

Recolección, envío, almacenamiento y visualización de datos considerando programación basada en bloques de placas MicroBit, ESP, sensores y Wifi

Alejandra Beatriz Lliteras^{1,2}[0000-0002-4148-1299], Juan Pablo Lozano Arce¹[0009-0001-7428-9455], Andrés Rodríguez¹[0000-0002-7600-6041]

¹ UNLP. Facultad de Informática, Centro LIFIA, Argentina

² CIPBA

{alejandra.lliteras, jlozano, andres.rodriguez}@lifia.info.unlp.edu.ar

Abstract. El Pensamiento Computacional es una de las habilidades a desarrollar desde la escuela secundaria y así brindar igualdad de oportunidades a los estudiantes para una participación plena como ciudadanos digitales. En este trabajo se presenta una infraestructura para poner en práctica aspectos de recolección de datos con sensores conectados a una placa Microbit la que conectada a una ESP permite enviarlos vía Wifi a un servidor Web. Se usa programación basada en bloques en MakeCode proponiendo una extensión. Se brinda además un servidor que permite recibir paquetes de datos, validarlos, almacenarlos y dejarlos disponibles para su posterior visualización. Se brindan guías para el despliegue del servidor propuesto y para la preconfiguración de la placa ESP para que puedan ser usados en diferentes contextos. Se presenta un caso de estudio usando la extensión propuesta y el servidor y se realizó una primera validación con estudiantes de cuarto año de secundario.

Keywords: Pensamiento Computacional, Ciudadanía Digital, Robótica Educativa, IoT, Recolección de Datos, Visualización de Datos, MicroBit, ESP, Sensor, WiFi, Comunicación Inalámbrica, Nivel Secundario, Programación Basada en Bloques, MakeCode

1 Introducción

El Pensamiento Computacional ha cobrado relevancia en el ámbito educativo en los últimos años. Si bien el término fue acuñado por Papert [1] hace más de tres décadas, en el año 2006 fue ampliamente difundido por Wing [2] y desde entonces su vigencia permanece. La autora sostiene que ese tipo de pensamiento es necesario para todos los individuos, no solo para quienes se dedican a las ciencias de la computación, sino que también para quienes se desempeñan en otras disciplinas y profesiones. Su objetivo, y el de otros autores en el tema, es agregar e integrar el Pensamiento Computacional como un saber fundamental en el currículum de la educación básica (inicial, primaria y secundaria) y de esta forma avanzar hacia una actualización de contenidos que no solo esté basada en habilidades informáticas sino en conocimientos de campos disciplinares curriculares en general.

Si bien existen diversas definiciones del Pensamiento Computacional y se ha trabajado en este tema durante muchos años, aun no existe una definición consensuada [3]. En este trabajo se considera que el Pensamiento Computacional es una forma de resolver y analizar problemas de manera que sean ejecutadas por una computadora y que, como mencionan Dening y Tedre, involucra habilidades y conceptos como la abstracción, la descomposición, la generalización, los algoritmos (y su diseño) así como, el diseño, la recolección y el análisis de datos [3].

Para desarrollar el Pensamiento Computacional existen diferentes esfuerzos, en general, asociados a temas relacionados a la programación y a las ciencias de la computación. Por ejemplo, en el año 2013 se creó Code.org¹ con origen en Estados Unidos y de alcance internacional. Por otro lado, en el mismo año, pero en Argentina, se suma la iniciativa Program.ar² de la mano de la fundación Sadosky³ [4]. Si bien estos programas implican avances significativos en cuanto a la enseñanza de la programación, en la actualidad, uno de los desafíos más grandes radica en la manera de promover este tipo de pensamiento a toda la currícula de la escuela secundaria en forma transversal y trascendiendo la enseñanza de la programación por sí sola.

En el año 2018 se sancionaron en Argentina los Núcleos de Aprendizaje Prioritario (NAP) de Educación Digital, Programación y Robótica⁴ que plantean una integración del Pensamiento Computacional en forma transversal a las diferentes áreas, evitando circunscribirla a la asignatura de Informática, sin embargo, su difusión y adopción aún hoy es escasa. Adicionalmente, en 2021 Argentina lanzó un plan Federal de Robótica Educativa e Internet de las Cosas en el que participan 14 provincias.

En este trabajo se destaca la importancia de desarrollar el Pensamiento Computacional en los estudiantes secundarios considerando el vertiginoso avance tecnológico [5]. Ante esta realidad es importante trabajar en post de que los estudiantes puedan comprender como funcionan las tecnologías, usarlas adecuadamente y pensar en cómo proponer nuevas alternativas, con una actitud responsable y crítica de modo que puedan comprender el impacto que éstas tienen en la sociedad, lo que en palabras de Ribble no es otra cosa más que formar a los estudiantes como ciudadanos digitales [6].

El desarrollo del Pensamiento Computacional puede ser abordado de diferentes formas, por ejemplo, mediante la visualización de datos [7], la programación [8], la robótica educativa [9], IoT (de la sigla en inglés para *Internet of Things*, previamente nombrado en castellano como Internet de las Cosas) [10] y la Inteligencia Artificial [11]. En este trabajo, se hará foco en el desarrollo de este tipo de pensamiento desde la programación de aspectos de robótica educativa y de IoT para la posterior visualización de datos.

Un primer paso hacia la solución de la incorporación del Pensamiento Computacional de manera transversal en la currícula es proveer a los docentes y alumnos de herramientas accesibles y simples de usar que engloben las diferentes aristas de este tipo de pensamiento y para la programación. En particular, para facilitar los primeros pasos en la programación, se suelen usar lenguajes de

¹ <https://code.org/>

² <https://program.ar/>

³ <https://fundacionsadosky.org.ar/>

⁴ <https://www.educ.ar/recursos/150123/nap-de-educacion-digital-programacion-y-robotica>

programación basados en bloques ya que estos bajan la complejidad que presentan los lenguajes de programación basados en textos y favorecen la adquisición de habilidades del Pensamiento Computacional [12].

Cuando se piensa en el desarrollo del Pensamiento Computacional considerando temas de robótica o IoT en el área de educación, en general se hace en términos de kits que ya vienen pre armados y con instrucciones concretas para que los docentes puedan trabajar en el aula usando placas como, por ejemplo, MicroBit⁵ o Arduino⁶. Las placas Arduino si bien están ampliamente difundidas para diversas actividades educativas, pueden presentar desafíos al momento de su uso tanto para docentes como para estudiantes de escuelas no técnicas principalmente, a quienes le puede resultar complejo el cableado de circuitos que se necesitan para trabajar con ellas. Por lo anterior y debido a que ya además traen integrados algunos componentes es que Pech y Novák proponen el uso de placas MicroBit [13]. A raíz de lo anterior es que para este trabajo se considera el uso de placas MicroBit, en particular, en su versión 2 y de aquí en adelante, estas placas serán nombradas simplemente como MicroBit asumiendo la versión. Estas placas tienen la opción de almacenar datos, por ejemplo, provenientes de sensores, o bien de enviarlos usando el Bluetooth que trae incorporado. En cambio, cuando el envío se quiere hacer inalámbricamente vía Wifi es necesario sumar un componente de hardware, como por ejemplo una placa ESP. En particular, en este trabajo se usará una placa ESP8266 para tal fin.

Los datos una vez recolectados por los sensores pueden ser luego cargados y analizados desde diversas plataformas de visualización, como por ejemplo Alfadatizando [14], una plataforma diseñada para realizar actividades educativas de visualización de datos para desarrollar el Pensamiento Computacional o bien pueden ser visualizados desde una plataforma específica diseñada ad hoc.

Como se mencionó anteriormente es importante proveer a los docentes y alumnos de herramientas accesibles y simples de usar que permitan desarrollar las diferentes aristas de este tipo de pensamiento. En este sentido, Fernandez Quiñonez propone diferentes maneras para programar sistemas IOT usando programación basada en bloques, cuando presenta el uso de placas MicroBit al que le conecta sensores para la toma de datos, almacena localmente los datos para su posterior visualización desde una aplicación móvil que muestra cómo desarrollar usando AppInventor⁷. En este caso, el dispositivo móvil desde el cual se accede a los datos posee activa la conexión vía Bluetooth al igual que la placa MicroBit [15]. Por otro lado, Aneesh Pradeep propone el uso de una placa ESP32 con fines educativos para trabajar aspectos de IOT, recolectando y analizando datos provenientes de sensores. Justifica su elección de esta placa por sobre otras, dado su bajo costo. Sugiere a los profesores que usen proyectos ya existentes para enseñar de IOT con la placa elegido y de cómo realizar conexiones a sensores. Ya que, como menciona el autor, esos proyectos suelen tener instrucciones paso a paso y están diseñados para ser fáciles de seguir. El autor no muestra de manera concreta como se programa la placa ni la manera en la que propone visualizar los datos [16].

⁵ <https://microbit.org/>

⁶ <https://www.arduino.cc/>

⁷ <https://appinventor.mit.edu/>

En este trabajo se propone una infraestructura que combina software y hardware para la toma de datos y su posterior visualización. Con foco en que sea de simple armado y configuración para usuarios sin conocimientos avanzados de informática y para que pueda ser usado en diferentes lugares, por diferentes personas y de manera autónoma. Permitiéndole de este modo diseñar actividades para el desarrollo del Pensamiento Computacional mediante programación basada en bloques, visualización de datos, el uso de placas, sensores y comunicación inalámbrica vía Wifi. El trabajo está organizado de la siguiente manera: en la Sección 2 se presenta el escenario de trabajo, en la Sección 3 un posible esquema de solución, en la Sección 4 se presenta un caso de estudio. En la Sección 5 se presenta una discusión sobre el caso de estudio y finalmente en la Sección 6 conclusiones y trabajos futuros.

2 Escenario de trabajo

En esta sección en primer lugar se esquematiza la problemática a abordar. Como se mencionó previamente, el objetivo de este trabajo es facilitar una infraestructura de trabajo que permita desarrollar habilidades de Pensamiento Computacional abordando aspectos de programación basada en bloques, visualización de datos, manejo de placas, sensores y comunicación inalámbrica. Para poder realizar esto se tomó como primer requerimiento que el envío de datos recolectados se realice vía Wifi a un servidor para tal fin, desde el cual es posible conectarse desde un cliente Web para visualizar los datos obtenidos. La Fig. 1 muestra un esquemático del escenario de trabajo, en donde se consideran diferentes sensores conectados a la placa MicroBit usando cable, una placa EP8266 conectada por cable a la placa MicroBit, la cual es responsable de transmitir datos vía Wifi a un Servidor que los almacena. Luego es posible acceder desde cualquier cliente web al servidor para visualizar gráficamente los datos.

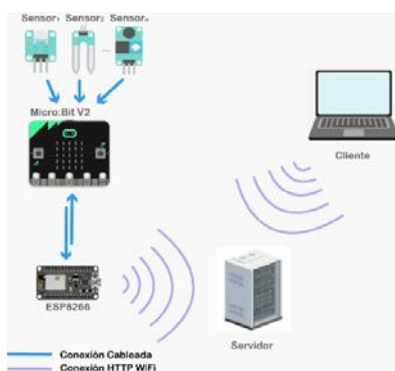


Fig. 1: Esquema del escenario de trabajo

3 Propuesta de solución

Dado que se decidió utilizar placas MicroBit, se tomó como entorno de programación a MakeCode que permite programación usando bloques, Python y JavaScript. El entorno provee familias de bloques para la programación y es posible ampliarlo mediante extensiones. En el caso del uso de Wifi como mecanismo de comunicación inalámbrico, a la fecha de iniciar con esta propuesta, no había una extensión disponible y por tal motivo se decidió proponer una nueva para usar la placa ESP.

Por otro lado, al querer enviar los datos recolectados a un servidor para su posterior visualización, una opción era dejar un servidor dedicado, sin embargo, es objetivo de este trabajo que se use en diferentes lugares, por diferentes personas y de manera autónoma, por esto es por lo que se diseñó e implementó un servidor y visualizador que puede ser desplegado donde el usuario desee siguiendo una serie de pasos detallados.

Se propone en este trabajo a) una extensión para el entorno MakeCode sumando una nueva familia de bloques para encapsular la complejidad técnica de configurar la placa ESP b) un servidor para la recepción, almacenamiento y visualización de datos c) guías para el despliegue del servidor propuesto y para la preconfiguración de la placa ESP

Respecto a la *extensión para el entorno MakeCode*, se logró simplificar la propuesta con los bloques que visualizan en la Fig. 2. Se creó una nueva familia de bloques llamada IRID donde para cada nuevo bloque al posar el cursor sobre el mismo es posible acceder a información de lo que hace y se muestra un ejemplo de valor para sus parámetros.

El bloque *“setup_esp”* permite configurar el nombre de la red a utilizar, la contraseña, la url del servidor y un número que indica la cantidad de veces que se espera la placa intente conectarse al servidor antes de dar un error y avisar que no es posible. El bloque *“send data”* se usa para el envío de datos al servidor. El primer parámetro indica desde donde se reciben los datos, el segundo con que tipo de sensor se toman y por último el dato del valor sensado. Finalmente, se deja disponible un bloque para *“ping”* que puede ser usado para analizar si el servidor está respondiendo.



Fig. 2: Familia de bloques para trabajar con la ESP

Notar que en este trabajo no se hace foco en aspectos de seguridad del servidor, por ello quedarán datos sensibles expuestos.

En cuanto al *Servidor para la recepción, almacenamiento y visualización de datos*, el mismo fue desarrollado con herramientas de código abierto. Cuenta con componente encargado de recibir, analizar y tratar la información enviada por la placa ESP8266 vía Wifi (desarrollado en Python⁸). El servidor al recibir datos, si éstos cumplen con el formato esperado, los almacena (en MongoDB⁹). Los datos quedan disponibles para ser visualizados mediante un componente implementado en el servidor para tal fin (usando Grafana¹⁰).

Por último, se desarrollaron dos *guías*, una relacionada al servidor y otra para preconfigurar la placa ESP. Una primera guía que explica paso a paso como poner en marcha el servidor¹¹ para la recepción, almacenamiento y visualización de datos. La segunda guía¹², dada que la forma de comunicar la placa MicroBit con la placa ESP8266 es mediante el puerto serie de estas, es necesario realizar preconfiguraciones en la placa ESP para usarlos, por tal motivo y por única vez es necesario cargarle esa preconfiguración a la placa.

4 Caso de Estudio

El diseño del caso de estudio se realizó para estudiantes de cuarto año de una escuela secundaria técnica, con orientación en programación de la ciudad de La Plata, Buenos Aires, Argentina. Se decidió organizar el material siguiendo el esquema de usar-modificar-crear para apoyar y profundizar la adquisición de Pensamiento Computacional [17]. Dada la disponibilidad ofrecida desde la escuela, se diseñaron tres encuentros de dos horas treinta minutos cada uno. Para un uso más motivador de la actividad, se cuenta con una maqueta en forma de casa, en la que se encuentran conectados diversos sensores (ej. Temperatura, humedad y gas) y actuadores (ej. luces led, motor que permite abrir puertas y un motor con un ventilador). A esta maqueta la llamamos “Casita”. La Fig. 3 muestra la maqueta armada.



Fig. 3:Maqueta de la “Casita”

⁸ <https://www.python.org/>

⁹ <https://www.mongodb.com/>

¹⁰ <https://grafana.com/>

¹¹ Disponible en <https://zenodo.org/records/13234008>

¹² Disponible en <https://zenodo.org/records/13240796>

En el *primer encuentro*, se presenta el entorno de programación MakeCode se introducen conceptos básicos de la programación en bloque usando la plataforma MakeCode para placas MicroBit. En este encuentro, se espera que los estudiantes adquieran habilidades básicas.

El *segundo encuentro*, suma el uso de sensores externos. Se presenta el esquema de la infraestructura de trabajo y se abordan aspectos de comunicación de datos usando ESP8266, se envían datos a un servidor web usando la programación en bloques y finalmente se visualizan los datos generados por sensores.

El *tercer encuentro*, retoma la actividad del encuentro anterior y se realiza una charla de cierre con los estudiantes a modo de relevamiento.

Para simplificar la programación durante los encuentros, se creó una nueva familia de bloques llamado “Casita” en el que se sumaron los bloques a usar de la familia “IRID” y se creó un bloque especial para que puedan sumarse comentarios. A esta nueva familia de bloques se le configuró la ayuda que permite MakeCode para que, al usar la maqueta, los estudiantes pudieran consultar en que Pin estaba conectado cada sensor o actuador. Lo que haría que funcionara como un kit. La nueva familia de bloques y la ayuda desplegada se visualizan en la Fig.4.

Los ejemplos y ejercicios se dejaron implementados y disponibles en una carpeta, a la que, en caso de ser requerido, se les daba acceso a los estudiantes.

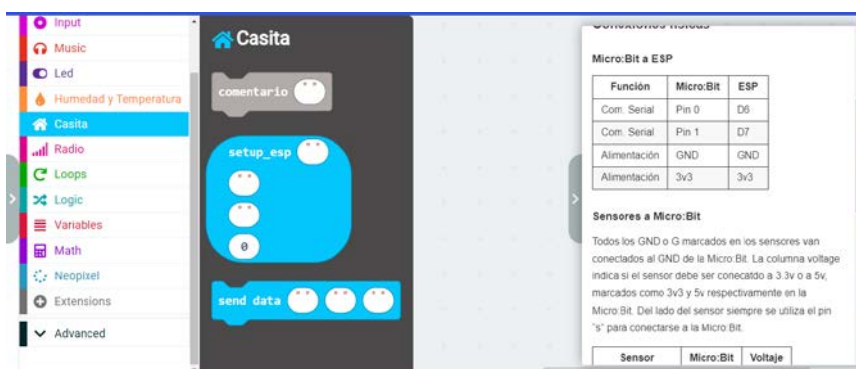


Fig. 4: Familia de bloques para trabajar con la “Casita”

4.3 Propuesta en acción

Para este caso de estudio, considerando la cantidad de encuentros disponibles, si bien se entregaron las guías para preconfiguración de la ESP y para el despliegue de un servidor propio, se decidió llevar la placa ESP preconfigurada, y un servidor ya levantado para que pudiesen directamente acceder al mismo. Los tres encuentros se realizaron durante el mes de noviembre de 2023.

La cantidad de asistentes varió en cada encuentro, en el primer encuentro se presentaron 12 estudiantes y cuatro docentes, en el segundo 15 estudiantes y dos docentes y en el tercero 14 estudiantes tres docentes. Se sumó además de la docente responsable del aula, una docente de apoyo técnico, un docente que imparte

tecnología informática en sexto año del secundario y un docente de otra comisión de cuarto año.

Varios de los estudiantes del curso tenían experiencia en desarrollo de placas Arduino y con lenguajes de programación textual. No todos los estudiantes poseían el mismo nivel de expertise en programación. Cuatro estudiantes, aun no programaban.

El aula disponía de mesas con pcs de escritorio y además se contaba con notebooks que la escuela recibió por el programa Conectar Igualdad¹³ y luego con el plan Juana Manso¹⁴. Lo anterior permitía que cada estudiante dispusiera de un equipo para trabajar, sin embargo, al momento de desarrollar las prácticas, se permitió el trabajo en grupo, con agrupación natural por parte de los estudiantes.

A continuación, la Fig. 5. muestra la manera en la que se toman los datos del sensor de temperatura y se le envían a la ESP para que se encargue de transmitirlos al servidor usando el bloque “send data”.



Fig. 5: Ejemplo de envío de datos de temperatura

Por último la Fig. 6 muestra la visualización de los datos cuando se accede desde un cliente al servidor.

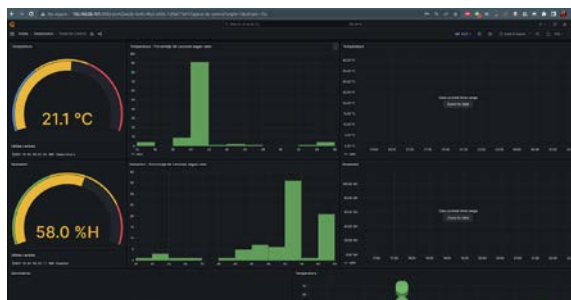


Fig. 6: Visualización de los datos almacenados en el servidor

¹³ <https://conectarigualdad.edu.ar/inicio>

¹⁴ <https://www.educ.ar/noticias/200546/educacioacuten-presenta-la-nueva-versioacuten-de-huayra-50nbsp>

5 Discusión

En esta primera validación de conceptos, el público destinatario estuvo conformado por estudiantes y si bien había docentes participando, éstos no guiaron los encuentros. El curso presentó algunas particularidades tales como la gran disparidad de conocimientos previos entre los estudiantes del mismo curso, así como la actitud frente a las actividades propuestas. En el segundo encuentro, debido a la falta de tiempo, no fue posible trabajar con la visualización de datos, motivo por el cual se retomó en el tercer encuentro. No hubo tiempo suficiente para cumplir con todo el ciclo de usar-modificar-crear propuesto, solo se pudo usar y modificar.

De todos los estudiantes presentes, se mantuvo constante el trabajo solamente con siete de ellos, quienes realizaron las actividades propuestas con interés.

El relevamiento realizado durante el último encuentro con los siete estudiantes que trabajaron en la propuesta mostró su entusiasmo para este tipo de iniciativas quienes destacaron la facilidad de realizar la actividad con programación basada en bloques para la parte de conexión al servidor y el envío de datos. Esta apreciación se debe a que los estudiantes previamente habían programado con la IDE de Arduino durante un taller de robótica educativa.

6 Conclusiones y Trabajos Futuros

En este trabajo se presenta, por un lado, una extensión para el entorno MakeCode para trabajar el envío de datos obtenidos desde sensores de manera inalámbrica (vía Wifi) a un servidor. Se usa una placa MicroBit, una placa ESP8266 y sensores, y se presenta un servidor que está diseñado para que los datos recolectados puedan ser almacenados y visualizados mediante cierto tipo de gráficos. Se diseñaron guías para que sea posible desplegar un servidor propio y para preconfigurar la placa ESP (por única vez). Se diseñó un caso de estudio y se puso en práctica con estudiantes de cuarto año de una escuela secundaria técnica, habiendo, por un lado, obtenido resultados favorables y por otro, recolectado evidencia para mejorar la propuesta.

Si bien se realizaron los tres encuentros planificados, en base a la experiencia resulta recomendable para futuras puestas en práctica, poder realizar una entrevista previa para conocer a los participantes y así refinar el detalle y la cantidad de encuentros, el que, ante la evidencia, debería haber sido al menos uno más para llegar a la etapa en la que los estudiantes crean por sí mismos.

Como trabajo futuro, se espera en el corto plazo sumar otro tipo de gráficos para la visualización de datos, en el mediano plazo, dejar disponibles los datos recolectados para que puedan ser visualizados desde otras plataformas, refinar las guías de trabajo, y las actividades de cada encuentro, sumando, al menos uno de carácter introductorio y opcional que permita reforzar aspectos generales de la programación basada en bloques para el caso de audiencias sin conocimientos previos. Luego de esto se espera realizar nuevas pruebas con estudiantes y docentes de nivel secundario. Se espera también sumar actividades que involucren otras maquetas además de la “Casita”. A largo plazo y una vez alcanzado un grado de madures aceptable de las guías de trabajo y de las actividades de los encuentros, se espera proponer este trabajo como una

actividad de “Hágalo usted mismo” (conocido por la sigla DIY en inglés de *Do it Yourself*), donde el equipo docente pueda configurar la infraestructura con el andamiaje adecuado provisto por las guías y llevar adelante por sí mismo los encuentros con sus estudiantes.

Los autores consideran que la propuesta conforma una posibilidad, no solo para trabajar aspectos de Pensamiento Computacional y visualización de datos con los estudiantes, sino que, además, una manera de contribuir en su formación como ciudadanos digitales.

Agradecimientos. El equipo de trabajo agradece a Jeziel Paladino por ponernos en contacto con la Profesora Sara Neiret de la Escuela de Educación Técnica N°5, General Manuel N. Savio, de la ciudad de La Plata (Buenos Aires, Argentina) a quien también agradecemos por ser quien, de manera muy generosa y comprometida, nos brindó un espacio en una de sus materias a cargo. Sumamos a este agradecimiento a sus estudiantes de 4to año (año 2023).

Referencias

1. Papert, S. (1980). "Mindstorms" Children. Computers and powerful ideas.
2. Wing, J. (2006). Computational thinking. *Communications of ACM*, 49(3), 33-35.
3. Denning, P. J., & Tedre, M. (2021). Computational thinking: A disciplinary perspective. *Informatics in Education*, 20(3), 361.
4. Pereiro, E., Montaldo, M., Koleszar, V., & Urruticoechea, A. (2022). Computational thinking, artificial intelligence and education in Latin America.
5. Lliteras, A. B. (2024). Alfadatizando como parte de la Ciudadanía Digital. In XXVI Workshop de Investigadores en Ciencias de la Computación (Puerto Madryn, 18 y 19 de abril de 2024). En Prensa.
6. Ribble, M. (2008). Passport to digital citizenship. *Learning & leading with technology*, 36(4), 14-17.
7. Özkök, G. A. (2021). Fostering Computational Thinking Through Data Visualization and Design on Secondary School Students. *J. Univers. Comput. Sci.*, 27(3), 285-302.
8. Ezeamuzie, N. O., & Leung, J. S. (2022). Computational thinking through an empirical lens: A systematic review of literature. *Journal of Educational Computing Research*, 60(2), 481-511.
9. Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational technology research and development*, 68(1), 463-484.
10. Udvaros, J., Forman, N., & Avornicului, M. (2023). Developing computational thinking with microcontrollers in Education 4.0.
11. Yim, I. H. Y., & Su, J. (2024). Artificial intelligence (AI) learning tools in K-12 education: A scoping review. *Journal of Computers in Education*, 1-39.
12. Sun, D., Looi, C. K., Li, Y., Zhu, C., Zhu, C., & Cheng, M. (2024). Block-based versus text-based programming: a comparison of learners' programming behaviors, computational thinking skills and attitudes toward programming. *Educational technology research and development*, 72(2), 1067-1089.

13. Pech, J., & Novák, M. (2020). Use Arduino and micro: bit as teaching platform for the education programming and electronics on the stem basis. V International Conference on Information Technologies in Engineering Education (pp. 1-4). IEEE.
14. AlfaDatizando: a Data Visualization Platform to work Computational Thinking in Digital Humanities. Lliteras Alejandra Beatriz, Artopoulos Alejandro, Fernandez Alejandro, Huarte Jimena. LACLO, octubre 2022. In *2022 XVII Latin American Conference on Learning Technologies (LACLO)* (pp. 1-6). IEEE.
15. Fernández Quiñonez, S. L. (2022). Implementación de sistemas iot utilizando técnicas de programación visual.
16. Pradeep, A. (2023). Enabling IoTs with ESP32 for Affordable Education. 5th International Conference on Inventive Research in Computing Applications (pp. 1368-1373). IEEE.
17. Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37.