

Puzles en dispositivos móviles como herramienta de apoyo al aprendizaje de Fundamentos de Programación

(Puzzles on mobile devices as a tool to support the learning of Fundamentals of Programming)

*Ingrid Cela¹, Manne Rodríguez¹, Orlando Erazo¹, Betty Chávez¹

¹ Universidad Técnica Estatal de Quevedo, Quevedo - Ecuador, ingrid.cela2013@uteq.edu.ec, manne.rodriguez2013@uteq.edu.ec, oerazo@uteq.edu.ec, bchavez@uteq.edu.ec

Recibido: 05/09/2022. Aceptado: 25/12/2022

Publicado: 27/12/2022

Resumen: Existe una búsqueda constante de más y nuevas herramientas que ayuden a enfrentar la problemática que implica fomentar la motivación para adquirir conocimientos y desarrollar habilidades en áreas o asignaturas complejas como lo es la programación. Una opción son los juegos serios, que se han convertido en una herramienta cada vez más utilizada en educación como un apoyo al proceso de enseñanza-aprendizaje, ayudando a los estudiantes a adquirir o reforzar los conocimientos de las asignaturas de manera entretenida. Con esta referencia, en este trabajo se propone el uso de puzles como un apoyo en el aprendizaje de los fundamentos de programación. Para ello, se diseñó un prototipo de juego serio basado en puzles para dispositivos móviles. El prototipo fue evaluado con la participación de estudiantes que cursaban un primer nivel de programación. Los resultados de esta evaluación fueron positivos, sugiriendo los participantes que el prototipo era atractivo y fácil de utilizar. Esto permite concluir que el empleo de puzles en dispositivos móviles al aprender programación por primera vez es una idea válida que puede apoyar el aprendizaje.

Palabras Clave: Rompecabezas, aprendizaje de programación, juegos serios, aprendizaje móvil.

Abstract: There is a constant search for more and new tools that help to face the problem that implies fostering motivation to acquire knowledge and develop skills in complex areas or subjects such as programming. One option is serious games, which have become an increasingly used tool in education as a support to the teaching-learning process, helping students to acquire or reinforce the knowledge of the subjects in an entertaining way. With this reference, this work proposes the use of puzzles as help in learning the fundamentals of programming. For this, a prototype of serious game based on puzzles for mobile devices was designed. The prototype was evaluated with the participation of students who were attending a first level of programming. The results of this evaluation were positive, with the participants suggesting that the prototype was attractive and easy to use. This fact allows concluding that the use of puzzles on mobile devices when learning programming for the first time is a valid idea that can support learning.

Keywords: Puzzles, programming learning, serious games, mobile learning.

Introducción

El aprendizaje de fundamentos de programación es considerado frecuentemente una tarea difícil que acarrea consecuencias negativas. Se trata de un problema con el que los académicos responsables siguen luchando. De hecho, hay un sinnúmero de factores que influyen en ello, entre los que se pueden mencionar la complejidad de la sintaxis de cada lenguaje de programación, la carga cognitiva que se produce en la persona interesada, y la falta de habilidades propias para la solución de problemas (Cheah, 2020; Insuasti, 2016). Aspectos como estos pueden tener como consecuencia el desinterés de parte de los aprendices, e incluso en muchos casos llevan a la deserción de los estudiantes (Sinchi Nacipucha & Gómez Ceballos, 2018).

Ante esta situación, y dada la demanda e importancia de la programación, se han planteado diferentes opciones para apoyar su proceso de enseñanza-aprendizaje. Una opción apunta a fomentar en primera instancia el desarrollo del pensamiento computacional como paso previo a la programación propiamente dicha. Este tiene como intención estimular la creatividad y desarrollar las habilidades de los estudiantes en la resolución de problemas (Gamito, Aristizabal, Basasoro, & León, 2022) mientras que la programación constituye una herramienta que permite poner en práctica el pensamiento computacional. También se han realizado estudios sobre métodos para la enseñanza de fundamentos de programación que proponen como medida de solución crear materiales didácticos, interactivos, atractivos, poco complejos y novedosos (Samar I.Swaid, 2015).

En este sentido, se contempla también la posibilidad de crear juegos digitales donde los estudiantes pueden practicar y desarrollar sus habilidades en pensamiento computacional (Cagin Kazimoglu, Mary Kiernan, Liz Bacon and Lachlan MacKinnon, 2012) (Kyu Han Koh, Ashok Basawapatna, Vicki Bennett, Alexander Repenning, 2010). Los juegos están experimentando una gran evolución en los últimos años debido a que son considerados como herramientas prácticas y motivadoras que permiten a los estudiantes desarrollar habilidades más allá de las que adquieren en sus lugares de estudio. Algunos juegos populares en el ámbito del pensamiento computacional y programación son Code Combat (CodeCombat Inc. , s.f.), Programming Hero (Programming Hero, s.f.), Code Monkey (CodeMonkey, s.f.), y muchos más. Estos juegos han sido creados pensando en niños desde los 3 años para ayudarles a desarrollar sus habilidades, hasta los estudiantes de educación superior que presentan conflictos al momento de comprender los fundamentos de programación.

Todas estas soluciones han sido evaluadas acorde al tipo de aplicación y al nivel de complejidad que representan, dando como resultado una gran aceptación entre ellos. También, muchas de ellas disponen de una versión para dispositivos móviles, lo que permite incluso apuntar hacia el aprendizaje móvil (Criollo-C, Guerrero-Arias, Jaramillo-Alcázar, & Luján-Mora, 2021). Sin embargo, algunas de las aplicaciones son pagadas, otras no cuentan con la opción de cambio de idioma ya que solo están disponibles en inglés, algunas presentan opciones con funciones que no son fáciles de identificar, otros no son compatibles para varios sistemas operativos, entre otras desventajas. Además, hay que tener presente que herramientas como las mencionadas no necesariamente se centran en la sintaxis del lenguaje, optando más bien por el razonamiento en general.

Con lo anteriormente dicho, en este trabajo se apuesta por el enfoque lúdico como se ha mencionado, pero concentrándose más en los fundamentos de programación y por ende en

un lenguaje de programación. Por ende, su finalidad es analizar la posibilidad de emplear una herramienta popular, como lo son los puzles, para apoyar el aprendizaje de fundamentos de programación a través del juego. Contrario a lo propuesto en otros tipos de aplicaciones (como por ejemplo (Parsons & Haden, 2006), (Schuster, Kalyan, Polozov, & Kalai, 2021), (Bender, Zhao, Dziena, & Kaiser, 2022)), la idea es explorada con el uso de un prototipo de aplicación móvil, procurando darle un enfoque práctico, atractivo e interactivo. Este prototipo ha sido evaluado inicialmente, dando resultados a favor de la idea propuesta.

Metodología

Entrevista previa

Inicialmente se dispuso de varias opciones de los posibles tipos de juegos que podrían ayudar a practicar fundamentos de programación a los estudiantes. Esto presentó un problema ya que se tenía que seleccionar el que más se ajuste a las necesidades de los usuarios sin que consista en un juego complejo y que presente una interfaz amigable. Por ello, para el diseño del prototipo de aplicación móvil se realizó una entrevista a tres docentes de la misma institución de los autores. Dos fueron de género femenino y uno masculino, todos tenían conocimientos en programación e impartían clases en asignaturas que comprenden la enseñanza de programación. En la entrevista se reflexionó sobre problemas comunes al aprender programación y la idea de usar juegos para dispositivos móviles para apoyar el aprendizaje.

Construcción del prototipo

Con un panorama más claro de la propuesta, se procedió a construir un prototipo que incluya los detalles necesarios. Se optó por un juego basado en puzles; es decir, el juego presenta porciones de código desordenado que debe ser organizado o “armado” correctamente para considerar el puzle resuelto. El código fue escrito utilizando el lenguaje Microsoft Visual C#. Siguiendo un enfoque de prototipado, se logró disponer de una versión funcional lista para la evaluación necesaria.

Prueba piloto

Se realizó una prueba piloto con la participación de tres estudiantes que se encontraban cursando el séptimo semestre de la carrera de Ingeniería en Telemática al momento de efectuar el estudio. Los participantes utilizaron libremente la aplicación, proporcionando los comentarios necesarios en voz alta. Esto permitió realizar ajustes como la emisión de sonidos al seleccionar las piezas del puzle, incentivar al jugador con puntos por cada nivel jugado correctamente y restar puntos por cada intento fallido. Por último, esta prueba ayudó también a verificar que el prototipo funcione correctamente y afinar los últimos detalles para la evaluación.

Participantes

Para realizar la evaluación se contó con la participación de quince estudiantes con edades entre 17 y 23 años, con una media de 18.87 y una desviación estándar de 1.73. De los participantes 14 corresponden al género masculino y sólo 1 persona fue de género femenino. El 100% de los participantes mencionaron que cuentan con smartphone, compatibles con el sistema operativo Android y que lo usan de 2 a 13 horas diarias para comunicarse con sus familiares y amigos, acceder a redes sociales, tomar y editar fotos, jugar, escuchar música, ver videos e investigar sobre temas que no entienden en las tareas educativas. De igual forma, la mayoría de los participantes respondieron que les gustan los juegos de estrategia, acción, aventura, deportes, acertijos y simulación; solo un participante mencionó lo contrario. Por otra parte, los participantes respondieron que su nivel en fundamentos de programación era básico porque solo contaban con

el aprendizaje elemental que recibieron en la secundaria y/o en la universidad. Entre los lenguajes de programación que habían utilizado estaban C# y Java. Por último, hasta la fecha de realización de la evaluación, todos los participantes se encontraban cursando por primera ocasión el primer nivel de la carrera Ingeniería en Software perteneciente a la facultad y universidad de los autores.

Lugar de evaluación

La evaluación se realizó en una sala de clases de la institución de los autores de este trabajo. Se decidió realizarla en este lugar porque los participantes eran alumnos de la institución y se encontraban cursando una carrera orientada a adquirir conocimientos en fundamentos de programación. Además, el lugar proporcionaba un ambiente controlado para evitar interrupciones que distraigan a los participantes, puesto que el objetivo es obtener resultados más confiables en la evaluación.

Procedimiento

La evaluación inició con una breve introducción acerca de la funcionalidad del prototipo de puzzles para practicar el aprendizaje de fundamentos de programación. Luego se emitieron algunas instrucciones a los participantes como que proceda a sentarse cómodamente en la silla para iniciar la utilización de la aplicación. Los participantes fueron instruidos para interactuar libremente con la aplicación por un tiempo de diez minutos. Transcurrido este periodo, se procedió a solicitar información contestando un cuestionario previamente preparado. Además, para constancia del trabajo realizado, los estudiantes firmaron un consentimiento informado.

Recolección de datos

La recolección de datos se efectuó mediante un cuestionario preparado para el efecto. Estuvo compuesto de tres partes. En la primera parte se solicitaron datos demográficos como edad, sexo, nivel educativo, uso de celular, atracción por los juegos y conocimientos de programación. La segunda parte se preparó adaptando el cuestionario denominado "System Usability Scale" (SUS) (Bangor, Kortum, & Miller, 2008) con un total de diez preguntas basadas en una escala de Likert de uno a cinco puntos. Por último, se incluyeron cinco preguntas: ¿Cree usted que esta aplicación puede ayudarle a practicar el aprendizaje de fundamentos de programación? ¿Considera usted que a esta aplicación le hace falta algo? (¿Qué?) ¿Recomendaría usted esta aplicación a otras personas interesadas en mejorar su aprendizaje de programación? ¿Se motivará usted a aprender programación utilizando esta aplicación? ¿Cuál cree que es el tipo de persona a la que está destinada idealmente esta aplicación?

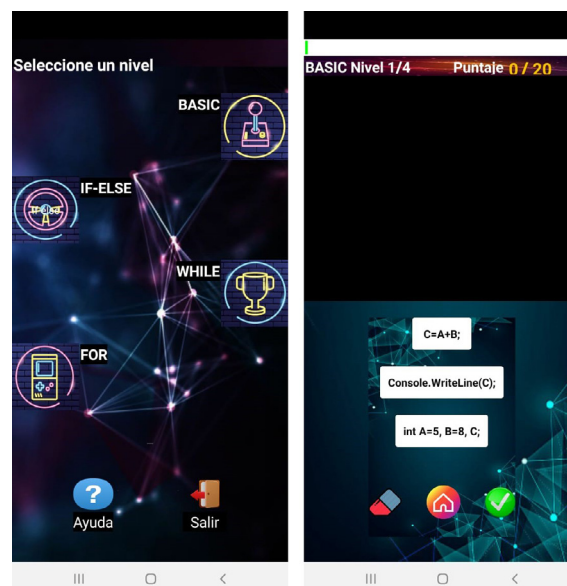
Resultados

A partir de las entrevistas iniciales se obtuvieron algunas recomendaciones para la propuesta. Los docentes mencionaron que los problemas más comunes en el aprendizaje de programación son la falta de interés porque los estudiantes suelen pensar que es algo complejo, al no poder comprender la lógica por la falta de asociación de expresiones que puedan representar un problema. Así, por ejemplo, los alumnos no distinguen entre una variable y una constante o entre una función y un procedimiento. Los entrevistados también piensan que la idea de usar juegos es buena porque los alumnos tendrían una herramienta que les permitirá practicar programación, debido a que a los jóvenes les gusta jugar y creen que mientras la mente más ocupada está, más se desarrolla. De esta forma, los alumnos lograrían encontrar el interés necesario para adentrarse en este medio, ayudándoles a adquirir conocimientos de manera interactiva y a relacionar las actividades que pueden realizar con valores, procedimientos o funciones.

Todos los participantes concordaron también en que el tipo de juego apropiado es el de estrategia porque permite pensar, relacionar y ver más allá de lo que pueden hacer, ayudando a los aprendices a tener organización de ideas. Solo una persona opinó que también sería bueno un juego de acción porque en la actualidad es el tipo de juego que más les llama la atención a los jóvenes. También, uno de los docentes opinó que el ambiente del juego debería tener colores cálidos para que el cerebro del jugador no se canse, con una interfaz sencilla y amigable, dependiendo del tema de programación para que llame la atención. Por otra parte, otro docente consideró que el software debe tener estrategias en la administración de movimientos, porque le invita a pensar en la decisión que va a tomar. Finalmente, una docente mencionó que el juego debería contar con herramientas que se puedan utilizar para crear una estrategia de solución a algún problema o para alcanzar una meta. De esta forma, los resultados permitieron confirmar que la idea más apropiada para satisfacer las necesidades de los estudiantes es la de utilizar juegos de estrategia como los basados en puzzles de forma que le permita al usuario armar un código de solución con solo elegir las opciones correctas en el orden correcto.

La Figura 1 muestra de forma representativa dos pantallas del prototipo construido como parte de la propuesta. Este integra rompecabezas de código, con un enfoque de juego serio (Susi, Johannesson, & Backlund, 2007) (Girard, Ecalle, & Magnan, 2013) que permite armarlos seleccionando las partes en el orden correcto. El juego cuenta de cuatro niveles asociados a las sentencias elementales de programación: básico (variables y asignaciones), if-else, while y for. Cada uno de los niveles consta de varias bases en las que el jugador va a ganar puntos conforme vaya avanzando. Durante el juego, los usuarios necesitan realizar la elección de piezas que les permita armar un código de solución en el orden correcto. Para empezar a armar el código de solución, el usuario debe tocar la opción que desea elegir y el contenido de esta se va a ir mostrando en la parte superior, simulando un ambiente de programación. Para completar cada nivel, los jugadores necesitan jugar cuatro bases con distintos tipos de código (o ejercicios) que les permitirá pasar al siguiente nivel. El juego recompensa a los jugadores mediante la asignación de cinco puntos por cada base.

Figura 1. Menú principal de la aplicación (izquierda) y ejemplo de puzzle básico (derecha)



El prototipo desarrollado se evaluó para analizar las posibilidades de ser usado con el fin propuesto. Para el efecto, los participantes respondieron diez preguntas después de interactuar con él. Los puntajes promedio obtenidos para cada uno de los ítems del cuestionario se presentan en la Tabla 1. Como puede notarse, el cuestionario SUS presenta los ítems impares en forma positiva y los pares en forma negativa. Observando los puntajes de cada ítem se puede notar que en general se obtuvo buenas puntuaciones; es decir, cercanas a 5 para los impares y cercanas a 1 para los pares. Puede considerarse como excepción el último ítem porque los participantes podrían haber pensado que las “muchas cosas que necesitan aprender antes de utilizar la aplicación” corresponden al aprendizaje de los fundamentos de programación.

Tabla 1. Promedios obtenidos por cada ítem del cuestionario SUS.

Ítems	Media
1.- Pienso que la aplicación puede ayudarme a aprender programación usando la aplicación frecuentemente.	4.40
2.- Encontré la aplicación innecesariamente compleja.	1.87
3.- Pienso que la aplicación fue fácil de usar.	4.67
4.- Necesitaría apoyo de un técnico/profesor para utilizar la aplicación.	1.6
5.- Encontré que las diferentes funciones de la aplicación fueron bien integradas.	4.27
6.- Pienso que había demasiadas inconsistencias en la aplicación.	2.20
7.- Imagino que la mayoría de las personas aprenderían a usar rápidamente la aplicación.	4.60
8.- Encontré la aplicación muy difícil de usar.	1.80
9.- Me sentí muy seguro/cómodo usando la aplicación.	4.53
10.- Necesito aprender muchas cosas antes de utilizar la aplicación.	2.73

Por otro lado, se procedió también a calcular el puntaje SUS de cada uno de los participantes a partir de los puntajes individuales. Esto se realizó mediante la metodología del cuestionario (Bangor, Kortum, & Miller, 2008). El puntaje promedio obtenido fue de 80,67. Este puntaje puede ser expresado mediante un adjetivo siguiendo la misma metodología. Esto es, el prototipo empleado tiene un grado de usabilidad de bueno a excelente, lo que se considera como aceptable. En definitiva, los resultados de usabilidad indican que el prototipo tuvo un gran impacto y acogida por parte de los participantes.

En cuanto a las preguntas adicionales, los resultados también fueron favorables. Para la primera pregunta, el 100% de los participantes manifestaron que la aplicación sí podría ayudarles a practicar el aprendizaje de fundamentos en programación. Entre las razones mencionadas están: posee instrucciones básicas de un lenguaje estructurado que enseña cómo utilizar los ciclos de repetición, evita tener que usar computadoras y permite aprender rápido, es un juego intuitivo, es fácil de usar.

En la segunda pregunta, el 33.33% de los participantes mencionaron que a la aplicación no le hace falta nada ya que cuenta con una interfaz agradable y lo esencial para poder reforzar el aprendizaje. No obstante, el 66.67% consideró que a la aplicación le hace falta contar con bloques que se puedan arrastrar para ir formando el código de solución, algo similar a la interfaz que

presenta Scratch (SCRATCH, s.f.); que el juego debería mostrar una introducción y conceptos que sirvan como guía en cada nivel. Además, los participantes sugirieron que se debería agregar más niveles en el juego y que convendría hacer desplazable la pantalla en las bases que presentan varias opciones para armar el puzle.

En tercer lugar, la totalidad de los participantes respondió que sí recomendarían la aplicación ya que constituye una buena idea utilizar rompecabezas para apoyar el desarrollo del pensamiento computacional y aprendizaje de programación y porque les permitió comprender las estructuras de los ciclos de repetición. De igual manera, en la cuarta pregunta todos los alumnos dijeron que sí se motivarían a practicar programación utilizando la aplicación dado que la idea de utilizar juegos les incentivó a seguir aprendiendo y mencionaron que mientras más práctica se realice, mejores resultados obtienen.

Por último, los resultados acerca de la pregunta que se les realizó a los participantes sobre el tipo de persona a la que creen que está destinado idealmente el software se muestran en la Tabla 2. En esta tabla se observa que se obtiene una cantidad de votos mayor a la cantidad de participantes debido a que no se limitó el elegir una sola opción como respuesta. La mayoría de los participantes consideró que el prototipo está pensado para adolescentes que se encuentran cursando el bachillerato debido a que es desde ahí donde se debería empezar a desarrollar la lógica en el pensamiento computacional. El 23.81% consideró que es para adultos jóvenes porque les permitirá practicar en cualquier lugar los conceptos impartidos por los docentes sin necesidad de contar con una computadora (una importante ventaja del aprendizaje móvil). El 19.05% piensan que deberían utilizar el prototipo los niños de primero a séptimo de básica ya que cuenta con una interfaz sencilla, atractiva y fácil de usar. La opción de que la aplicación esté destinada para niños que aún no van a la escuela pasó desapercibida y solo una persona mencionó que debería ser dirigida para todo tipo de usuarios porque es un prototipo educativo que podría ayudar a mejorar la lógica de programación.

Tabla 2. Tipo de persona a la que podría estar destinado el prototipo

Tipo de persona	Cantidad	%
Niños que aún no van a la escuela	0	0%
Niños de primero a 7mo de básica	4	19.05%
Adolescentes (bachilleres)	11	52.38%
Adultos jóvenes (universitarios)	5	23.81%
Otro	1	4.76%

Conclusiones

Frente a la problemática que ocasiona el aprendizaje de Fundamentos de Programación (Cheah, 2020; Insuasti, 2016), provocando en muchos casos el desinterés y la deserción (Sinchi Nacipucha & Gómez Ceballos, 2018), en este trabajo se propone apostar por juegos de razonamiento para dispositivos móviles como un complemento o ayuda al proceso de enseñanza-aprendizaje. Para ello, se partió analizando la idea con profesionales del área y mediante entrevistas se determinó su factibilidad inicial. Luego, se diseñó un prototipo de juego que integra rompecabezas de código, como una forma de apoyar el aprendizaje en fundamentos de programación. Este permite armar

rompecabezas de códigos de solución a problemas varios, mediante una interfaz sencilla, amigable y fácil de aprender a usar. Esto se evidenció mediante una evaluación de usabilidad del juego con un grupo de estudiantes de la carrera de Ingeniería en Software que se encontraban cursando la asignatura de correspondiente a los fundamentos de programación. Los resultados permitieron calcular el puntaje SUS que corresponde a una usabilidad aceptable, junto con comentarios positivos de los participantes. En el global, los participantes sugieren que este enfoque podría ayudar a desarrollar las habilidades de resolución de problemas de los estudiantes que están aprendiendo las bases de programación.

Aunque la idea ha sido evaluada favorablemente, el prototipo usado no es un producto final quedando abierta la posibilidad de evolucionar para llegar a ese punto. Para el efecto, se pueden considerar diferentes posibilidades. Una idea consiste en que el software ofrezca material didáctico que sirva como introducción a cada tema para aquellos alumnos que no tienen ninguna base de programación relacionada y que están por adentrarse en esta área. Para ello, se puede también recurrir a la posibilidad de proporcionar un video introductorio por cada nivel y a la vez un ejemplo con un ejercicio y su código de solución correspondiente, lo que permitiría a los usuarios tener una idea de lo que van a tener que realizar en cada nivel. Finalmente, el software podría considerar dos versiones, una para el estudiante y otra para los docentes, de manera que este último pueda configurar los puzzles o ejercicios según su necesidad. Esta idea incluso podría abrir la posibilidad de usar la aplicación en otros niveles.

Por otro lado, en el futuro podría considerarse también la posibilidad de recurrir a algún otro paradigma de interacción emergente. Un ejemplo de esto es la interacción basada en gestos que puede ser usada en aplicaciones para complementar el proceso de enseñanza-aprendizaje (Vicuña & Erazo, 2016). Esta idea ha sido probada en otros temas/áreas (Zúñiga Paredes, Rodríguez-Chérrez, Yáñez-Moreira, & Buste, 2020), pero podría emplearse para resolver los puzzles frente a una pantalla mediante el movimiento de las manos en el aire para realizar un gesto que permita seleccionar el elemento o porción de código que corresponda (Erazo, Vicuña, Pico, & Oviedo, 2018).

Referencias

- Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human-Computer Interaction*, 24(6), 574-594.
- Bender, J., Zhao, B., Dziena, A., & Kaiser, G. (2022). Learning Computational Thinking Efficiently: How Parsons Programming Puzzles within Scratch Might Help. *Australasian Computing Education Conference*, (págs. 66-75).
- Cagin Kazimoglu, Mary Kiernan, Liz Bacon and Lachlan MacKinnon. (2012). Learning Programming at the Computational Thinking Level via Digital Game-Play. *ScienceDirect*, 9, 522-531.
- Ceah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, 12(2). doi:https://doi.org/10.30935/cedtech/8247
- CodeCombat Inc. . (s.f.). *Code Combat*. Obtenido de <https://codecombat.com/>
- CodeMonkey. (s.f.). *Code Monkey*. Obtenido de <https://www.codemonkey.com/>
- Criollo-C, S., Guerrero-Arias, A., Jaramillo-Alcázar, Á., & Luján-Mora, S. (2021). Mobile learning technologies for education: Benefits and pending issues. *Applied Sciences*, 11(9). doi:https://doi.org/10.3390/app11094111

- Erazo, O., Vicuña, A., Pico, R., & Oviedo, B. (2018). Analyzing Mid-Air Hand Gestures to Confirm Selections on Displays. *International Conference on Technology Trends. Springer, Cham.*, (págs. 341-352).
- Gamito, R., Aristizabal, P., Basasoro, M., & León, I. (2022). El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con Scratch. *Innoeduca. International Journal of Technology and Educational Innovation*, 8(1), 59-74. doi:<https://doi.org/10.24310/innoeduca.2022.v8i1.12093>
- Girard, C., Ecalle, J., & Magnan, A. (2013). Serious games as new educational tools: how effective are they? *A meta-analysis of recent studies. Journal of computer assisted learning*, 29(3), 207-219.
- Insuasti, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Revista educación y desarrollo social*, 10(2), 234-246.
- Kyu Han Koh, Ashok Basawapatna, Vicki Bennett, Alexander Repenning. (2010). Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning. *Simposio IEEE 2010 sobre lenguajes visuales y computación centrada en el ser humano*. Leganés, España.
- Parsons, D., & Haden, P. (2006). Parson's programming puzzles: a fun and effective learning tool for first programming courses. *Proceedings of the 8th Australasian Conference on Computing Education*, (págs. 157-163).
- Programming Hero. (s.f.). *Programming hero*. Obtenido de <https://www.programming-hero.com/>
- Samar I.Swaid. (2015). Bringing computational thinking to STEM education. *ScienceDirect*, 3, 3657-3662.
- Schuster, T., Kalyan, A., Polozov, O., & Kalai, A. T. (2021). Programming puzzles. *35th Conference on Neural Information Processing Systems*.
- SCRATCH. (s.f.). Obtenido de <https://scratch.mit.edu/>
- Sinchi Nacipucha, E. R., & Gómez Ceballos, G. P. (2018). Acceso y deserción en las universidades. Alternativas de financiamiento. *ALTERIDAD. Revista de Educación*, 13(2), 274-287. doi:<https://doi.org/10.17163/alt.v13n2.2018.10>
- Susi, T., Johannesson, M., & Backlund, P. (2007). *Serious games: An overview*. Technical Report HS- IKI -TR-07-001.
- Vicuña, A., & Erazo, O. (2016). Interacción basada en gestos de dibujos para complementar el proceso de enseñanza-aprendizaje. *Enfoque UTE*, 7(4), 1-15.
- Zúñiga Paredes, A., Rodríguez-Chérrez, M., Yáñez-Moreira, M., & Buste, B. (2020). Interacción basada en gestos para mejorar la ortografía. *Revista InGenio*, 3(1), 102-113.