de acercar la robótica a los estudiantes. Un ejemplo es AI Plus <sup>1</sup>, un proyecto europeo que introduce la inteligencia artificial y la robótica, como campo fundamental para su aplicación, en el currículum de enseñanzas secundarias en estudiantes de secundaria. Dentro de las enseñanzas universitarias, se pueden destacar herramientas como R2P2 [3], un simulador de robots en espacios bidimensionales para el aprendizaje de algoritmos de inteligencia artificial, Unibotics [8] una colección de problemas basada en el simulador de robots Gazebo o Cnosos [6], una biblioteca para la simulación de algoritmos clásicos de grafos sobre coches robot basados en Arduino.

Otros ejemplos son Scratch for Lego Mindstorm <sup>2</sup> y MiniBloq <sup>3</sup> y Scratch for Arduino <sup>4</sup>, que tienen lenguajes basados en piezas de puzzle para programar la lógica. No obstante, en las asignaturas, al ser pertenecientes a grados de informática, se busca que aprendan y mejoren su programación. Además, estos lenguajes están limitados frente a la programación, así como en el número de sensores y actuadores que soportan y la carencia de los robots utilizados. Otros ejemplos son editores como BitBloq <sup>5</sup> y Tinkercad <sup>6</sup>, que tienen lenguajes específicos de dominio (DSL) gráficos y también permiten programar libremente, pero no tienen los sensores y actuadores utilizados, así como carecen de robots.

En este trabajo proponemos un recurso software educativo abierto, que pueda ser utilizado en combinación con dispositivos o pueda ser utilizado independientemente. Además, esta herramienta promueve el

<sup>1</sup><https://aipudc.azurewebsites.net/scope/>

<sup>2</sup><https://scratch.mit.edu/ev3>

<sup>3</sup><http://blog.minibloq.org/>

<sup>4</sup><http://s4a.cat/>

<sup>5</sup><https://bitbloq.cc/>

<sup>6</sup><https://www.tinkercad.com/>

trabajo autónomo, permite a los estudiantes probar sus implementaciones, corregirlas y llevarlas a un robot físico. La mayor ventaja consiste en poder realizar el siguiente flujo de trabajo: el alumnado puede probar la ejecución de su propio código fuente en casa o en otros sitios diferentes al laboratorio de prácticas, mostrándole una simulación bastante fiel a la realidad sobre como funcionaría en el robot seleccionado. De esta forma, en la clase se pueden dedicar a probar su algoritmo desarrollado y ajustarlo sobre los robots con ayuda del profesor.

Durante el presente curso, este simulador ha dado soporte a dos asignaturas de diferentes universidades, que además tratan temáticas diferentes, pero tienen en común el uso de robots basados en Arduino. Estas son Software para Robots de la Universidad de Oviedo y Diseño de Algoritmos de la Universidad de Cantabria. Se ha introducido la interacción y sincronización entre ambas, haciendo uso del mismo software para facilitar el mantenimiento y ampliación de éste. Finalmente, se ha evaluado su uso por parte de los estudiantes de una tercera asignatura y universidad, Interacción Persona-Ordenador de la Universidad Rey Juan Carlos.

## 2. Descripción de la solución propuestas

La pantalla principal de la herramienta desarrollada se muestra en la figura 1. Esta herramienta soporta dos tipos de robots: un robot móvil y un actuador lineal. En el caso del robot móvil, cuenta con un total de 6 simulaciones diferentes, 2 simulaciones para 3 configuraciones diferentes (2, 3 o 4 infrarrojos).

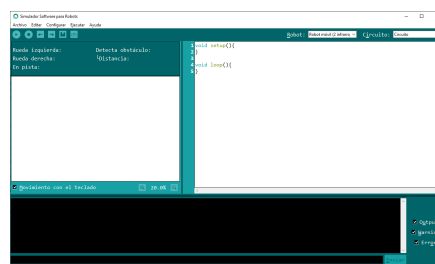


Figura 1: Pantalla principal del simulador

La configuración de los controles es la siguiente: arriba a la izquierda se muestran las listas desplegables opciones para importar, exportar y abrir archivos de código en *Archivo*. *Editar* contiene deshacer y rehacer. El menú opción *Configurar* permite configurar los pines y los botones, con un «pop-up» adaptado al tipo de robot. *Ejecutar* permite ejecutar y detener el código fuente, y ampliar y reducir el zoom del simulador. *Ayuda* contiene el Manual de Ayuda y el Acerca De.

Debajo de estas listas están los botones *Ejecutar*, *Stop*, *Deshacer*, *Rehacer*, *Guardar* y *Cargar*. El área blanca de la derecha es la zona que muestra el código fuente. El simulador soporta Wiring, que es el lenguaje utilizado para programar en Arduino <sup>7</sup>, y simula diferentes librerías de Arduino para ejecutar el simulador. Algunas librerías de dicho lenguaje están escritas en C o C++. Este zona tiene resaltado de sintaxis, y por defecto muestra lo mismo que el IDE de Arduino, los métodos principales de trabajo: *Setup* y *Loop*. El área negra inferior es la consola, donde se muestra la salida del código fuente (output), «warnings» y errores, y a su derecha hay tres «checkbox» que permiten realizar un filtrado según este tipo de información.

Los «combobox» en la parte superior derecha permiten seleccionar el robot, entre el actuador lineal y el robot móvil (de 2, 3 o 4 infrarrojos). En el último caso, hay un segundo «combobox» que permite seleccionar el circuito (circuito, laberinto, recta, obstáculo, recta y obstáculo). Según lo que se seleccione en dichos combobox, mostrará una cosa u otra en el área blanca de la izquierda.

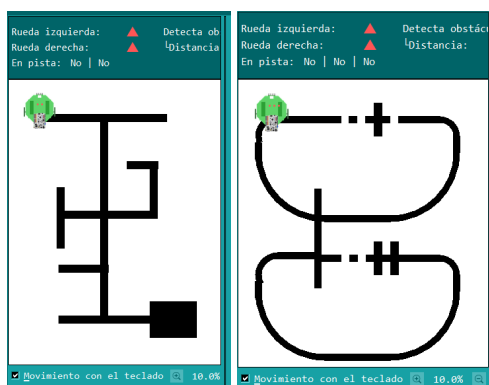


Figura 2: Simulación del robot móvil de dos sensores en el circuito del laberinto (izquierda) y tres sensores sobre circuito de nodos con etiqueta (derecha)

La figura 2 muestra el simulador del robot móvil en el laberinto (izquierda) y en un circuito de nodos (derecha). En la parte superior se muestra información del robot que facilita su visualización, como es el sentido de giro de las ruedas y si están en marcha, si está detectando un obstáculo en frente de él usando un sensor de ultrasonidos, así como la distancia con el obstáculo, y que detectan los sensores infrarrojos inferiores que se utilizan para seguir los circuitos (si está sobre una línea negra muestra «Sí» o si no hay línea debajo de los sensores muestra «No»). En la parte inferior, tiene el «checkbox» que permite controlar el robot mediante teclado o mediante código fuente, y botones para hacer zoom. La figura 3 muestra la simulación del robot móvil sobre un circuito cerrado con la representación

<sup>7</sup><https://www.arduino.cc/>

de un obstáculo y sobre ésta la misma información del caso anterior. Finalmente, la figura 4 muestra la simulación del robot lineal, que cambia la información superior por la dirección del movimiento, y si los sensores de final de línea están pulsados o no, y añade en su sección inferior el control del Joystick en los ejes X e Y, y su botón.

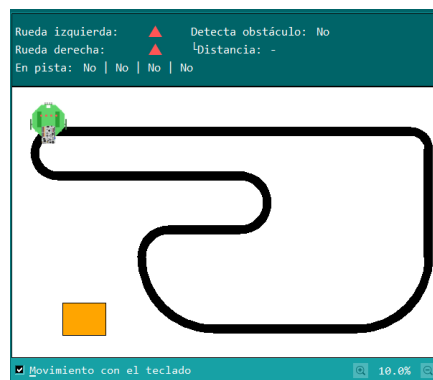


Figura 3: Simulador circuito y obstáculo

Este simulador cuenta con licencia GNU General Public License v3.0. El código fuente Python<sup>8</sup> del programa <sup>9</sup> está acompañado con un manual del usuario en «manual-usuario.pdf». La aplicación actualmente cuenta con soporte en Windows y tiene un script (build.py) para crear el ejecutable.

### 3. Conclusiones

Se presenta un simulador abierto para robots basados en Arduino, que permite simular el código fuente y mostrar lo que se ha programado sobre los robots simulados. Esta herramienta es novedosa pues no se

<sup>8</sup><https://www.python.org/>

<sup>9</sup><https://github.com/gonzalezgarciaacristian/simulador-robotic-software>

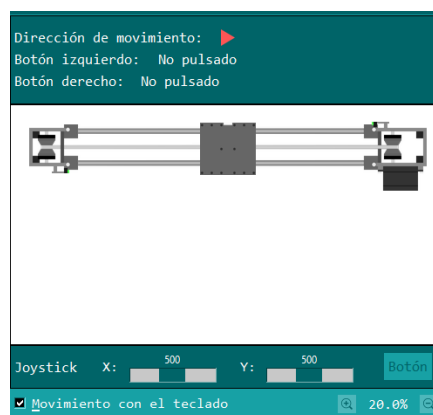


Figura 4: Simulador del actuador lineal

encuentran herramientas de este estilo en la literatura, debido a que los robots son creados específicamente para la asignatura utilizando impresoras 3D y muchas herramientas existentes son para robots profesionales que cuestan centenas o miles de euros, o bien, a pesar de ser más baratas no se adaptan a los robots utilizados.

Los datos preliminares de su utilización en el último curso constatan el ahorro del tiempo tanto para el profesor como para el estudiante, pues permite saber el origen software o hardware de los fallos en el código, ya que la simulación es próxima a la realidad. Además, permite a los estudiantes trabajar de forma autónoma, cuando antes debían reservar un aula específica en la escuela universitaria para poder utilizar los robots. Esto es tanto un gran beneficio para ellos, sobre todo en estudiantes de evaluación diferenciada, como para el profesor a la hora de depurar fallos y detectar robots defectuosos. Finalmente, se ha utilizado en diferentes asignaturas de diferentes universidades, con el propósito de dar un mayor soporte a profesor y estudiantes que utilicen robots basados en Arduino y crear una herramienta de código abierto para ellos que sea fácilmente adaptable a sus asignaturas.

Durante el paso del curso, se habló con los estudiantes y se les preguntó su opinión a través de un cuestionario anónimo realizado en Google Forms. Debido a que el simulador no soporta todas las prácticas de la asignatura, pudieron hacer una comparativa entre la ayuda que este les ofrece. También había algún alumno repitiendo curso, o que había trabajado con Arduino en el instituto. El simulador les pareció sencillo e intuitivo, fácil de utilizar, ha mejorado la asignatura, y les ofreció gran ayuda en discernir si el código funciona o no. Durante las sesiones, también se han reportado fallos de accesibilidad, usabilidad y «bugs», de los cuáles algunos han sido ya reparados en la versión actual del código.

Como trabajo futuro se desea ofrecer soporte a la configuración de distintos modelos de robots y circuitos sin el requisito de codificación, pues actualmente hay que modificar un fichero en JSON para crearlos de forma manual. Asimismo, se plantea la inclusión un simulador de Arduino junto a los diferentes sensores utilizados en clase para permitir la simulación de circuitos. De esta forma, se planea unificar el uso en una misma herramienta que dé soporte dentro de asignaturas enfocadas al aprendizaje de programación de sistemas embebidos. También se añadirá el soporte para comunicarse directamente con la placa Arduino y así subir el software desde el propio simulador. Finalmente, se encuentra en desarrollo una extensión para la evaluación del aprendizaje de los estudiantes mediante el estudio de los datos extraídos del uso del simulador, de forma que se puedan detectar posibles puntos en los que los estudiantes encuentren dificultades.

## 4. Agradecimientos

Este proyecto forma parte de la convocatoria 2022/2023 de certificación de proyectos de innovación docente de la Universidad de Oviedo. A Diego Fernández Suárez como autor del TFG con el prototipo inicial bajo dirección de Cristian González García.

## Referencias

- [1] Francisco Bellas, Martín Naya, Gervasio Varela, Luis Llamas, Abraham Prieto, Juan Carlos Becerra, Moisés Bautista, Andrés Faiña, y Richard Durro. The robobo project: Bringing educational robotics closer to real-world applications. En *Robotics in Education*, pp. 226–237, 2018.
- [2] Jesús Cabero-Almenara, Julio Costas. La utilización de simuladores para la formación de los alumnos. *Prisma Social*, 2016.
- [3] Mario Cobos, María D. R-Moreno, y David F. Barrero. R2p2: Un simulador robótico para la enseñanza de inteligencia artificial. En *Actas de las XXVI Jornadas sobre Enseñanza Universitaria de la Informática*, pp. 285–292, 2020.
- [4] Edgar Dale. *Audiovisual methods in teaching*. ERIC, 1969.
- [5] Adelaida Delgado Domínguez y Jaume Aloy Vich. Simuladores para la docencia de algoritmos de asignaturas de sistemas operativos. En *Actas de las XXVIII Jornadas sobre la Enseñanza Universitaria de la Informática*, pp. 39–46, 2022.
- [6] Ana I Gómez, Domingo Gómez, y Lara González de Cos. Cnosos: una biblioteca para enrutamiento en arduino car kits. En *Actas de las XXVIII Jornadas sobre la Enseñanza Universitaria de la Informática*, pp. 367–370, 2022.
- [7] Cristina Pardo. Experiencia didáctica con arduino y tinkercad: metodología «learning by doing» aplicada a estudiantes universitarios del itinerario de tecnología del máster en profesorado. En *Actas de las XXVIII Jornadas sobre la Enseñanza Universitaria de la Informática*, pp. 175–182, 2022.
- [8] David Roldán-Álvarez, Sakshay Mahna, y José M Canas. A ros-based open web platform for intelligent robotics education. En *Robotics in Education: RiE 2021 12*, pp. 243–255. Springer, 2022.
- [9] Fang Chuan Ou Yang, Hui Min Lai, y Yen Wen Wang. Effect of augmented reality-based virtual educational robotics on programming students' enjoyment of learning, computational thinking skills, and academic achievement. *Computers and Education*, 195, 4 2023.