Herramienta Educativa de Programación Visual para el Robot e-puck en Webots con Blockly

Jorge Aaron Rivas Abril
School of Computer Science
Universidad Nacional de San Agustín
Arequipa, Perú
jrivasab@unsa.edu.pe

Arthur Patrick Meza Pareja
School of Computer Science
Universidad Nacional de San Agustín
Arequipa, Perú
amezapa@unsa.edu.pe

Abstract* — Este trabajo presenta una plataforma educativa basada en la web que integra Blockly, una biblioteca de programación visual, con Webots, un entorno de simulación robótica. El objetivo es simplificar el aprendizaje en robótica permitiendo a los usuarios programar un robot virtual e-puck mediante bloques de código arrastrables. Esta propuesta busca superar barreras comunes como el alto costo y la disponibilidad limitada de robots físicos. Al combinar la programación visual con la simulación en tiempo real, la plataforma promueve el pensamiento computacional y hace que la enseñanza de la robótica sea más accesible, especialmente en entornos con recursos limitados. La herramienta propuesta actúa como un puente entre los conceptos teóricos y su aplicación práctica en la educación STEM.

I. Introducción

La enseñanza de la robótica representa una vía efectiva para fortalecer las competencias STEM, promoviendo el pensamiento lógico, la creatividad y la resolución de problemas desde edades tempranas. Sin embargo, su implementación en contextos educativos se ve frecuentemente limitada por factores como los altos costos del hardware, la necesidad de mantenimiento técnico y la disponibilidad restringida de laboratorios físicos. Estas limitaciones han sido ampliamente documentadas en estudios sobre el diseño e implementación de entornos educativos tecnológicos, incluyendo propuestas de laboratorios remotos como alternativa viable al modelo tradicional [1].

En respuesta a estos desafíos, han emergido plataformas virtuales que permiten la interacción remota con entornos experimentales. Una de las más destacadas es **LabsLand**, la cual ofrece acceso a laboratorios reales de robótica desde cualquier parte del mundo, permitiendo a los estudiantes programar mediante código o mediante bloques visuales. Esta última opción, basada en Blockly, ha facilitado el acceso a la programación a estudiantes con poca o ninguna experiencia previa, ampliando significativamente la inclusión educativa en este campo [3].

La programación visual mediante bloques ha demostrado ser una herramienta pedagógica de gran impacto. Un ejemplo emblemático es **Scratch**, desarrollado por el MIT Media Lab, que ha revolucionado la forma de enseñar programación a niños y principiantes gracias a su interfaz intuitiva y su enfoque lúdico [6]. En la misma línea, **Open Roberta Lab** ha adaptado este paradigma al ámbito de la robótica educativa, permitiendo programar sistemas virtuales con bloques personalizables y visualizar los resultados en tiempo real [9].

Para facilitar la creación de estos entornos, se han desarrollado herramientas como **Blockly**, una librería modular de Google que permite construir lenguajes visuales de programación completamente adaptables. Su estructura basada en JavaScript y su capacidad para traducir bloques en lenguajes como Python, JavaScript o Lua han convertido a Blockly en una base sólida para aplicaciones educativas interactivas [10].

En cuanto a la simulación robótica, **Webots** se posiciona como uno de los entornos más robustos y versátiles. Este software de código abierto permite programar robots virtuales utilizando múltiples lenguajes y simular escenarios complejos con precisión física, siendo ampliamente utilizado en la formación técnica y universitaria [12]. Su compatibilidad con Python lo convierte en una herramienta accesible para estudiantes, y su integración con interfaces visuales abre la puerta a nuevas formas de enseñanza remota [15].

Este trabajo propone una plataforma educativa basada en la web que combina **Blockly y Webots** para la programación del robot e-puck, eliminando la necesidad de hardware físico y ofreciendo una experiencia completamente virtual. Inspirado en proyectos como **Webots-Blockly**, desarrollado por el Storming Robots Development Group, este proyecto adapta la integración al robot e-puck mediante bloques específicos para sus sensores y actuadores, con el objetivo de facilitar la enseñanza de la robótica en entornos con recursos limitados [13].

II. Estado del arte

En la actualidad, el ámbito de la robótica y la ingeniería se beneficia enormemente de la experiencia práctica para la correcta formación de los estudiantes. Sin embargo, los laboratorios tradicionales presentan desafíos significativos, incluyendo altos costos de infraestructura, mantenimiento y supervision. [1] Estas limitaciones han impulsado el desarrollo de alternativas que permitan la experimentación y el aprendizaje de forma más accesible.

A. Laboratorios Remotos y Virtuales en Robótica

La evolución tecnológica ha propiciado la aparición de laboratorios remotos y virtuales como soluciones viables para la enseñanza de la robótica. Calvo et al. [2] clasifican los entornos experimentales según la accesibilidad a los recursos (local o remoto) y la naturaleza del sistema físico (real o virtual), como se ilustra en la Tabla I

TABLA I

Clasificación de los entornos experimentales [2].

	Real	Virtual
Local	Laboratorios presenciales con plantas reales	Laboratorios presenciales con plantas simuladas
Remoto	Teleoperación de una planta real	Laboratorio remoto con plantas simuladas

Los laboratorios remotos permiten a los usuarios interactuar con hardware real a distancia a través de internet, superando las barreras geográficas y de acceso. Un ejemplo notable es LabsLand, una plataforma que ofrece acceso a laboratorios reales de diversas disciplinas, incluyendo robótica, desde cualquier parte del mundo [3]. LabsLand proporciona entornos como el "Robot Arduino (código)", donde los usuarios programan robots reales en código Arduino y visualizan los resultados en tiempo real a través de una cámara [4]. Además, ofrece el "Robot Arduino (visual)", que utiliza una interfaz de programación por bloques, generada con Blockly, para la misma tarea, adaptándose a usuarios con diferentes niveles de conocimiento en programación [5]. Este enfoque permite ofrecer entornos experimentales tanto para usuarios avanzados como para principiantes.

B. Herramientas de Programación Visual

La programación por bloques ha emergido como una herramienta fundamental para simplificar la enseñanza de la programación y el pensamiento computacional, especialmente para principiantes y niños.

Scratch, desarrollado por el MIT Media Lab en 2007 por Mitchel Resnick y su equipo, es el referente más prominente en este campo y ha sido pionero en la programación basada en bloques [6], [7]. Su diseño intuitivo permite a los usuarios crear juegos, animaciones e historias mediante el arrastre y la conexión de bloques que representan comandos de programación. Además de la programación por bloques, Scratch permite la manipulación de sonido, vídeo e imágenes. Su gran portabilidad, que permite el intercambio de gráficos entre proyectos, y su soporte para múltiples lenguajes lo han convertido en una plataforma ampliamente adoptada globalmente [6].

Otro ejemplo significativo es el software Open Roberta Lab [9]. Originado en el proyecto Roberta en 2002 para promover el interés de los jóvenes en las áreas STEM (Science, Technology, Engineering and Mathematics) [8], Open Roberta Lab evolucionó hasta convertirse en la plataforma actual en 2014. Esta aplicación de código abierto se ejecuta en la nube, donde el usuario puede programar por bloques y simular la placa base del robot, incluyendo sus conexiones internas a los elementos deseados. También dispone de una visualización simulada de los resultados, ofreciendo una experiencia educativa integral [9].

En este contexto, Blockly, una librería de Google iniciada en 2011 y lanzada en 2012, se presenta como una herramienta versátil para la creación de interfaces de programación visual [10], [11]. Implementada en JavaScript, Blockly permite la creación y manejo de bloques que representan conceptos fundamentales de programación, como variables, expresiones lógicas, bucles y más. Su diseño modular permite que los bloques se combinen como piezas de un rompecabezas para formar programas válidos. Una ventaja clave de Blockly es su capacidad para traducir estos programas visuales a diversos lenguajes de programación textuales como Python, JavaScript, PHP, Lua y Dart. Su naturaleza de código abierto y su capacidad de personalización la hacen ideal para proyectos educativos específicos, como el que se propone en este trabajo para el control de robots, donde se han implementado bloques específicos para robótica, como el control de motores y sensores.

C. Simuladores de Robots

Dado que la interacción con robots reales puede ser costosa o inaccesible, los simuladores de robots se han convertido en una pieza clave en la educación y la investigación en robótica.

Webots, desarrollado por Cyberbotics Ltd. desde 1998, es un software de simulación de código abierto ampliamente utilizado para la programación y simulación de robots [12]. La versión R2021a fue utilizada en este proyecto. Webots está diseñado para funcionar en Windows, Linux y macOS, utilizando Qt para la interfaz gráfica, Open Dynamics Engine (ODE) como motor de física y OpenGL 3.3 para el renderizado. Permite la creación de escenas complejas, la importación de modelos 3D y la programación de robots en múltiples lenguajes, incluyendo Python, Java, MATLAB, Robot Operating System (ROS), C o C++. Webots proporciona un entorno realista donde los algoritmos de control pueden ser probados y validados antes de su implementación en hardware físico. Este proyecto aprovecha las capacidades de Webots para simular el comportamiento del robot e-puck en diversos escenarios experimentales.

Para este proyecto, el lenguaje de programación Python ha sido seleccionado para el control de los robots en Webots [15]. Python, creado por Guido Van Rossum, es un lenguaje de alto nivel con una semántica fácil de leer, lo que lo hace popular para desarrollo web, inteligencia artificial y análisis de datos.

D. Proyectos Relacionados e Integración Blockly-Webots

La integración de herramientas de programación visual con simuladores de robots no es un concepto nuevo y representa una línea de investigación activa en la robótica educativa. El proyecto Webots-Blockly de Storming Robots Development Group [13], [14], ha servido como inspiración fundamental para este trabajo. Este repositorio explora la fusión de la librería Blockly para la interfaz de programación y el software Webots para la simulación, sentando las bases para el desarrollo de aplicaciones educativas interactivas en robótica. Este proyecto actual se inspira en gran medida en esa integración, desarrollando una aplicación web que permite al usuario programar un robot de forma remota (simulada) mediante bloques, adaptándose a las características específicas de cada robot y sus elementos (sensores de distancia, bumpers, encoders, sensores de color, GPS, giroscopio).

III. METODO PROPUESTO

En la Figura 1 se presenta una representación gráfica del método propuesto para la creación de una plataforma

educativa basada en la web, la cual permite la programación visual del robot e-puck en el entorno de simulación Webots, mediante el uso de la librería Blockly. Este método está compuesto por cinco etapas fundamentales: análisis de requisitos, diseño del entorno, integración de tecnologías, implementación de bloques personalizados, y pruebas de funcionamiento.

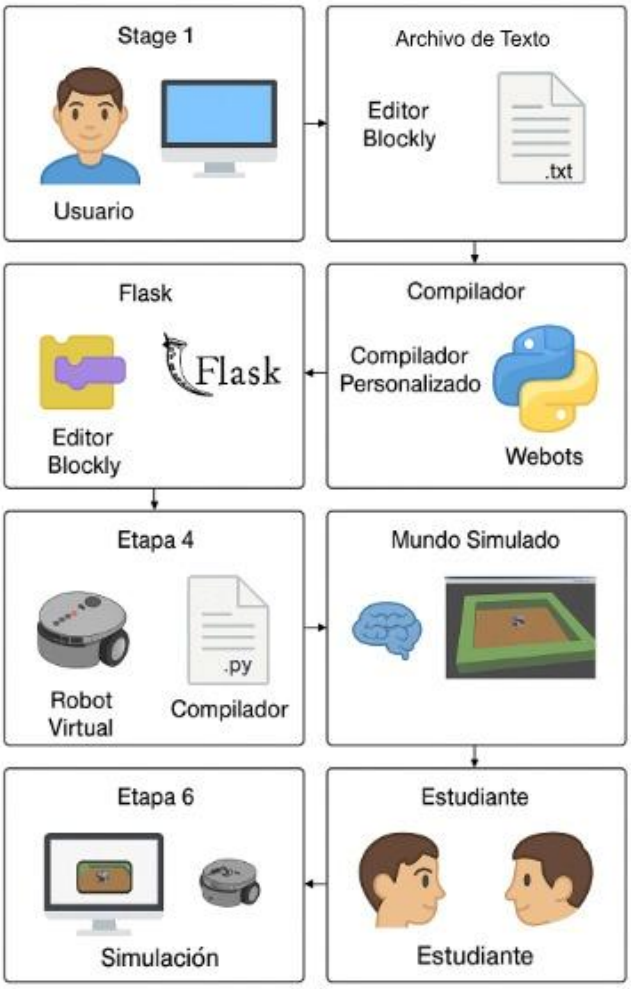

Figura 1. Método propuesto para la integración de Blockly y Webots.

A. *Análisis de Requisitos*

Esta etapa consistió en la identificación de los elementos esenciales para lograr una interacción efectiva entre el usuario y el entorno de simulación. Se establecieron tres tipos de requisitos principales. En primer lugar, los requisitos funcionales incluyeron el desarrollo de una interfaz amigable orientada a estudiantes, la capacidad de cargar y ejecutar programas, así como la visualización en tiempo real del robot e-puck [12]. En segundo lugar, los requisitos tecnológicos contemplaron la compatibilidad con navegadores modernos y el soporte para la comunicación entre JavaScript (utilizado por Blockly) y Python (utilizado por Webots) [10], [12], [15]. Finalmente, desde el punto de vista educativo, se priorizó la facilidad de uso para usuarios sin experiencia previa en programación textual, junto con el soporte para conceptos clave de la robótica educativa como sensores, motores y lógica de control [1], [2].

B. *Diseño del Entorno*

El diseño del entorno se estructuró en dos módulos principales. El primero, denominado Módulo Blockly, corresponde a la interfaz visual basada en bloques arrastrables y conectables, implementada utilizando la librería Blockly de Google [10]. En este módulo se definieron bloques personalizados para representar comandos específicos del robot e-puck, como por ejemplo mover, girar o leer sensores de distancia [13]. El segundo módulo corresponde a Webots, un entorno de simulación donde se carga un modelo virtual del robot e-puck. Este entorno es responsable de simular el comportamiento físico del robot en respuesta a los comandos generados desde la interfaz visual [12].

C. *Integración de Tecnologías*

La conexión entre la interfaz visual y el entorno de simulación se llevó a cabo mediante una arquitectura cliente-servidor. El cliente, o *front-end*, ejecuta la interfaz basada en Blockly, genera el código correspondiente en Python y lo envía al servidor. El servidor, o *back-end*, fue desarrollado utilizando Flask, un microframework de Python que permite recibir el código generado, interpretarlo mediante la API de Webots y ejecutar la simulación correspondiente [13].

Esta integración permitió mantener una estructura sencilla y funcional del sistema, facilitando la interacción entre el entorno de programación visual y el simulador sin requerir hardware físico ni comunicación en tiempo real.

D. *Implementación de Bloques Personalizados*

Se desarrollaron bloques visuales con funcionalidades específicas para el control del robot e-puck. Entre estos se incluyen bloques de movimiento básico, como avanzar, retroceder y girar; bloques para la lectura de sensores, como sensores de distancia, contacto y giroscopio; y bloques correspondientes a estructuras de control, incluyendo condicionales y bucles [11], [13]. Cada bloque fue definido con su representación visual correspondiente, así como con su traducción directa al lenguaje Python, manteniendo la coherencia con la API proporcionada por Webots para el modelo e-puck [12], [15].

E. *Pruebas y Validación Funcional*

La herramienta fue sometida a pruebas en distintos escenarios de simulación con el objetivo de validar la correcta traducción de los bloques visuales a código Python, así como la ejecución efectiva de dicho código en el entorno Webots [13]. Las pruebas incluyeron la participación de estudiantes sin experiencia previa en programación textual, lo que

permitió evaluar tanto la facilidad de uso de la herramienta como su capacidad para permitir la realización de tareas básicas, tales como evitar obstáculos o seguir una línea dentro del entorno simulado [1], [7].

REFERENCES

[1] E. G. García, "Diseño de un laboratorio robótico remoto para la enseñanza," 2011. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] I. Calvo, E. Zulueta, U. Gangoiti, y J. M. López, "Laboratorios remotos y virtuales en enseñanzas técnicas y científicas," Ikastorratza. e-Revista de didáctica, 2008.

[3] LabsLand, "LabsLand." [En línea]. Disponible en: https://labsland.com/es.

[4] LabsLand, "Robot arduino (código)." [En línea]. Disponible en: https://labsland.com/es/labs/arduino-robot

[5] LabsLand, "Robot arduino (visual)." [En línea]. Disponible en: https://labsland.com/es/labs/arduino-visual-robot

[6] M. Resnick, J. Maloney, N. Rusk, E. Eastmond, A. Millner, J. Silver, y A. Blanton, "Scratch," Lifelong Kindergarten group, MIT Media Lab. [En línea]. Disponible en: http://info.scratch.mit.edu, 2003.

[7] Scratch, “Scractch.” [En linea] Disponible en : https://scratch.mit.edu/

[8] K. Olaskoaga, "Open roberta lab, una alternativa a ev3-g," 2021. [En línea]. Disponible en: https://www.hispabrickmagazine.com/pdfs/HBM024_ES/HBM024_ES-47-50.pdfl

[9] Open Roberta Lab, "Open roberta lab." [En línea]. Disponible en: https://lab.open-roberta.org/

[10] Blockly, "Blockly." [En línea]. Disponible en: https://developers.google.com/blockly

[11] Blockly, "Define blocks." [En línea]. Disponible en: https://developers.google.com/blockly/guides/create-custom-blocks/define-blocks

[12] Cyberbotics, "Webots." [En línea]. Disponible en: https://cyberbotics.com/

[13] J. Cheng, V. Hu, y J. Lee, "Webots-blockly," 2021. [En línea]. Disponible en: https://github.com/victorhu3/Webots-Blockly

[14] Storming robots development group, "Storming robots development group." [En línea]. Disponible en: https://www.srbots.net/prod/index.html

[15] Python, "Python." [En línea]. Disponible en: https://www.python.org/downloads/windows/