Como citar este artículo: J. Á. Ariza and S. G. Gil, "RaspyLab: A Low-Cost Remote Laboratory to Learn Programming and Physical Computing Through Python and Raspberry Pi," in *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 17, no. 2, pp. 140-149, May 2022, doi: 10.1109/RITA.2022.3166877.


# *RaspyLab*: Un laboratorio remoto de bajo costo para aprender programación y computación física mediante Python y Raspberry Pi

Jonathan Álvarez Ariza, Sergio González Gil

***Abstract*— This article describes the development and assessment of *RaspyLab* which is a low-cost Remote Laboratory (RL) to learn and teach programming with Raspberry Pi and Python language. The RL is composed of 16 stations or nodes that contain hardware components such as display LCD, robotic arm, temperature sensor, among others, and two modes of programming (graphical and text-based) for the students to experiment with their designed algorithms. The concept of the RL was conceived as a pedagogical tool to support the students of Engineering and Computer Science (CS) in an online learning format, given the context of the COVID-19 pandemic. The laboratory has been used by (*n=30*) CS students during the second semester of 2020 in the subject of mathematical logic through the methodology of Problem-Based Learning (PBL). To evaluate preliminary the laboratory, it was used a survey with 3 open-ended questions and 12 closed-ended questions on a Likert scale according to the Technology Acceptance Model (TAM). The outcomes show a good reception of the laboratory, an enhancement of the students' learning regarding the concepts addressed in the course, and an interest of the students for the laboratory to be included in other subjects of the curricula.**

***Index Terms*— Remote laboratory, Problem-Based Learning (PBL), physical computing, programming, Raspberry Pi, Python.**

## I. Introduction

LOS Laboratorios Remotos (LRs) han servido para apoyar las actividades de enseñanza y aprendizaje durante las últimas dos décadas. Aspectos como el incremento de los estudiantes en las aulas de clase, la reducción de los costos respecto a la infraestructura en los laboratorios y la necesidad de modernización y flexibilidad del currículo han guiado a la popularización de estas herramientas en la educación en ingeniería y Ciencias de la Computación (CC) [1]. Actualmente, la pandemia por COVID-19 y los periodos de cuarentena han generado una reducción o eliminación de las clases presenciales que han sido reemplazadas por metodologías de tipo blended-learning o virtuales en un corto periodo de tiempo [2], [3]. Este hecho ha fortalecido el uso de los LRs en la educación superior como una solución temporal para proporcionar experimentación y educación de calidad a los estudiantes. No obstante, no todas las universidades o escuelas han contado con importantes recursos económicos para hacer esta rápida transición, lo cual ha generado una carencia de competencias, habilidades y motivación para aprender en los estudiantes. Si este problema persiste, los estudiantes pueden tener una alta tasa de perdida y deserción de sus programas académicos agravando los efectos educativos producidos por la pandemia. Además, desde una perspectiva pedagógica, existen un conjunto de preguntas acerca de la pertinencia e impacto de las clases en modalidad en línea (online) en el proceso educativo de los estudiantes.

Dados estos elementos, este articulo describe el desarrollo e implementación de *RaspyLab* que se encuentra disponible en [4], el cual es un laboratorio remoto de bajo costo para aprender y enseñar programación con Raspberry Pi y programación en lenguaje Python. El concepto de este LR ha sido pensado no solamente como una herramienta técnica sino como un instrumento pedagógico que sirva a tutores o profesores para crear metodologías que permitan enriquecer el aprendizaje de los estudiantes. El LR cuenta con dos modos de programación y 16 estaciones de trabajo o nodos. En el primer modo de programación, los estudiantes pueden utilizar bloques gráficos (Visualización de Algoritmos VAs) que han sido diseñados para comunicarse con las entradas y salidas de propósito general (GPIOs), periféricos y dispositivos de hardware en cada nodo. En el segundo modo de programación, los estudiantes pueden programar directamente en lenguaje Python. Estos modos de programación dan apoyo a estudiantes quienes son principiantes o intermedios en programación.

El LR ha sido evaluado con 30 estudiantes de primer año de Ciencias de la Computación (CC) en la materia de lógica matemática durante el segundo semestre de 2020 y sus opiniones han sido recolectadas con una encuesta de acuerdo con el modelo TAM [5], [6]. Los estudiantes diseñaron varios algoritmos colaborativamente en grupos de dos o tres integrantes con los modos de programación descritos mediante la metodología de Aprendizaje Basado en Problemas (ABP). Los resultados de la investigación muestran principalmente los

Manuscrito recibido el día de mes de año; revisado día de mes de año; aceptado día de mes de año. English version received Month, day-th, year. Revised Month, day-th, year. Accepted Month, day-th, year.

Jonathan Álvarez Ariza, Corporación Universitaria Minuto de Dios-UNIMINUTO, Facultad de Ingeniería, Programa de Tecnología en Electrónica, 111021 Bogotá, Colombia. E-mail para correspondencia: digpot@gmail.com

Sergio González Gil, Corporación Universitaria Minuto de Dios-UNIMINUTO, Facultad de Ingeniería, Programa de Ingeniería de Sistemas, 111021 Bogotá, Colombia. E-mail: segonzalez@uniminuto.edu

There exists a Spanish version of this article available at XXXX DOI (Digital Object Identifier) Pendiente

siguientes puntos: (1) Una buena recepción y aceptación del LR por los estudiantes. (2) Un incremento de la motivación e interés para aprender los conceptos abordados en el curso. Y (3) una intensión por parte de los estudiantes de usar el LR en el futuro en espacios libres o en otras materias del currículo. Nuestro interés con el LR es doble. Por un lado, proporcionar una herramienta de calidad a nivel tecnológico relacionada con la programación y la computación física [7]. Por otro lado, en el estudio proponemos una arquitectura de bajo costo para LRs que puede ser adaptada por los investigadores o educadores en función de sus intereses en ingeniería y CC. La arquitectura solo necesita un servicio de Servidor Privado Virtual (SPV) en Linux con algunos paquetes de Python, Raspberry Pis y dispositivos de hardware como sensores, pantallas de cristal líquido (LCDs) o robots de bajo costo hechos en impresoras 3D.

Mientras varios laboratorios remotos están orientados hacia la robótica, los sistemas de control, el Internet de las Cosas (IoT) o redes de comunicación, existe una carencia de laboratorios para enseñar y aprender programación y computación física desde cero, reuniendo los modos gráficos y basados en texto. Similarmente, el costo de la infraestructura para los laboratorios remotos puede ser una restricción para su despliegue en entornos de educación superior. Con la arquitectura de bajo costo propuesta en este estudio, buscamos abordar esta problemática. Estos dos aspectos enmarcan las contribuciones principales de este trabajo.

El resto del articulo está organizado de la siguiente manera. La sección 2 expone los antecedentes de este estudio. La Sección 3 describe el diseño y desarrollo del LR desde las perspectivas de software y hardware. La sección 4 ilustra la metodología educativa, participantes e instrumentos adoptados para este estudio. La sección 5 muestra los resultados del estudio y discute las implicaciones educativas del LR. Finalmente, la sección 6 esboza las conclusiones de este estudio.

## II. Antecedentes

Esta sección aborda los trabajos relacionados y el concepto clave en las bases educativas del LR el cual es la computación física.

### *A. Trabajos relacionados*

Concerniente a los desarrollos en Laboratorios Remotos (LRs) para la enseñanza y aprendizaje de programación, de Lima et al.[8] describe un LR que integra codificación basada en bloques y en texto para enseñar introducción a la programación. La propuesta fue evaluada con 18 estudiantes de secundaria y 5 profesores. El LR tuvo una buena aceptación y los estudiantes estuvieron satisfechos con la Interfaz Gráfica de Usuario (IGU) y la interacción con los dispositivos de hardware. En [9], los autores presentan un LR con Python para aprender programación en los cursos CS1 y CS2. 77 estudiantes estuvieron inscritos en la metodología del estudio y los autores indican que la interacción con los experimentos del LR proporcionó una experiencia de aprendizaje de alta calidad que motivó a los estudiantes. Similarmente, en [10] es presentado un LR llamado *Block.ino* orientado a Arduino con fines de aprendizaje de la programación. El LR fue empleado por 144 estudiantes en los niveles de educación secundaria y universitaria. Los autores resaltan la factibilidad y escalabilidad de la plataforma debido al uso de software libre y abierto (FOSS).

Además, los investigadores han utilizado a la robótica en el diseño de LRs para auspiciar el aprendizaje y la comprensión de los estudiantes. Por ejemplo, la robótica puede reducir la brecha de genero respecto a las habilidades de programación en cursos introductorios, como también, ha demostrado ser preferible por la mujeres en estos cursos [11], [12]. En esta línea, los estudios [13], [14] muestran propuestas sobre LRs que incorporan a la robótica. En el primer estudio, los investigadores describen una aplicación web orientada al aprendizaje de la programación de manera autodidacta usando VAs. El estudio se llevó a cabo con 46 estudiantes entre los 11 a 14 años. Una evaluación preliminar con 20 preguntas fue administrada a los estudiantes para caracterizar sus conocimientos en programación. Cuatro módulos fueron propuestos con el empleo del LR y los autores indican que los estudiantes consideraron el LR interesante para aprender programación de una forma autodidacta. Además, los autores reportaron una mejora de las habilidades de los estudiantes en cada módulo. El segundo estudio propone un robot educativo con una plataforma móvil para las áreas STEM. El robot está compuesto de un microcontrolador, motores, batería y sensores. Los autores discuten los componentes técnicos de la plataforma y las ventajas para el aprendizaje en los aspectos de mejoramiento de las habilidades de programación y la resolución de problemas.

En cuanto a propuestas que incluyan la Visualización de Algoritmos (VAs) para introducción a la programación, los estudios [15], [16] ilustran cómo esta clase de herramientas pueden influir en el aprendizaje y la motivación de los estudiantes. El primer estudio explora cómo los VAs pueden mejorar el aprendizaje y la comprensión de las mujeres estudiantes de secundaria. El estudio se llevó a cabo con 24 estudiantes alemanas en dos talleres, utilizando dos herramientas de VAs (*mBlock* y una aplicación de programación web (WPA)). Para analizar los datos, se realizó un estudio con estadística inferencial para encontrar diferencias significativas en las habilidades de programación y la motivación con ambas herramientas de VAs. Los resultados indican que los estudiantes disfrutaron programar con la herramienta (*mBlock*) que proporcionó un alto nivel de facilidad de uso de la programación. Los autores afirman que un diseño basado en la herramienta *Scratch* crea un entorno productivo para estudiantes mujeres sin un conocimiento previo de la programación. El segundo estudio describe el empleo de la herramienta *Scratch* para enseñar programación en paralelo con el método clásico en el curso COMPE112 (Programación en lenguaje C). En el estudio en primer lugar, se compararon los resultados de los años 2010 y 2011 en términos de los estudiantes adscritos al curso vs. quienes lo perdieron con la metodología. En segundo lugar, 55 estudiantes respondieron un

cuestionario en escala Likert donde el análisis mostró que *Scratch* hace que la programación sea más visual y disfrutable ayudando a la comprensión de los conceptos en esta área.

Estos estudios evidencian que el enfoque asumido en este trabajo es coherente desde la perspectiva educativa en la selección de los elementos tales como los modos de programación y el uso de dispositivos de hardware que han sido incluidos en el diseño y despliegue del LR. Las siguientes secciones discuten en detalle estos componentes.

### *B. Computación Física*

Tradicionalmente, los cursos de programación han sido abordados con una importante carga de componentes teóricos en vez de actividades prácticas que ayuden a mejorar el aprendizaje y la comprensión de los estudiantes. Esta tendencia ha sido modificada por la incorporación gradual de la computación física como un concepto central en el currículo de ingeniería y CC. La Computación Física (CF) no es un concepto nuevo en CC, ha estado en el panorama educativo por al menos 20 años. Hodges et al. [7] definen a la CF como una combinación de hardware y software para construir sistemas que sensen e interactúen con el mundo real. Greenwold [17] afirma que la CF es un tipo de interacción humana con máquinas en las que estas manipulan objetos y espacios reales. O' Sullivan e Igoe [18], quizás los primeros en acuñar este término, indican que la CF es una interacción, donde el término interacción es comprendido como "un proceso iterativo de escuchar, pensar, y hablar entre dos o más actores". La CF es esencialmente una actividad con computadores para manipular objetos en el mundo real. Esta interacción es creada principalmente por transductores los cuales son elementos que transforman la variable sensada en un equivalente eléctrico de voltaje o corriente. Más aun, desde un punto de vista constructivista, la CF beneficia las habilidades cognitivas, perceptuales y sociales de los estudiantes [7].

Los principios educativos de la CF están basados principalmente en el constructivismo. Los educadores han sentido la necesidad de hacer que el currículo de ingeniería y CC sea más inclusivo e interesante para los estudiantes. Aspectos como la alta tasa de pérdida o el abandono de los cursos por parte de los estudiantes de programación han requerido metodologías y currículos que se integren más activamente con los estudiantes. Así, el constructivismo parte del principio de que el conocimiento y el aprendizaje son creados activamente por los estudiantes en un proceso de interacción [19], por ejemplo, con la tecnología, los compañeros y los profesores. Entre las ventajas de incluir la CF en los currículos se resaltan el incremento de la motivación, la autoconfianza, el auspicio de la creatividad, la colaboración, la inclusión, el aprender haciendo y el compromiso [7], [20]. El enfoque de la CF ha permitido la incorporación de la robótica en los cursos de programación para mejorar el aprendizaje de los estudiantes.

Como se describió en la sección de trabajos relacionados, la robótica ha contribuido a la reducción de la brecha de género en las habilidades de programación y es preferida por las mujeres en estos cursos. Además, los estudiantes pueden aprender mejor cuando construyen sus robots y los pueden programarlos [21]. En este proceso, los estudiantes pueden transformar diseños de tipo "*Black-box*" (cerrados) en diseños "*White-box*" que pueden ser modificados, reparados y mejorados de acuerdo con los requerimientos de aprendizaje. También, la CF es adecuada para ser integrada con metodologías educativas como el ABP donde el trabajo es orientado a la resolución de problemas en pequeños grupos con la guía del tutor o profesor como un facilitador del proceso de aprendizaje [22].

## III. Diseño y desarrollo del laboratorio remoto *RaspyLab*

Dadas las condiciones para el diseño del LR, las siguientes generalidades fueron consideradas a nivel técnico y educativo.

1. Seleccionar una arquitectura de software rápida que proporcione acceso a los diferentes servicios para los actores en el laboratorio: estudiantes, profesores y personal administrativo. La arquitectura debe estar instalada en un servidor Linux.
2. Inicialmente, crear 16 nodos o estaciones de trabajo. Cada estación debe contar con una Raspberry Pi 4 (2GB), una cámara (8MP o 5MP) y los diferentes dispositivos de hardware para interactuar con los algoritmos. Con la capacidad instalada, entre 30 y 40 estudiantes pueden realizar las diferentes actividades propuestas por el profesor o instructor en un curso.
3. Proporcionar una interfaz de usuario para los estudiantes. La interfaz debe ser accedida mediante protocolo HTTPS en los buscadores que soporten los códecs de video para la cámara disponible en cada nodo.
4. Para la interfaz de los estudiantes fueron creados dos modos de programación. El primero de ellos con bloques gráficos para programadores principiantes y el segundo modo basado en texto para programadores intermedios.
5. La arquitectura de software debe emplear software libre y abierto (FOSS).

Con los anteriores requerimientos fue creada la arquitectura de software y la configuración de hardware para el laboratorio. Cada uno de estos componentes será explicado en las siguientes subsecciones.

### *A. Arquitectura de software*

Para la arquitectura de software fue escogido un Servidor Público Virtual (SPV) en Linux Ubuntu debido a las restricciones de presupuesto y tiempo en el proyecto. Con el SPV garantizamos consistencia y robustez en el servicio del laboratorio y una plataforma centralizada para el *webmaster* y los programadores sin la necesidad de un servidor físico. También en el SPV se usaron varios paquetes de Python para las funciones de acceso de usuario, procesamiento del código

de los estudiantes, enlace con las bases de datos relacionales y la comunicación con la Raspberry Pi en cada nodo. Se seleccionaron estos paquetes para acelerar el proceso de construcción del laboratorio, proporcionando flexibilidad y escalabilidad para incluir experimentos adicionales en futuros estudios. La Fig.1 muestra la arquitectura general del LR.

*1) Interfaz del estudiante*

Para la interfaz del estudiante, fue desplegado un servidor Apache en el SPV. El servidor Apache aloja las páginas web para el acceso de los estudiantes y el tipo de programación

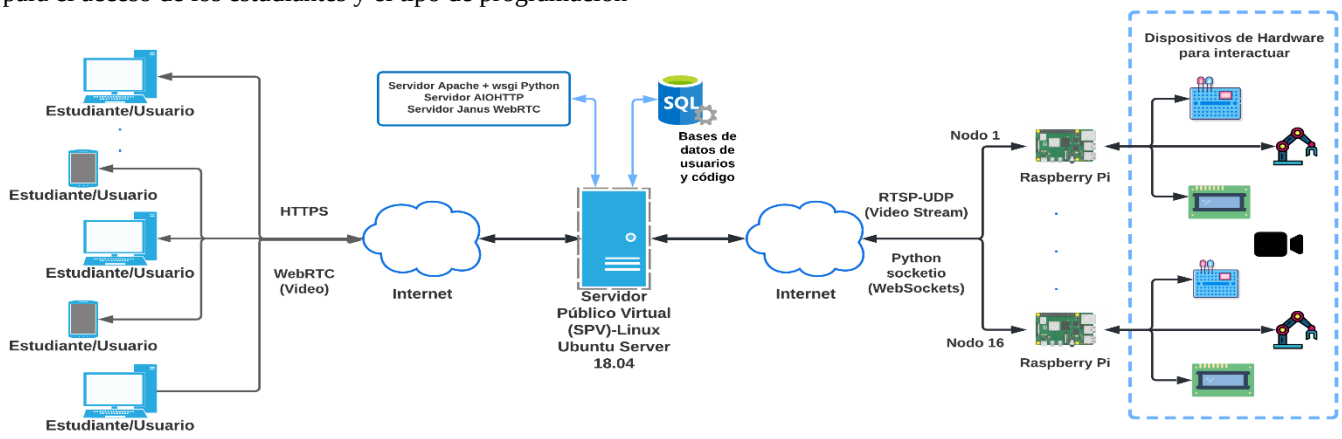



**Fig. 1.** Arquitectura de red propuesta para el LR *RaspyLab*.

seleccionado en los modos de bloques gráficos o texto. La Fig.2 ilustra la interfaz de usuario para la opción de programación basada en bloques.

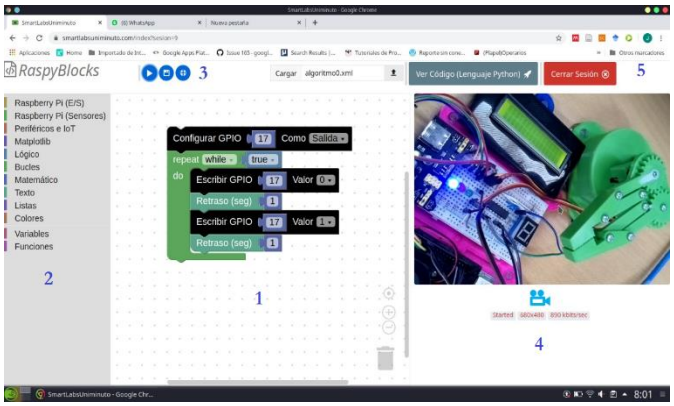


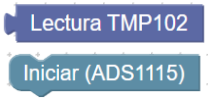
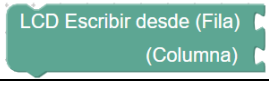
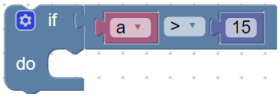
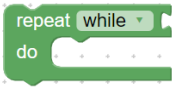

**Fig. 2.** Interfaz de usuario para la opción de programación en bloques. 1. Área de trabajo. 2. Paleta de bloques. 3. Barra de herramientas para las opciones de ejecutar, guardar y cargar. 4. Video. 5. Botones para mostrar el código en Python y cerrar sesión.

Respecto al modo de programación en bloques, se modificó el núcleo de la herramienta de Google *Blockly* [23], la cual es reconocida en educación como una herramienta web para la Visualización de Algoritmos (VAs). *Blockly* transforma cada bloque hacia un equivalente, en este caso, en lenguaje de programación Python de acuerdo con los requerimientos de la Raspberry Pi. Los estudiantes tienen las opciones de ejecutar, guardar, cargar y ver el código respectivo en el lenguaje Python para sus VAs. Asimismo, se desarrollaron varias categorías de bloques para manejar los diferentes dispositivos de hardware, pines de propósito general (GPIOs) y periféricos en la Raspberry Pi. Entre las categorías más relevantes se encuentran las siguientes: *Raspberry Pi GPIOs*, *retrasos*, *display LCD*, *periféricos*, *servomotor*, *Internet de las Cosas (IoT)*, *lógico y bucles*. La Tabla 1 muestra una descripción de estas categorías.

Cómo se mencionó, los bloques gráficos se seleccionaron para estudiantes que son principiantes y desean aprender desde cero. Varios estudios en el campo de las CC [24], [25] han demostrado que la VAs puede enriquecer el proceso de aprendizaje de los estudiantes en la creación de algoritmos y la comprensión de los procesos y conceptos inmersos en ellos. Shaffer et al. [26] argumentan que la VAs puede ser usada por los instructores como parte de una lectura, un laboratorio, o una tarea para enseñar un concepto. No obstante, algunos investigadores y educadores han encontrado que la VAs puede servir como un "puente" para aprender lenguajes basados en texto [27]. Dados estos puntos, decidimos emplear la opción de programación basada en texto para que los estudiantes contrasten sus algoritmos gráficos directamente con este modo. Así pues, se instalaron las diferentes librerías en lenguaje Python para la Raspberry Pi en cada nodo para proporcionar la funcionalidad a los dispositivos de hardware para ambos modos de creación de algoritmos.

TABLA I.
Descripción de las principales categorías para el modo de programación gráfico.

| Categoría | Descripción | Bloque Ejemplo |
|---|---|---|
| *Raspberry Pi (GPIO)* | Bloques para configurar (Entradas, salidas, modulación por ancho de pulso (PMW)), escritura de GPIOs, lectura de GPIOs y retrasos en segundos. | Escribir GPIO 27 Valor 1 |

| | | |
|---|---|---|
| *Sensores* | Bloques para leer sensores y conversor análogo digital como (TMP102, HDC1008, ADS1115, INA219, TSL2591). |   |
| *Display LCD* | Bloque para manejar display (LCD) (2*16 caracteres). |   |
| *Periféricos e IoT* | Bloques para manejar el periférico Asynchronous Receiver-Transmitter (UART) e Internet de las Cosas (IoT) mediante MATLAB *ThingSpeak* [28]. |   |
| *Lógico* | Bloques para sentencias lógicas y condicionales. |   |
| *Bucles* | Bloques para bucles (*for*, *while*). |   |

En la misma línea del modo gráfico, se desarrolló una consola para que los estudiantes escribieran directamente su código en Python. Algunos estudios en CC han mostrado que el lenguaje Python ayuda a reducir la tasa de deserción de los estudiantes y ayuda a mejorar la comprensión e interés en la introducción a la programación [29], [30]. La interfaz es básicamente la misma que el modo gráfico solo con un cambio en el editor en el área de trabajo. En la interfaz fue utilizado el editor de código ACE disponible en (https://ace.c9.io/) para resaltar las diferentes sentencias en el lenguaje de programación para que los estudiantes las reconocieran. La Fig.3 ilustra la interfaz en este modo. Concerniente al acceso al laboratorio, fueron empleados los paquetes de Python *Flask* y *Flask logging* [31] para manejar el acceso del usuario y las acciones cuando un estudiante envía un código para un nodo especifico. Después, el estudiante selecciona un nodo (1 a 16) y el tipo de programación que desea. La Fig.4 ilustra un ejemplo de acceso al laboratorio y la Fig.5 muestra el diagrama general de eventos para el acceso y cierre de sesión para un estudiante.

El estudiante o usuario debe registrarse con un correo y una contraseña acreditados. Los datos personales de los estudiantes son guardados en una base de datos restringida en el SPV. Los datos son proporcionados por la universidad en términos de los nombres de los estudiantes, correos, código de identificación (ID) y esta información es alojada en la base de datos mencionada.

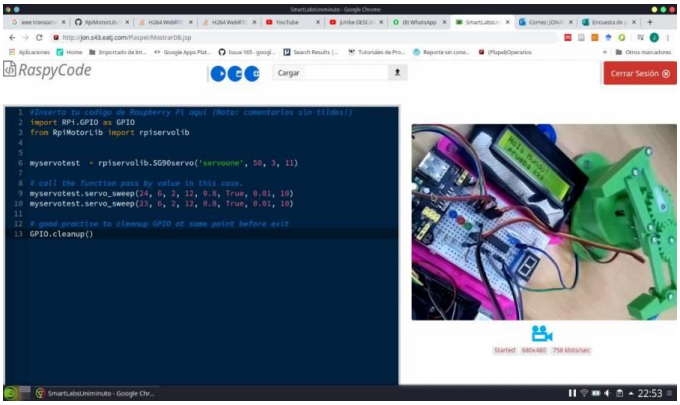



**Fig. 3.** Interfaz de usuario para el modo de programación basada en texto.


**Fig. 4.** Opciones de acceso al LR para un estudiante. 1. Selección de nodo. 2. Opciones de programación (basada en bloques o en texto). 3. Mensaje de bienvenida. 4. Botón de cierre de sesión.

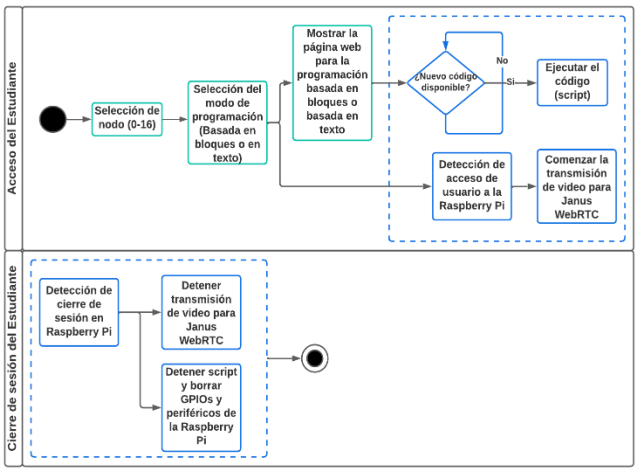



**Fig. 5.** Diagrama para el acceso de un estudiante en los eventos de acceso y cierre de sesión del LR. En azul (funciones ejecutadas por la Raspberry Pi), verde (funciones ejecutadas por el SPV).

Además, el paquete *Flask logging* maneja el acceso de los estudiantes y limita el uso de las sesiones a los usuarios no autorizados.

*2) Transmisión de Video*

Uno de los elementos críticos en la construcción del

laboratorio fue la transmisión de video. Cada nodo cuenta con una cámara Raspberry Pi que muestra el funcionamiento de los elementos de hardware. Para probar la latencia de la transmisión de video para los usuarios o estudiantes, se usaron los servidores y puertas de enlace (gateways) descritos en la Tabla II. El video fue codificado usando la herramienta *FFmpeg* [32] en la Raspberry Pi y enviándolo a través del protocolo de transmisión en tiempo real (RTSP) o usando el protocolo UDP para su posterior procesamiento en el SPV.

TABLE II.
Servidores y puertas de enlace (gateways) probados para el LR.

| **Servidor o Gateway** | **Descripción** | **Latencia Experimentada (segs)** |
|---|---|---|
| Nginx + protocol HLS | Un servidor web, con proxy inverso, balanceador de carga y cache HTTP. | 10 a 20 segs. |
| Node-Media-Server + HTTP FLV | Una implementación de Node.js de servidor de medios RTMP/HTTP-FLV/WS-FLV/HLS/DASH | 5 a 12 segs. |
| Janus WebRTC Gateway | Un Gateway WebRTC para propósito general. | 0.5 a 2 segs. |

Aunque la latencia es establecida por el cliente, esta represento un aspecto crítico para la eficiencia y robustez del laboratorio. El video es del tipo *on-demand*, lo cual significa que en función del usuario y del nodo seleccionado, el video es generado. Con este método se garantiza un consumo de ancho de banda bajo y una reducción del uso de la RAM y el procesador del SPV. Respecto de la Tabla II, la primera prueba fue realizada con el servidor *Nginx* y el protocolo HLS [33] los cuales generaron una latencia que oscilaba entre 10 a 20 segundos. Debido a esta alta latencia para la interacción de los estudiantes, se probaron otras alternativas hasta la selección del Gateway *Janus* WebRTC [34], [35]. *Janus* transforma la transmisión de video codificado de la cámara de la Raspberry Pi en protocolo UDP al estándar WebRTC para los usuarios. Como concepto, la comunicación basada en web en tiempo real (WebRTC) es un estándar relativamente nuevo para las comunicaciones en tiempo real punto a punto que tiene varias ventajas para juegos, transmisión de video y envío de datos desde sensores [36], [37]. El standard WebRTC es totalmente compatible con buscadores como Chrome o Mozilla Firefox. Sin embargo, WebRTC usa el protocolo UDP, lo que puede conducir a cortes y congelamiento de la transmisión de video por varios milisegundos.

*3) Conexión de la Raspberry Pi con el SPV*

Cada Raspberry Pi se conecta con el SPV realizando un requerimiento de tipo *long polling*. Debido a las restricciones en la red donde los nodos están instalados, no se puede acceder directamente a cada Raspberry Pi. En cambio, la Raspberry Pi genera un requerimiento del tipo long polling a través de WebSockets empleando el servidor asíncrono AIOHTTP [38]. Cuando un estudiante envía su código, este es temporalmente almacenado en una pequeña base de datos para ser posteriormente procesado. Este evento es detectado en la Raspberry Pi que extrae el código (script) para ejecutarlo. Todos los eventos son sincronizados con las acciones de acceso y cierre de sesión en el LR de acuerdo con la Fig.5. Durante todo el proceso, un temporizador tipo *watchdog* vigila la ejecución del código en la Raspberry Pi. Si existe un problema de ejecución del código o la Raspberry Pi se bloquea, el temporizador watchdog reinicia tanto la Raspberry Pi como la comunicación con el SPV.

*B. Configuración de Hardware*

Para la configuración de hardware se usaron los elementos que se describen en la Tabla III.

TABLA III.
Componentes de hardware empleados en cada nodo del LR.

| **Componente** | **Descripción** |
|---|---|
| Raspberry Pi 4 (2GB) | Computadora de placa única Raspberry Pi 4 con 2GB de RAM. |
| Cámara | Cámara Raspberry Pi de (8MP o 5MP). |
| LCD 2*16 Alfanumérico | Display de cristal líquido-LCD de (2*16 caracteres). |
| Display de 7 segmentos | Display de 7 segmentos. |
| HDC1008 | Sensor digital de humedad y temperatura de alta precisión. |
| Protoboard | Protoboard con 800 puntos. |
| Brazo robótico | Brazo robótico con dos grados de libertad diseñado en impresión 3D. |
| Leds | Leds para utilización con GPIOs. |
| Adaptador DC | Adaptador 5V, 3A para la Raspberry Pi. |
| Tarjeta SD | Tarjeta microSD de 32GB. |
| Soporte de Cámara | Soporte flexible realizado en impresión 3D para cámara de Raspberry Pi. |

Estos componentes permiten a los estudiantes realizar computación física, experimentando en tiempo real con sus algoritmos. En el hardware se empleó la computadora de placa única Raspberry Pi 4. Existen algunas razones para esta selección. En primer lugar, esta versión proporciona varias opciones de memoria de acceso aleatorio (RAM) de 2GB o 4GB que son superiores a su predecesor, la Raspberry Pi modelo 3. Estas opciones de memoria son más adecuadas para proporcionar alta eficiencia de acuerdo con los requerimientos de video y procesamiento que requiere el laboratorio remoto. En segundo lugar, con estas opciones de memoria y características, por ejemplo, en procesador (cuatro núcleos Cortex A-72 @ 1,5 GHz), GPIOs y protocolos, se puede garantizar la escalabilidad para futuros desarrollos, por ejemplo, en procesamiento de imágenes, robótica, sistemas de control, entre otros.

Se aceleró y se redujeron los costos en el proceso de construcción del hardware diseñando varios componentes en impresión 3D, como cajas, brazo robótico y soporte para la cámara de la Raspberry Pi como se ilustra en la Fig.6. Con esta configuración de hardware, el instructor o profesor puede crear diferentes metodologías que pueden ser adaptadas para las

necesidades de aprendizaje de los estudiantes. Así, los estudiantes pueden comprender conceptos, por ejemplo, en programación y robótica básica. El costo total de los componentes de hardware por nodo en la Tabla III es de USD 111.

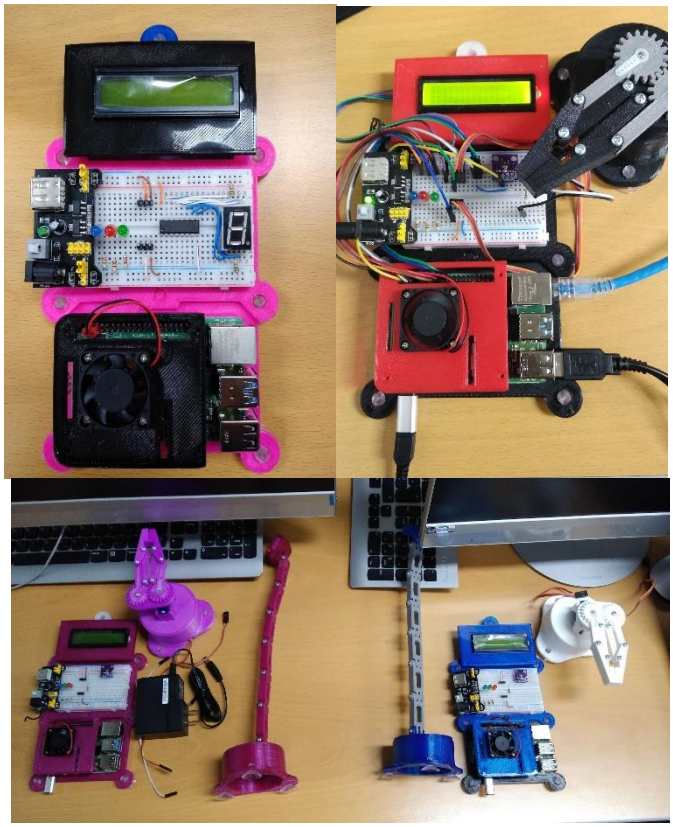


**Fig. 6.** Ejemplos de configuración de hardware para el LR.

## IV. METODOLOGÍA EDUCATIVA

Las bases educativas del LR se basan en la teoría de aprendizaje constructivista [19], [39] y la computación física. En el constructivismo, el aprendizaje y el conocimiento son construidos activamente por los estudiantes a través de las interacciones con compañeros, pero también con la tecnología. En esta interacción con actividades prácticas, los estudiantes construyen sus algoritmos para manipular artefactos (hardware) que pueden ayudar a mejorar la comprensión de los conceptos con la finalidad de crear abstracciones y generalizaciones [21]. No obstante, el proceso de mediación entre la tecnología y el aprendizaje requiere tiempo para ser reconocido como una parte importante del proceso educativo de los estudiantes, en este caso, en programación y pensamiento algorítmico. Por esta razón, los resultados presentados en este estudio son preliminares. Tomando en cuenta estos elementos, la Fig.7 muestra la metodología con el uso del LR que ha sido elaborada con la taxonomía de Bloom en el dominio de aprendizaje cognitivo [40]. La metodología tiene cinco pasos que son explicados de la siguiente manera:

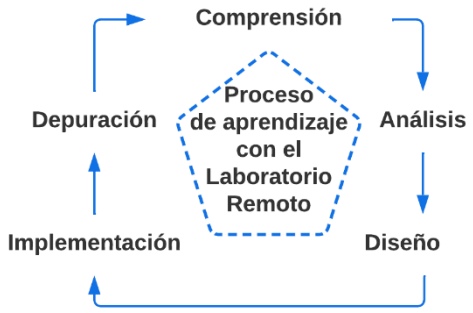



**Fig. 7.** Esquema de aprendizaje propuesto para el LR.

1. *Comprensión:* los estudiantes comprenden los conceptos abordados en el curso. Inicialmente los estudiantes identifican los usos de estos conceptos en la solución de problemas dentro de situaciones reales o casos de estudio. El instructor o profesor pueden usar metodologías sincrónicas o asincrónicas para el aprendizaje como videos, lecturas en línea, tutoriales, etc.
2. *Análisis:* el profesor o instructor propone un problema que reúna el conocimiento y las habilidades adquiridas por los estudiantes. Posteriormente, los estudiantes analizan cómo desarrollar una solución utilizando los aspectos mencionados colaborativamente.
3. *Diseño:* los estudiantes inician el diseño de sus algoritmos para la situación problema planteada. El algoritmo puede ser representado en términos de un pseudo código, un diagrama de flujo, etc. También los estudiantes comprenden las restricciones y limitaciones del diseño.
4. *Implementación:* con el diseño, los estudiantes implementan sus algoritmos en el LR seleccionando el modo de programación más adecuado de acuerdo con las necesidades de aprendizaje y los requerimientos del currículo.
5. *Depuración:* los estudiantes evalúan los problemas, errores, o posibilidades para mejorar sus algoritmos en el LR. Los estudiantes corrigen los errores e implementan de nuevo los diseños de sus algoritmos en el LR.

Estos pasos fueron asumidos en las actividades del curso de Lógica Matemática ofrecido a estudiantes de primer semestre de CC. El enfoque educativo seleccionado en el curso fue el Aprendizaje Basado en Problemas (ABP) en donde varios problemas fueron resueltos colaborativamente en grupos de dos o tres personas, como se mencionó con ayuda del LR. La materia abarca temas en lógica proposicional, conjuntos, operadores lógicos, y lenguajes formales que fueron fomentados para aprender con el LR. El laboratorio fue utilizado por cada estudiante durante 6 horas por semana durante 5 semanas. Los estudiantes usaron ambos modos de

programación, contrastando sus algoritmos gráficos con la sintaxis del lenguaje Python.

### *A. Participantes*

30 estudiantes de primer semestre de CC participaron de la metodología y de la fase de pruebas del LR. El promedio de edad fue de 18.5 años con una distribución de genero de 90% hombres y 10% mujeres.

### *B. Instrumentos*

Para recolectar la información sobre las percepciones de los estudiantes fue realizada una encuesta en escala Likert con 12 preguntas cerradas y dos preguntas abiertas. El rango de las preguntas cerradas fue establecido desde 1: totalmente en desacuerdo hasta 4: totalmente de acuerdo. La encuesta se basó en el modelo de aceptación de la tecnología (TAM) propuesto por Davis et al. [6], [41] para evaluar el uso del LR. El enfoque del TAM es un buen predictor de las intenciones de uso del LR y ha sido usado ampliamente en diversos estudios en educación [5], [42]–[44]. El TAM es determinado por tres factores y los estímulos-respuesta externos los cuales pueden ser fomentados por aspectos como la motivación, el interés, el compromiso, la ansiedad, entre otros, experimentados por los estudiantes. La relación entre estos constructos es ilustrada en la Fig.8.

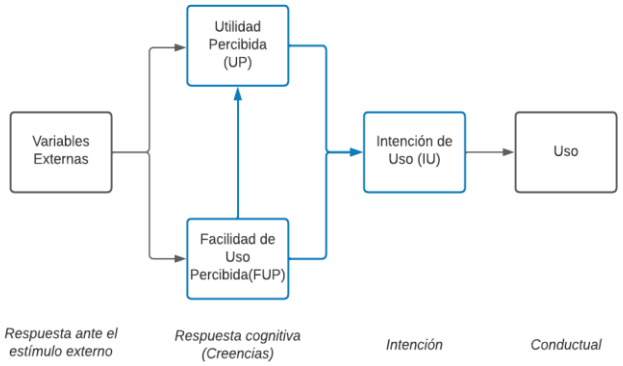



**Fig. 8.** Modelo de aceptación de la tecnología TAM adaptado de [41].

1. *Utilidad Percibida (UP)*: define el grado en el que los estudiantes encontraron que el LR contribuyó a potenciar su aprendizaje en los conceptos y temas abordados en el curso.
2. *Facilidad de Uso Percibida (FUP)*: indica el grado en el cual los estudiantes usaron fácilmente las características y la interfaz del LR sin un excesivo esfuerzo o confusión.
3. *Intención de Uso (IU)*: Define el grado en el cual los estudiantes están conscientes de usar el LR en el futuro en otros espacios educativos para aprender programación y computación física.

Este modelo es adecuado para evaluar preliminarmente el LR porque proporciona información sobre las percepciones que los estudiantes obtuvieron al emplear el LR, como, asimismo indica el grado de aceptación del LR. Sin embargo, una evaluación por un mayor tiempo es requerida para analizar las implicaciones educativas del LR en los estudiantes.

De este modo, 20 estudiantes, (66.66%) de los participantes respondieron la encuesta final. Los datos fueron analizados usando IBM SPSS Statistics V.23. La Tabla IV muestra la encuesta con su respectivo valor del Alfa de Cronbach (α) de acuerdo con los factores en el TAM. En todos los factores, el valor del Alfa de Cronbach (α) fue superior a 0.7 lo cual demuestra la consistencia interna del instrumento [45].

TABLE IV.
Encuesta para las percepciones de los estudiantes sobre el LR de acuerdo con el modelo TAM. (*n=20*).

| Factor TAM | Pregunta (P) | Alfa de Cronbach (α) |
|---|---|---|
| **UP** | **P1**. ¿El LR me ayudó a comprender cómo se podrían aplicar los temas del curso en situaciones reales? | **0.782** |
| | **P2**. ¿Los modos de programación en el LR me ayudaron a comprender y aplicar los conceptos del curso? | |
| | **P3**. Con el LR, ¿pude experimentar y aprender a mi propio ritmo? | |
| | **P4**. ¿Creo que la LR es una herramienta que beneficia mi aprendizaje? | |
| | **P5**. Mediante el LR, ¿pude contrastar los bloques gráficos con el respectivo código en lenguaje Python para aprender su sintaxis? | |
| **FUP** | **P6.** ¿Considero que el acceso al LR fue fácil? | **0.759** |
| | **P7**. ¿Considero que fue fácil aprender a manejar la interfaz del LR? | |
| | **P8**. ¿Pude ejecutar mis algoritmos y observar su funcionamiento con el video en tiempo real? | |
| | **P9**. ¿Considero que la interfaz del LR es adecuada para crear mis algoritmos? | |
| | **P10**. ¿Considero que es fácil aprender las características del LR? | |
| **IU** | **P11**. ¿Me gustaría que el LR se incluyera en otras asignaturas del plan de estudios en el futuro? | **0.772** |
| | **P12**. ¿Usaría el laboratorio remoto en algunos de mis espacios libres para seguir experimentando y aprendiendo? | |

## V. Resultados y Discusión

Esta sección describe y discute los resultados experimentales de la metodología. Además, se discute las ventajas y limitaciones del LR. La Tabla V describe las estadísticas descriptivas para las respuestas de los estudiantes en la encuesta.

TABLA V.
Media y Desviación Estándar (DE) para las preguntas de la encuesta. (*n=20*).

| Factor TAM | Pregunta | Media | DE |
|---|---|---|---|
| | **P1** | 3.6 | 0.503 |
| | **P2** | 3.5 | 0.513 |

| | | | |
|---|---|---|---|
| **UP** | **P3** | 3.45 | 0.510 |
| | **P4** | 3.75 | 0.444 |
| | **P5** | 3.35 | 0.489 |
| **FUP** | **P6** | 3.05 | 0.759 |
| | **P7** | 3.4 | 0.598 |
| | **P8** | 3.25 | 0.786 |
| | **P9** | 3.6 | 0.502 |
| | **P10** | 3.5 | 0.606 |
| **IU** | **P11** | 3.75 | 0.444 |
| | **P12** | 3.7 | 0.470 |

Con esta información, se calculó la media, mediana y desviación estándar de cada factor en el TAM. La Tabla VI muestra estos valores.

TABLA VI.
Media, mediana y Desviación Estándar (DE) para los factores del TAM. (*n=20*).

| Factor TAM | Media | Mediana | DE |
|---|---|---|---|
| **UP** | 3.53 | 3.4 | 0.363 |
| **FUP** | 3.36 | 3.4 | 0.47 |
| **IU** | 3.725 | 4 | 0.41 |

Basado en estos valores, los estudiantes tienen una buena Utilidad Percibida (UP) del LR. Los estudiantes indicaron que el LR les ayudó a aprender y aplicar los conceptos en el curso en tiempo real. Asimismo, ellos argumentaron la importancia de esta clase de laboratorio a pesar de la distancia y los aislamientos asociados con la pandemia por COVID-19. El modo de programación que más les agradó a los estudiantes fue el basado en bloques. Respecto a lo anterior, algunos estudiantes comentaron lo siguiente:

***E1***. *La experiencia de usar una herramienta como esta tuvo un impacto muy positivo para aprender los temas del curso.*

***E2.*** *Fue interesante la forma en que pude programar. Los resultados se mostraron directamente en el video en tiempo real.*

***E3.*** *Para mí fue interesante el hecho de que las funciones se ejecutaran en tiempo real en el laboratorio remoto.*

*E4. Con el video en tiempo real, me di cuenta de varios errores en mis códigos y los arreglé.*

*E5. Fue muy interesante la forma de modificar y contrastar los códigos en bloques y en el lenguaje de programación.*

*E6. Fue interesante cómo pude experimentar, colocando y modificando las diferentes instrucciones y códigos en la interfaz y viendo su respuesta en tiempo real.*

El factor FUP obtuvo el menor puntaje en la encuesta. Algunos estudiantes manifestaron dificultades en el acceso al LR y en el video en tiempo real debido a la conexión de internet y problemas de los códecs de video. Como se mencionó, el estándar WebRTC tiene algunos problemas de congelamiento y caída de video en algunos casos. Sin embargo, la reconexión del cliente es garantizada por el Gateway *Janus* y varios scripts que fueron diseñados en lenguaje Python. Además, los problemas de acceso al LR fueron reparados y los problemas de robustez experimentados en la transmisión de video serán corregidos para trabajar con estudiantes en cursos futuros. Estos problemas influenciaron la percepción de los estudiantes en este factor. A pesar de estos problemas, los estudiantes evaluaron el LR en una escala de 1 a 5 como se muestra en la Fig.9. La media de estos valores fue 4.25.

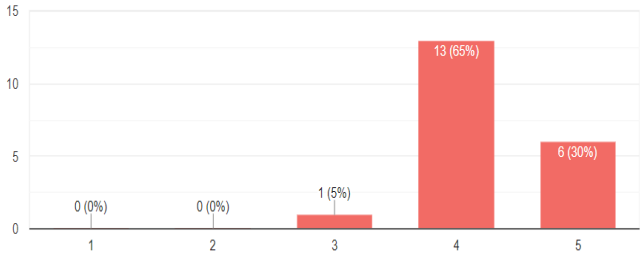



**Fig. 9.** Evaluación general del LR. (*n=20*).

El factor IU fue evaluado con el puntaje más alto, a pesar de los problemas técnicos descritos en el LR. Los estudiantes afirmaron que desean que el laboratorio sea incluido en otras materias del plan de estudio. Incluso los estudiantes están dispuestos a emplear parte de su tiempo libre en el aprendizaje de otros conceptos de las CC con el LR. Este hecho demuestra que el laboratorio tuvo una buena recepción y aceptación de los estudiantes y contribuyó a fortalecer su aprendizaje. Con estos factores, fue calculada la matriz de correlaciones de Pearson para conocer la relación entre los factores del TAM. La Tabla VII muestra estos valores. Para este caso existen relaciones estadísticamente significativas, por un lado, entre el factor UP y FUP ($r(20)=0.710$ $p<0.01$) como es demostrado por los resultados analizados previamente. Por otro lado, el factor IU es principalmente determinado por su relación con el factor UP ($r(20)=0.532$ $p=0.016$). Los estudiantes encontraron útil el LR para su aprendizaje y este hecho fue más relevante que el factor FUP en la intención de usar el LR. De una manera similar los estudiantes indicaron los aspectos positivos y por mejorar en el LR. Una síntesis de algunos de estos comentarios se presenta en la Tabla VIII.

TABLA VII.
Resultados de la correlación de Pearson para los factores del TAM (*n=20*).

| Factor TAM | UP | FUP | IU |
|---|---|---|---|
| **UP** | 1 | | |

| | | | |
|---|---|---|---|
| **FUP** | 0.710* | 1 | |
| **IU** | 0.532** | 0.428 | 1 |

*La correlación es significativa al nivel de 0.01. Valor actual $p$=0.00. **La correlación es significativa al nivel de 0.05. Valor actual $p$=0.016.

TABLA VIII.
Síntesis de los aspectos positivos y por mejorar en el LR.

| Descripción | Tema | Ejemplos de comentarios |
|---|---|---|
| **Aspectos Positivos** | **Modo de programación** | "Creo que la programación con bloques es más dinámica" |
| | | "La programación con bloques es más gráfica e intuitiva para los estudiantes" |
| | | "Considero que ambas formas de programar son interesantes porque facilitan el aprendizaje" |
| | **Procesamiento y video en tiempo real** | "Fue interesante el video en tiempo real porque pudimos observar nuestras fallas en la estructura del código" |
| | | "La cámara en tiempo real fue muy interesante" |
| | | "El laboratorio cuenta con diferentes herramientas y su funcionamiento en modalidad de clases virtuales fue interesante" |
| | **Aprendizaje** | "La experiencia de usar una herramienta como esta tuvo un impacto muy positivo para aprender los temas del curso" |
| | | "El hecho de la distancia no es un obstáculo para aprender y ver el funcionamiento de los algoritmos que uno crea en casa" |
| | | "Fue interesante cómo pude experimentar, colocando y modificando diferentes instrucciones y códigos en la interfaz del LR" |
| **Aspectos por mejorar** | **Cámara y video** | "El laboratorio es bueno, pero, en algunas ocasiones, no daba video" |
| | | "Para mí muchas veces el video generaba errores que dificultaban el trabajo con el laboratorio" |
| | | "Algunas veces el video tuvo algunos errores" |
| | | "Tal vez la posición de la cámara" |
| | **Funciones de reinicio** | "Sería bueno si el proceso de reinicio fuera más rápido en caso de falla" |

En suma, estos elementos muestran que el LR tuvo una buena recepción de los estudiantes que lo usaron en función de sus necesidades de aprendizaje en el curso. El enfoque del TAM fue un punto de partida adecuado para conocer las percepciones de los estudiantes respecto al LR y cómo este fue una herramienta tecnológica que apoyó el aprendizaje y la motivación en los estudiantes.

## VI. Conclusiones y trabajo futuro

En este artículo, se describió un laboratorio remoto de bajo costo que fue usado para aprender y enseñar programación y computación física con Raspberry Pi y lenguaje Python. El concepto del LR surgió de una rápida transición de las clases presenciales a un formato de aprendizaje en línea debido a las restricciones de movilidad y cuarentena por COVID-19 que limitaron el acceso de los estudiantes a los laboratorios, lo cual tuvo consecuencias para su proceso de aprendizaje. Con el desarrollo del LR, en primer lugar, se probó que es posible construir laboratorios remotos de bajo costo en un corto periodo de tiempo de acuerdo con las necesidades de aprendizaje de los estudiantes. La estructura de software con el SPV, los paquetes de Python, y el uso de impresoras 3D para la configuración del hardware permitieron acelerar el diseño y construcción del LR. En segundo lugar, los resultados demostraron que los estudiantes aprovecharon las características del LR para su aprendizaje y la compresión de los temas en el curso. Asimismo, los estudiantes manifestaron su intención de usar el LR en futuros cursos o incluso en su tiempo libre. Este hecho muestra que los diferentes componentes del LR en software y hardware motivaron a los estudiantes a aprender programación y experimentar con actividades prácticas. El LR fue orientado no solamente como una herramienta tecnológica sino también como un elemento pedagógico para promover el aprendizaje y la comprensión de los estudiantes. El trabajo futuro se orientará a mejorar el funcionamiento técnico del LR, especialmente en lo relacionado con la transmisión de video y el acceso de los estudiantes. También, se adelantará un estudio de las implicaciones educativas del LR en las habilidades y competencias de los estudiantes en ingeniería y CC. Complementariamente, las características del laboratorio serán mejoradas para cumplir con los requerimientos del estándar *IEEE 1876-2019 - IEEE Standard for Networked Smart Learning Objects for Online Laboratories.*

## REFERENCIAS

[1] I. Gustavsson *et al.*, "On objectives of instructional laboratories, individual assessment, and use of collaborative remote laboratories," *IEEE Trans. Learn. Technol.*, vol. 2, no. 4, pp. 263–274, 2009, doi: 10.1109/TLT.2009.42.
[2] M. D. H. Rahiem, "The emergency remote learning experience of university students in Indonesia amidst the COVID-19 crisis," *Int. J. Learn. Teach. Educ. Res.*, vol. 19, no. 6, pp. 1–26, 2020, doi: 10.26803/ijlter.19.6.1.
[3] C. Hodges, S. Moore, B. Lockee, T. Trust, and A. Bond, "The difference between emergency remote teaching and online learning," *Educ. Rev.*, vol. 27, 2020.
[4] Á. A. Jonathan and G. G. Sergio, "SmartLabsUniminuto," 2021. [Online]. Available: https://www.smartlabsuniminuto.com/. [Accessed: 21-Feb-2022].
[5] I.-F. Liu, M. C. Chen, Y. S. Sun, D. Wible, and C.-H. Kuo, "Extending the TAM model to explore the factors that affect intention to use an online learning community," *Comput. Educ.*, vol. 54, no. 2, pp. 600–610, 2010.
[6] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Q.*, pp. 319–340, 1989.
[7] S. Hodges, S. Sentance, J. Finney, and T. Ball, "Physical computing: A key element of modern computer science education," *Computer (Long. Beach. Calif).*, vol. 53, no. 4, pp. 20–30, 2020.
[8] J. P. C. de Lima, L. M. Carlos, J. P. S. Simão, J. Pereira, P. M. Mafra, and J. B. da Silva, "Design and implementation of a remote lab for teaching programming and robotics," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 86–91,

2016.
[9]H. Guerra, A. Cardoso, V. Sousa, and L. M. Gomes, "Remote Experiments as an Asset for Learning Programming in Python.," *Int. J. Online Eng.*, vol. 12, no. 4, 2016.
[10] J. B. da Silva, G. de Oliveira, I. N. da Silva, P. M. Mafra, and S. M. S. Bilessimo, "Block. ino: Remote Lab for Programming Teaching and Learning," *Int. J. Adv. Eng. Res. Sci.*, vol. 7, 2020.
[11] M. M. McGill, "Learning to program with personal robots: Influences on student motivation," *ACM Trans. Comput. Educ.*, vol. 12, no. 1, pp. 1–32, 2012.
[12] M. A. Rubio, R. Romero-Zaliz, C. Mañoso, and P. Angel, "Closing the gender gap in an introductory programming course," *Comput. Educ.*, vol. 82, pp. 409–420, 2015.
[13] P. G. Feijóo and F. De la Rosa, "RoBlock: programming learning with mobile robotics," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 2016, p. 361.
[14] M. M. Rahaman, E. Mahfuj, M. M. Haque, R. S. Shekdar, and K. Z. Islam, "Educational robot for learning programming through Blockly based mobile application," *J. Technol. Sci. Eng. (JTSE)[US ISSN 2693-1389]*, vol. 1, no. 2, pp. 21–25, 2020.
[15] M. Seraj, E.-S. Katterfeldt, K. Bub, S. Autexier, and R. Drechsler, "Scratch and Google Blockly: How Girls' Programming Skills and Attitudes are Influenced," in *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, 2019, pp. 1–10.
[16] D. Ozoran, N. Cagiltay, and D. Topalli, "Using scratch in introduction to programming course for engineering students," in *2nd International Engineering Education Conference (IEEC2012)*, 2012, vol. 2, pp. 125–132.
[17] S. Greenwold, "Spatial computing," *Massachusetts Inst. Technol. Master*, 2003.
[18] D. O'Sullivan and T. Igoe, *Physical computing: sensing and controlling the physical world with computers*. Course Technology Press, 2004.
[19] M. Ben-Ari, "Constructivism in computer science education," *Acm sigcse Bull.*, vol. 30, no. 1, pp. 257–261, 1998.
[20] M. Przybylla and R. Romeike, "Physical Computing and Its Scope--Towards a Constructionist Computer Science Curriculum with Physical Computing.," *Informatics Educ.*, vol. 13, no. 2, pp. 241–254, 2014.
[21] D. Alimisis and C. Kynigos, "Constructionism and robotics in education," *Teach. Educ. Robot. Constr. Pedagog. methods*, pp. 11–26, 2009.
[22] M. Moallem, W. Hung, and N. Dabbagh, *The Wiley handbook of problem-based learning*. Wiley Online Library, 2019.
[23] N. Fraser, "Ten things we've learned from Blockly," in *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, 2015, pp. 49–50.
[24] D. Mason and K. Dave, "Block-based versus flow-based programming for naive programmers," in *2017 IEEE blocks and beyond workshop (B&B)*, 2017, pp. 25–28.
[25] Y. Matsuzawa, Y. Tanaka, and S. Sakai, "Measuring an impact of block-based language in introductory programming," in *International Conference on Stakeholders and Information Technology in Education*, 2016, pp. 16–25.
[26] C. A. Shaffer *et al.*, "Algorithm visualization: The state of the field," *ACM Trans. Comput. Educ.*, vol. 10, no. 3, p. 9, 2010.
[27] M. Noone and A. Mooney, "Visual and textual programming languages: a systematic review of the literature," *J. Comput. Educ.*, vol. 5, no. 2, pp. 149–174, 2018.
[28] Mathworks, "ThingSpeak - MATLAB & Simulink." .
[29] A. Jayal, S. Lauria, A. Tucker, and S. Swift, "Python for teaching introductory programming: A quantitative evaluation," *Innov. Teach. Learn. Inf. Comput. Sci.*, vol. 10, no. 1, pp. 86–90, 2011.
[30] U. Nikula, J. Sajaniemi, M. Tedre, and S. Wray, "Python and roles of variables in introductory programming: experiences from three educational institutions," *J. Inf. Technol. Educ. Res.*, vol. 6, no. 1, pp. 199–214, 2007.
[31] M. Grinberg, *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.," 2018.
[32] F. Korbel, *FFmpeg Basics: Multimedia handling with a fast audio and video encoder*. Frantisek Korbel, 2012.
[33] D. DeJonghe, *Nginx CookBook*. O'Reilly Media, 2020.
[34] A. Amirante, T. Castaldi, L. Miniero, and S. Pietro Romano, "Janus: a general purpose WebRTC gateway," in *Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications*, 2014, pp. 1–8.
[35] A. Amirante, T. Castaldi, L. Miniero, and S. Pietro Romano, "Performance analysis of the Janus WebRTC gateway," in *Proceedings of the 1st Workshop on All-Web Real-Time Systems*, 2015, pp. 1–7.
[36] R. Manson, "Getting started with WebRTC: Explore WebRTC for real-time peer-topeer communication, ser," *Community Exp. Distill. Birmingham, UK Packt Pub*, 2013.
[37] Google Inc, "WebRTC." [Online]. Available: https://webrtc.org/. [Accessed: 22-Feb-2022].
[38] "Welcome to AIOHTTP — aiohttp 3.8.1 documentation." [Online]. Available: https://docs.aiohttp.org/en/stable/. [Accessed: 23-Feb-2022].
[39] S. Hadjerrouit, "Constructivism as guiding philosophy for software engineering education," *Acm Sigcse Bull.*, vol. 37, no. 4, pp. 45–49, 2005.
[40] M. E. Hoque, "Three domains of learning: cognitive, affective and psychomotor," *J. EFL Educ. Res.*, vol. 2, no. 2, pp. 45–52, 2016.
[41] T. Teo, *Technology acceptance in education*. Springer Science & Business Media, 2011.
[42] Z. Shana and E. S. A. Abulibdeh, "Cloud computing issues for higher education: theory of acceptance model," 2017.
[43] M. Chow, D. K. Herold, T.-M. Choo, and K. Chan, "Extending the technology acceptance model to explore the intention to use Second Life for enhancing healthcare education," *Comput. Educ.*, vol. 59, no. 4, pp. 1136–1144, 2012.
[44] A. Tlili, F. Essalmi, and M. Jemni, "Improving learning computer architecture through an educational mobile game," *Smart Learn. Environ.*, vol. 3, no. 1, pp. 1–14, 2016.
[45] C. H. Yu, "An introduction to computing and interpreting Cronbach Coefficient Alpha in SAS," in *Proceedings of 26th SAS User Group International Conference*, 2001, vol. 2225, pp. 1–6.

**Jonathan Álvarez Ariza** es profesor de tiempo completo del departamento de Tecnología en Electrónica en la Corporación Universitaria Minuto de Dios-UNIMINUTO (Bogotá, Colombia). Es ingeniero electrónico y Master en educación (M.Ed.) de las universidades colombianas Universidad Central y Universidad de La Salle, respectivamente. También, lidera el semillero en control automático (SeCon) y ha participado de varios proyectos de educación en ingeniería. Sus áreas de investigación son el control automático, los sistemas embebidos y la educación en ingeniería.

**Sergio González Gil** es profesor de tiempo completo de la Corporación Universitaria Minuto de Dios-UNIMINUTO (Bogotá, Colombia) desde 2012. Es ingeniero electrónico de la Universidad Antonio Nariño (Colombia) y recibió un máster en administración de negocios (MBA) de la universidad Viña del Mar (Chile). También es especialista en pedagogía y telecomunicaciones. Sus áreas de investigación son la educación en ingeniería, la robótica y las telecomunicaciones.