

DOI: <https://doi.org/10.23913/pag.v9i18.899>

Una experiencia del aprendizaje móvil como apoyo para el estudio de la programación

An Experience of Mobile Learning as a Support for the Study of Programming

Uma experiência de aprendizagem móvel para apoiar o estudo de programação

Irene Aguilar Juarez

Universidad Autónoma del Estado de México, México

iaguilarj@uaem.mx

<https://orcid.org/0000-0003-4747-0336>

Bernardo Alexis Rojas Espinoza

Universidad Autónoma del Estado de México, México

rojasespinozabernardo@gmail.com

<https://orcid.org/0000-0001-9879-1143>

Joel Ayala de la Vega

Universidad Autónoma del Estado de México, México

joelayala2001@yahoo.com.mx

<https://orcid.org/0000-0003-3279-4143>

Resumen

Este escrito muestra el desarrollo y evaluación de una *app* orientada a apoyar el aprendizaje de los principios de la programación orientada a objetos. Para el desarrollo de este estudio se usó una metodología en dos fases: por un lado, el desarrollo del *software* y por otro, la evaluación de la percepción del aprendizaje de los alumnos. En la fase del desarrollo de la *app* se separaron las actividades en dos dimensiones: la dimensión didáctica y la dimensión tecnológica. Se empleó el aprendizaje móvil (*m-learning*) como medio comunicativo-interactivo para promover el interés y la motivación de los alumnos en el diseño instruccional; respecto a la dimensión tecnológica, se eligió al paradigma de desarrollo en cascada con fases solapadas. En la etapa de evaluación de la aplicación se

usó un cuestionario cerrado para conocer la percepción del alumnado referente a la funcionalidad y utilidad de la *app*. Los resultados obtenidos de la evaluación mostraron que el manejo didáctico fue apropiado en la *app* y permitió promover el interés para que los estudiantes aprendieran los fundamentos de la programación. También muestran una buena aceptación del alumnado hacia la aplicación desarrollada y subrayan la necesidad de nuevas y mejores aplicaciones.

Palabras clave: aprendizaje móvil, codificación, lenguaje de programación.

Abstract

This paper shows the development and evaluation of an app oriented to support the learning of object-oriented programming principles. For the development of this study, a two-phase methodology was used: on the one hand, the development of the software and on the other hand, the evaluation of the students' perception of learning. In the app development phase, the activities were separated into two dimensions: the didactic dimension and the technological dimension. Mobile learning (m-learning) was used as a communicative-interactive medium to promote students' interest and motivation in the instructional design; regarding the technological dimension, a cascade development paradigm with overlapping phases was chosen. In the application evaluation stage, a closed-ended questionnaire was used to determine the students' perception of the app's functionality and usefulness. The results obtained from the evaluation showed that the didactic management of the app was appropriate and allowed promoting the interest of the students to learn the basics of programming. They also show a good acceptance of the students towards the developed application and underline the need for new and better applications.

Keywords: mobile learning, coding, programming language.

Resumo

Esta escrita mostra o desenvolvimento e avaliação de um aplicativo que visa apoiar o aprendizado dos princípios da programação orientada a objetos. Para o desenvolvimento deste estudo, foi utilizada uma metodologia em duas fases: por um lado, o desenvolvimento do software e por outro, a avaliação da percepção da aprendizagem dos alunos. Na fase de desenvolvimento do aplicativo, as atividades foram separadas em duas dimensões: a dimensão didática e a dimensão tecnológica. A aprendizagem móvel (m-learning) foi utilizada como meio comunicativo-interativo para promover o interesse e a motivação dos alunos no design instrucional; Quanto à dimensão tecnológica, optou-se

pelo paradigma de desenvolvimento em cascata com fases sobrepostas. Na etapa de avaliação do aplicativo, foi utilizado um questionário fechado para conhecer a percepção dos alunos quanto à funcionalidade e utilidade do aplicativo. Os resultados obtidos com a avaliação mostraram que a gestão didática foi adequada no aplicativo e permitiu promover o interesse para que os alunos aprendessem os fundamentos da programação. Mostram também uma boa aceitação dos alunos em relação à aplicação desenvolvida e sublinham a necessidade de novas e melhores aplicações.

Palavras-chave: aprendizagem móvel, codificação, linguagem de programação.

Fecha Recepción: Enero 2022

Fecha Aceptación: Julio 2022

Introducción

La computadora se ha convertido en una herramienta indispensable en las tareas profesionales de todas las áreas; por esta razón se ha incrementado la demanda de desarrolladores de *software* con habilidades sobresalientes en la resolución de problemas de todo tipo que se integren fácilmente en el proceso industrial.

Es muy importante lograr que los alumnos universitarios comprendan y apliquen los procesos necesarios en la creación de algoritmos eficaces desde los primeros semestres, ya que son el pilar para desarrollar programas funcionales con los distintos paradigmas de programación. No obstante, en las aulas el aprendizaje de la programación representa para los alumnos un reto difícil de lograr, pues algunos de ellos inician su contacto con la programación con nulo conocimiento previo de los fundamentos de la programación, apenas con un uso básico de la computadora y el Internet. Ante esta situación es necesario abordar la enseñanza de la programación aprovechando la tecnología educativa para mejorar los resultados de aprendizaje de los alumnos de nuevo ingreso.

Iniciar con el aprendizaje de la programación exige para el alumno abordar varios conocimientos de forma simultánea, tales como:

- Uso de la lógica computacional en el diseño de algoritmos para la solución de problemas.
- Uso de *software* de modelado de algoritmos y de diseño de *software*.
- Uso de los conceptos generales de la programación (código fuente, estructuras de control, uso de distintas variables, etc.
- Uso de los entornos de desarrollo como herramientas para codificar algoritmos (deben aprender sobre el proceso de edición de programas fuente, compilación, prueba y proceso de ejecución de programas).

- Operaciones relacionadas con la programación independiente del paradigma de programación (asignación de datos, invocación y declaración de procedimientos o funciones y ámbito de variables; paso de parámetros y uso de argumentos).

Además, los alumnos deben comprender conceptos específicos de un paradigma, ya sea de la programación estructurada o de la programación orientada a objetos. Respecto a la programación orientada a objetos, algunos de los conceptos a aprender son:

- La metodología de desarrollo de las aplicaciones orientadas a objetos.
- El lenguaje de modelado para el diseño de las aplicaciones.
- Las reglas de sintaxis y semántica de un lenguaje de programación
- Instanciación, polimorfismo, herencia, casteo de objetos, invocación de métodos, recolección de basura, diferentes tipos de clases, uso de excepciones, etc.
- Los conceptos relacionados con el uso de patrones de diseño, reúso de código y trabajo colaborativo

Son muchos los beneficios de que los alumnos aprendan el paradigma orientado a objetos, sobre todo las habilidades para su desempeño en entornos profesionales, entre ellos la reutilización del código, el mantenimiento del *software*, la construcción de sistemas complejos *software*, el uso de arquitecturas complejas usadas en problemáticas reales, el uso de programas con interfaces visuales y sobre todo el desarrollo de *software* sencillo de mantener (Paredes, 2021).

Aprender a programar es complejo porque no solo se trata de memorizar conceptos, escribir código y echarlo a andar; se trata de comprender y aplicar estructuras lógicas y las técnicas computacionales en la solución de los problemas desde un paradigma de programación.

Las situaciones que dificultan la enseñanza de la programación en el nivel superior son, entre otras:

- El uso de estrategias didácticas poco eficientes centradas en la memorización de conceptos o técnicas, pero no centradas en el desarrollo del pensamiento lógico.
- Planes de estudio con temarios amplios y complejos para alumnos sin conocimientos previos, pues frecuentemente los alumnos ingresan al nivel superior solo con conocimientos básicos sobre el uso de la computadora.
- Asignación de poco tiempo a las clases durante semestres o cuatrimestres (en promedio cuatro horas a la semana)
- Actividades de enseñanza con énfasis en la codificación en un lenguaje de programación y poca atención al desarrollo de la lógica computacional.

Ante la dificultad de la materia, frecuentemente se observa que en los estudiantes de cursos o asignaturas de programación surge un desinterés por la materia, lo que genera problemas de aprendizaje y una tasa alta de estudiantes reprobados y desertores (Zatarain, 2018).

Por otro lado, el diseño versátil de los recursos educativos es muy importante por su función mediadora, ya que bien diseñados los materiales educativos promueven el interés, la motivación y además la utilización de la multimedia ayuda a que los alumnos aprendan haciendo uso de diferentes formatos tales como imágenes, sonidos, videos. También es posible aprovechar el tipo de aprendizaje que ellos prefieran. Estos recursos representan un sistema de símbolos que son textuales o icónicos (Criollo, 2017).

En consideración de estas situaciones nos planteamos el objetivo de proveer a los alumnos de un material de estudio que incentive su interés por la programación, que disminuya su frustración ante sus dudas y que mantenga su motivación. Se aprovechan las tecnologías de comunicación actuales para desarrollar una *app* móvil que introduzca a los alumnos en la ardua tarea de comprender el proceso de la programación orientada a objetos.

Trabajos previos

La dificultad de aprender a programar se relaciona con el alto nivel de abstracción que algunos conceptos de programación requieren. Para disminuir la dificultad que enfrenta el alumno, promover la motivación y facilitar el aprendizaje de los lenguajes de programación, las instituciones académicas han desarrollado o usado varias tecnologías, entre ellas:

- 1) *Software* educativo (GCompris, Raptor, DFD) tanto para PC como para dispositivos móviles (Ballesteros, Rodríguez, Lozano y Nisperuza, 2020; Shiguay, 2019)
- 2) Plataformas educativas especializadas en codificación (Franco, García, Guevara y Erazo, 2020; Massachusetts Institute of Technology, 2018; Pérez, Roig y Jaramillo, 2020).
- 3) Cursos en línea especializados (Coursera, Codecademy, edX, Khan Academy, Codewars, Udemy, Dash, Bento.io, Code Avengers).
- 4) Juegos serios (Pilas Bloques, Alice, Greenfoot) (Kuz y Ariste, 2021; Torbado, 2021; Xinogalos y Tryfou, 2021).
- 5) Programas de trazado de código y animaciones (Jeliot, Greenfoot, BlueJ).

El *software* educativo desarrollado para dispositivos móviles es una categoría en continuo crecimiento y difusión, ya que la alta disponibilidad de los teléfonos inteligentes

facilita que docentes y alumnos usen las aplicaciones didácticas para promover las habilidades de los alumnos.

Uso de las *apps* en la enseñanza

Un beneficio del material didáctico digital es la posibilidad de enriquecer la experiencia de aprendizaje mediante el *software* que integra elementos semánticos estructurales y elementos didácticos como los ejercicios y el planteamiento de problemas. También se pueden aprovechar las plataformas tecnológicas que sirven como soporte que actúan como instrumento de mediación para acceder a otros materiales (Criollo, 2017). El uso de dispositivos móviles en la educación constituye una vertiente del *e-learning* denominada *m-learning* (*mobile-learning* o aprendizaje móvil). Se caracteriza por la facilidad de transporte, la inmediatez en la adquisición, el acceso del estudiante sin barreras tipo espacio-tiempo, el compartimiento de información y la personalidad que cada individuo le imprime al uso de su dispositivo.

Las ventajas más destacadas del uso de aplicaciones móviles son:

- 1) Da espacio para mayor cantidad de interacciones, de forma síncrona y asíncrona. Así, a través de una aplicación que permita conversaciones, conferencias en vivo o pregrabadas, se promueven espacios de intercambio profesor-estudiante que se sitúan más allá del aula.
- 2) Potencia el aprendizaje centrado en el alumnado. Esto es posible si las aplicaciones son seleccionadas de manera pertinente para responder a las necesidades que tenga un grupo de estudiantes específicos.
- 3) Permite la personalización del aprendizaje. De acuerdo con sus características, el estudiante puede avanzar en la medida que aumenta su comprensión de los conceptos. Además, es quien controla el uso de la aplicación y por lo tanto su avance.
- 4) Favorece la comunicación entre el alumnado y las instituciones educativas, es una herramienta útil para cerrar la brecha entre los estudiantes y las diferentes personas que puedan requerir comunicarse con ellos.
- 5) Favorece el aprendizaje colaborativo. Las aplicaciones móviles facilitan la creación de grupos en redes sociales, dar apertura a blogs colaborativos, creación de documentos compartidos en herramientas como Google Docs, participación en foros, etc.
- 6) Permite una evaluación inmediata de contenidos educativos. En la medida en que el estudiante va completando las tareas propuestas, el docente tiene la facilidad de corregirlas usando la misma aplicación.
- 7) Se hace un empleo productivo del tiempo en el aula. Al mejorarse la interacción con el docente, se aprovecha el tiempo presencial.

8) Contribuye a mejorar la formación continua. Consiste en utilizar los recursos disponibles en las sociedades para la formación de las personas a lo largo de la vida y para prevenir y remediar carencias a determinadas edades.

9) Se vinculan las tareas formales e informales en el aprendizaje. Este vínculo es el resultado de la intersección de los múltiples universos digitales de usuarios conectados entre sí por nexos, lo que forma una red de dimensiones indefinidas, dinámica, asimétrica y en constante evolución.

10) Apoya al alumnado con discapacidades. Sin lugar a duda la inclusión es una de las tareas que se vuelve prioridad para la educación formal con el objetivo de garantizar la igualdad de derechos que está estipulada en la Declaración Universal de los Derechos Humanos.

Los dispositivos móviles constituyen una oportunidad para el acceso y construcción del conocimiento ya que posibilita el acceso multisensorial y desarrollo de la comunicación interactiva. Esta permite la utilización de soportes gráficos, gracias a lo cual, apoyada de recursos de tipo auditivo o táctil, la convierten en una herramienta para el trabajo con personas con dificultades a la hora de comunicarse (Rodríguez, Rocío, Zambrano y Rodríguez, 2019). Desde hace más de una década se ha explorado el uso de los teléfonos inteligentes en la enseñanza de los alumnos. Es por eso por lo que en esta propuesta se pretende aplicar esta tecnología a favor de facilitar el aprendizaje de los fundamentos de la programación en alumnos de nuevo ingreso en el Centro Universitario UAEM Texcoco.

Métodos

Para alcanzar los objetivos planteados en esta investigación se aplicaron dos fases de trabajo. En la primera se desarrolló la *app* siguiendo las recomendaciones de la ingeniería del *software* y del desarrollo de recursos educativos. En la segunda fase se evaluó la funcionalidad y aceptación del *software*.

Para el desarrollo del *software*, se organizaron las actividades de análisis y diseño en dos dimensiones diferenciadas: la dimensión didáctica y la dimensión tecnológica. Se definieron los flujos de trabajo con base en las recomendaciones de Aguilar y León (2014). En la dimensión didáctica se realizó un análisis de las necesidades académicas con base en el programa de estudio de la materia Programación I del plan de estudios de ingeniería en Computación de la Universidad Autónoma del Estado de México (UAEM). Se delimitaron los temas, se diseñaron las secuencias didácticas y se sometieron a la aprobación de los profesores que imparten la materia.

Respecto a la dimensión tecnológica para el desarrollo de la *app*, se eligió el modelo en cascada con fases solapadas. El modelo de desarrollo es el procedimiento que se sigue durante el proceso de producción de un sistema de *software*, también llamado *paradigma del proceso*. El modelo indica el orden de las etapas involucradas en el desarrollo del *software* y proporciona un criterio para comenzar, para continuar a la siguiente etapa y para finalizar. Las empresas grandes y competitivas trabajan con modelos de desarrollo de *software* y con sistemas de calidad que permiten verificar que el desarrollo se lleve a cabo adecuadamente.

El modelo de cascada con fases solapadas permite un solapamiento de actividades, por ejemplo, facilita que antes de completar el análisis de requerimientos se pueda adelantar el diseño de la aplicación, así los usuarios pueden participar con mayor conocimiento de la propuesta. Otros beneficios de este modelo es que se pueden identificar ideas importantes en el avance del desarrollo y puede funcionar en proyectos grandes y entre etapas se puede reducir la documentación. Las desventajas del modelo son que se requiere la identificación clara de los requerimientos desde el principio y en el solapamiento de fases pueden darse ambigüedades en los procesos (Gómez, Cervantes y González, 2019). En la figura 1 se muestran las etapas de la modelo cascada con fases solapadas

Figura 1. Modelo de cascada con fases solapadas



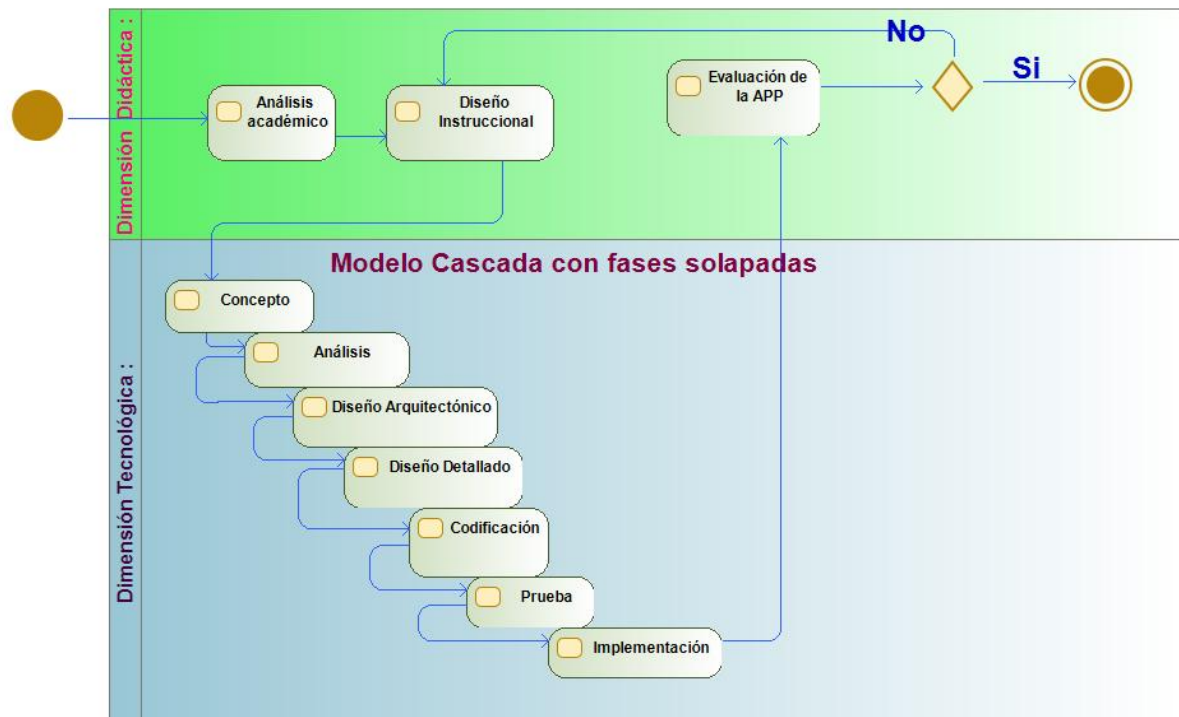
Fuente: Gómez *et al.* (2019, p. 149)

En la figura 2 se describen, a partir de un diagrama de actividades, los pasos que se aplicaron en esta investigación. Pueden observarse las fases referentes a la etapa técnica del desarrollo del *software* separadas por dimensiones: por un lado, el aspecto didáctico, y por el otro, la dimensión tecnológica.

Cabe señalar que en la dimensión tecnológica se usaron las técnicas propias de la ingeniería del *software* para modelar el *software* como son los diagramas de lenguaje

unificado de modelado (UML, por sus siglas en inglés) de casos de uso, diagramas de actividades, diagramas de clases, modelado de datos, diagramas de arquitectura, etc.

Figura 2. Metodología de la investigación



Fuente: Elaboración propia

Las herramientas que se usaron durante el análisis, diseño y desarrollo fueron:

- Editores de texto e imágenes.
- Google Chrome, explorador web.
- Visual Studio Code versión 1.56.2, gratuito.
- Postman: manejo y pruebas de servicios REST, versión 8.7.0, gratuito.
- Git y Github, control de versiones del proyecto en la nube, versión 2.22.0.
- Firebase: gestor de datos gratuito, con ciertos límites de memoria en almacenamiento y peticiones.
- Node.js: entorno de ejecución para JavaScript.
- Android Studio: herramienta para crear aplicaciones móviles tipo Android.
- Angular CLI e Ionic Framework: para crear aplicaciones web y nativas multiplataforma como Angular o React de alta calidad y es de código abierto.

Respecto al diseño instruccional de la *app*, se abordaron los temas básicos que requiere un alumno para adentrarse a la programación orientada a objetos. Con la aprobación de los docentes que imparten la materia en el Centro Universitario UAEM Texcoco y con base en el plan de estudios de Programación I de la carrera de ingeniería

en Computación se definieron los siguientes temas para los cuales se diseñaron guías instruccionales:

- Los componentes de la computadora y su funcionamiento.
- La definición de los algoritmos.
- La metodología para resolver problemas usando la computadora.
- El concepto de *variables, constantes, ámbito de variables*.
- La definición de clases, objetos, constructores, el control de acceso.
- La diferencia entre una clase y el objeto de una clase.
- La invocación y ejecución de un constructor como parte de la creación de un objeto.
- Las operaciones que solo pueden ser invocadas por objetos.
- El estado de un objeto con respecto a los valores de sus campos y como una operación puede cambiar dicho estado.
- El uso de la herencia y los diferentes tipos de clases.

Para la segunda fase de la investigación, se distribuyó la aplicación a un grupo de alumnos y se les dio dos semanas para que la usaran. Una vez pasado ese periodo, se aplicó una encuesta tipo Likert, formada de 20 afirmaciones con cinco posibilidades de respuesta: indistinto, totalmente en desacuerdo, un poco en desacuerdo, un poco en acuerdo, totalmente de acuerdo. La encuesta fue contestada de forma digital mediante un formulario de Google Forms, se dieron siete días para la recepción de respuestas.

Las afirmaciones planteadas fueron:

- 1) Los temas abordados son importantes para aprender a programar.
- 2) La explicación de los temas fue muy clara.
- 3) Los contenidos estaban completos.
- 4) Los contenidos *no* fueron necesarios.
- 5) Faltaron varios temas.
- 6) Faltaron imágenes.
- 7) Faltaron videos.
- 8) Me habría gustado que incluyera juegos.
- 9) Me habría gustado que incluyera código en Java.
- 10) Los colores son agradables.
- 11) El desempeño fue eficiente.
- 12) La *app* fue fácil de usar.
- 13) La *app* fue confusa.

- 14) Había errores en la explicación.
- 15) Las preguntas de evaluación fueron claras.
- 16) Fue fácil obtener puntos.
- 17) Me gustó la *app* para estudiar.
- 18) Me gustaría tener otra *app* con temas más avanzados.
- 19) Creo que si me servirá para aprender a programar.
- 20) Creo que no es útil esta *app*.

Resultados

Como resultado de esta investigación se obtuvo en primer lugar una aplicación móvil con las siguientes propiedades: el proyecto de la *app* tiene un peso de 609 Mb y la APK resultante tiene un peso de tan solo 24 Mb, por lo que resulta rápida de descargar e instalar.

En la APP se optó por un diseño instruccional que abordó conceptos fundamentales e introductorios del área que todo programador debe conocer: la computadora, la programación básica, la programación estructurada y los temas teóricos fundamentales de la programación orientada a objetos.

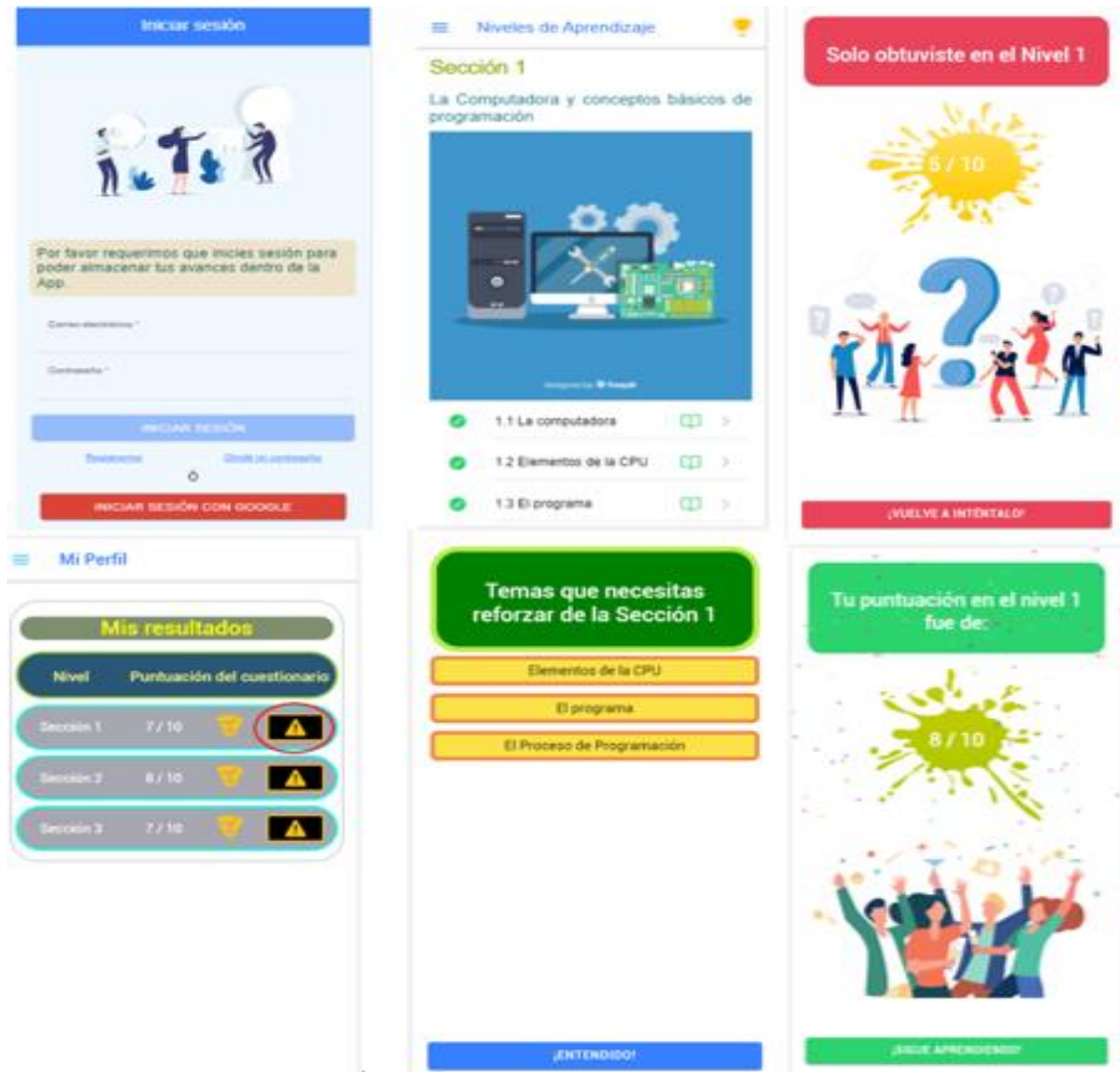
Durante el desarrollo se generaron cuatro versiones de madurez incremental; en cada versión se añadieron nuevas funcionalidades

- En la primera versión se probó la conectividad hacia la base de datos y las consultas de gestión de datos.
- En la segunda versión se implementó y se probó la navegación entre pantallas y el aspecto gráfico de las mismas
- En la tercera versión se probó el alojamiento, la gestión de datos del usuario, los resultados de las pruebas y la ejecución de los cuestionarios de evaluación.
- En la última versión el completado del diseño didáctico dentro de la *app* y la implementación de la funcionalidad de inicio de sesión.

Para la conclusión de la *app* se llevaron a cabo 32 commits Git en total; respecto a las imágenes, se ocuparon 93 imágenes distribuidas en las diferentes pantallas, todas de tipo PNG o SVG, que se obtuvieron de manera gratuita de la plataforma Freepik.

En la figura 3 se muestra una colección breve de algunas pantallas de la aplicación resultante: el inicio de sesión, la visualización de contenidos, los resultados después de la evaluación y por nivel, la ventana de avance y un diagnóstico de temas a reforzar en cada sección.

Figura 3. Pantallas de la app



Fuente: Elaboración propia

A modo de resumen en la tabla 1 se muestra la clasificación y conteo de las líneas de código que se realizó mediante una extensión de Visual Studio Code (VSCodeCounter). Destaca que, sumando solo aquellas que contienen código, se obtuvo la cantidad de 28 057.

Tabla 1. Descriptores del proyecto de desarrollo

Lenguajes	Archivos	Líneas de código	Comentarios como documentación	En blanco	Total
JSON	11	18 333	81	6	18 420
TypeScript	209	5358	279	1,571	7208
HTML	64	3207	11	1014	4232
CSS	65	678	43	184	905
XML	10	419	5	8	432
JavaScript	2	56	8	5	69
Markdown	1	6	0	3	9

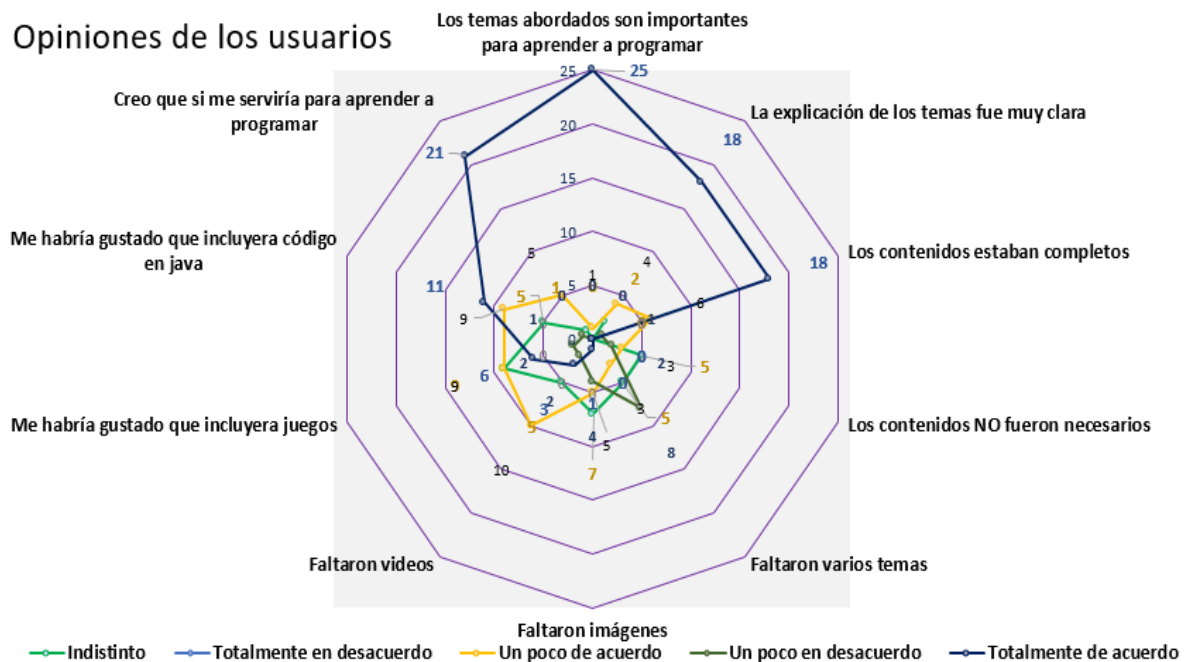
Fuente: Elaboración propia

En la fase de evaluación se obtuvieron 25 encuestas contestadas por los alumnos de ingeniería en Computación, quienes probaron la *app*. De los datos obtenidos, se puede inferir lo siguiente respecto a los contenidos:

- El diseño instruccional fue aceptado por los alumnos. A 95 % la selección de los temas mostrados en la *app* les pareció necesarios e importantes para aprender a programar, al igual que el contenido explicativo; opinaron que los contenidos están completos y tiene una explicación clara.
- A 80 % de las opiniones les gustaría que la *app* incluyera juegos, por lo que se interpreta que los alumnos buscan más interacción con la *app* y que el aspecto lúdico es importante para ellos.
- 90 % de los alumnos opinaron que les habría gustado que la *app* incluyera código en Java, lo que indica que les sería de utilidad los aspectos de sintaxis del lenguaje para aplicar lo aprendido conceptualmente.

En la figura 4 se observa en un diagrama las respuestas de los usuarios respecto a la pertinencia de los contenidos.

Figura 4. Opiniones de los usuarios



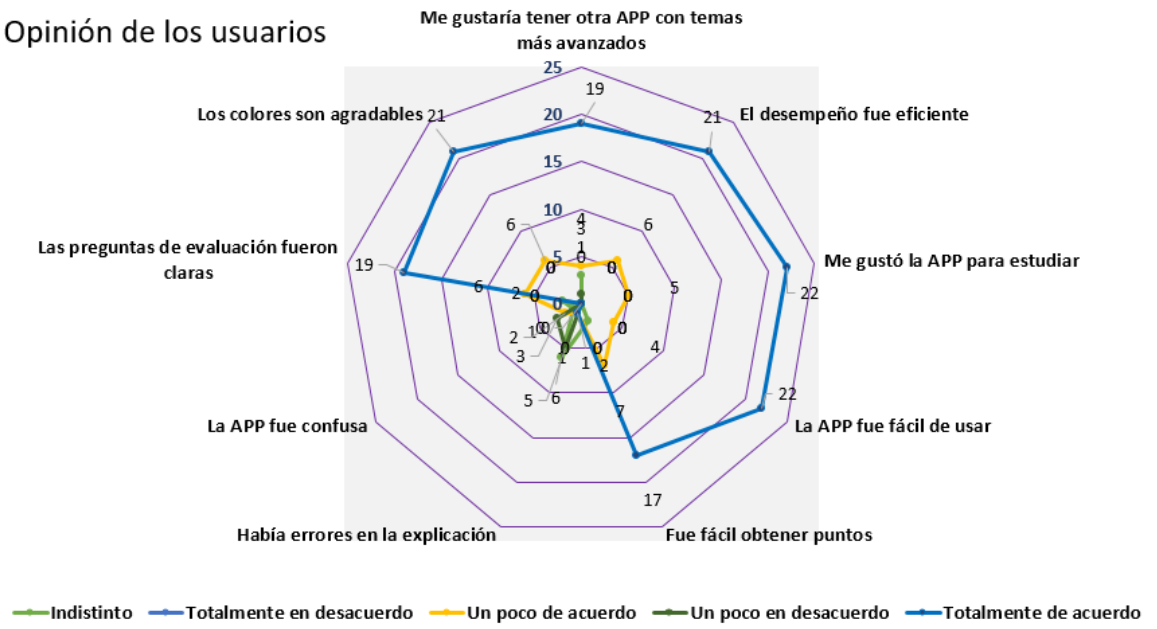
Fuente: Elaboración propia

- En cuanto al diseño, los colores, el tipo y tamaño de la fuente, la interactividad y eficiencia de la APP, completamente a todos les gustó esta *app* para estudiar.
- 90 % de las opiniones de los alumnos indican que les gustaría tener otra *app* con temas más avanzados, lo cual apunta que esta *app* les favoreció para entender las bases de la programación y ahora buscan un reto mayor; también indica que fue de ayuda para que los alumnos se desafiaron a sí mismos a aprender temas con mayor complejidad.
- 95 % de las opiniones de los alumnos expresaron que la *app* les servirá para aprender a programar, lo cual indica que por sí sola P fue exitosamente aceptada por los alumnos y que en definitiva la usan como herramienta de apoyo.

La figura 5 muestra los resultados respecto a los aspectos de usabilidad y funcionalidad.

Figura 5. Opinión de los alumnos respecto a la usabilidad de la APP

Opinión de los usuarios



Fuente: Elaboración propia

Discusión

Los resultados obtenidos referentes a la utilidad y la funcionalidad son satisfactorios. Para los estudiantes, la aplicación sí ha resultado eficiente para ampliar sus conocimientos; los resultados superan a los resultados obtenidos en un estudio similar realizado por Zamora (2019, p. 3), quien señala un 46 % de los alumnos encuestados en su investigación manifiestan que utilizan sus teléfonos para mejorar el aprendizaje por medio de aplicaciones. Cabe mencionar que esta aplicación tiene la ventaja de ser promovida en la universidad como complemento para el aprendizaje del alumno, ya que el uso de las aplicaciones dirigido por los profesores genera mejores resultados, así lo reporta Syaimar y Sutiarsó (2018, citado en Rodríguez, Ballesteros y Lozano, 2020), quienes demostraron que vincular la tecnologías digitales al aula puede ser una solución para superar algunos problemas en el aprendizaje, en su caso en el aprendizaje de las matemáticas.

También Zamora (2019, p. 3) describe en su investigación que 62 % de los alumnos encuestados manifiestan que usan los teléfonos celulares para la realización de tareas escolares. En este caso, los resultados nos permiten conocer la buena aceptación de los alumnos hacia la *app* desarrollada y sus sugerencias ayudarán a ampliar los escenarios

de uso. El diseño instruccional en esta aplicación tomó en consideración las observaciones que hacen respecto a la dimensión didáctica Hinojo, Aznar y Romero (2018), pues ellos concluyen que aún hay trabajo pendiente en la planificación curricular de las aplicaciones *m-learning*. El diseño instruccional, la usabilidad y el fácil acceso de la aplicación han sido los factores señalados por los alumnos como aquellos que facilitan una revisión de los contenidos de la programación de forma más eficiente y cómoda, por lo que el uso de esta resultó satisfactoria y motivante.

El resultado general de esta experiencia en México coincide con el descubrimiento que hacen Hinojo *et al.* (2018) a partir de 22 tesis doctorales cuyo tema de estudio fue el aprendizaje móvil en España. Entre otros hallazgos, su metaanálisis estadístico demostró que los dispositivos móviles mejoran el proceso de enseñanza-aprendizaje; la mayoría de las tesis desarrolladas se efectuaron en la educación superior (15 de 22) en diversas disciplinas. Esta situación permite inferir que se debe trabajar en promover la planificación didáctica adecuada para involucrar a un mayor número de estudiantes y docentes en el uso de este modelo de aprendizaje, a pesar de la renuencia que Zamora (2019) identifica, ya que aún se presenta en algunos docentes que son reacios al cambio y a la innovación.

El aprendizaje móvil enfrenta constantemente retos para su implementación, por ejemplo, los profesores requieren apoyo tecnológico para lograr aplicaciones pertinentes que ellos no saben desarrollar, se requieren metodologías ágiles que vinculen a programadores con docentes en el desarrollo de las *app*.

Las sugerencias de los alumnos de tener materiales para temas de mayor profundidad se justifican con la necesidad que ellos tienen de nuevos materiales educativos, acordes a la tecnología disponible, que promuevan el desarrollo de habilidades y conocimientos durante el proceso de enseñanza-aprendizaje. Los alumnos manifestaron interés por visualizar en la *app* más recursos multimedia. Esta situación genera la necesidad de involucrar a más personas en el desarrollo de las aplicaciones, ya que los contenidos multimedia requieren más tiempo y recursos para su desarrollo.

Conclusiones

Como resultado de esta investigación, se logró crear una *app* ligera que permite instalarse en la mayoría de los dispositivos móviles disponibles para los alumnos del centro universitario para quienes fue desarrollada. Con respecto al diseño didáctico de la *app*, el diseño instruccional fue adecuado, ya que el estudiantado la aceptó plenamente. Aplicar la división de tareas con base en las áreas de especialización de los autores permitió desarrollar un diseño didáctico útil y pertinente para el alumno. En la encuesta

se observa una tendencia a solicitar la inclusión de código Java que permita tener una conexión de los conceptos con la implementación de algunos aspectos de codificación, esto indica que para los alumnos este medio de estudio les puede ayudar en otros temas de importancia.

Respecto al papel del docente, también es importante observar que mediante este tipo de aplicaciones el profesor puede tener una mayor diversidad de materiales educativos para guiar a sus alumnos y promover la autogestión del alumno. Se abren nuevas posibilidades de interacción con el alumno y se requiere mayor creatividad para diseñar actividades de aprendizaje flexibles e innovadoras. Esta aplicación aún puede mejorar, por ejemplo, mediante la gestión automatizada del envío del resumen de evaluaciones al docente o sugerencia de materiales de estudio complementario a los alumnos en caso de lograr bajos puntajes en las evaluaciones.

Futuras líneas de investigación

Este primer ejercicio nos permite tener datos para desarrollar una nueva versión en la que se agreguen más videos y explicaciones referentes al código de un lenguaje de programación orientado a objetos, también vale la pena explorar aspectos de gamificación para lograr un interés mayor en el alumno. Un aspecto más a explorar son actividades propias de la enseñanza del pensamiento computacional que puedan implementarse en este tipo de aplicaciones para facilitar el desarrollo de las habilidades del análisis algorítmico en los alumnos. Para el docente se pueden agregar medios de comunicación con un conjunto de usuarios (grupo) y editar las reglas o incentivos en las reglas de gamificación. El desarrollo requerirá un equipo mayor de programadores y desarrolladores de contenido, por lo que hay que explorar metodologías de desarrollo que vinculen el diseño didáctico con el desarrollo del software.

Referencias

- Aguilar, I. y León, M. (2014). Metodología para el desarrollo de objetos de aprendizaje. Ponencia presentada en el 3.^{er} Congreso Nacional de Tecnologías en la Educación. Puebla, 2014. Recuperado de https://www.academia.edu/11150239/Modelo_de_producci%C3%B3n_de_objetos_de_aprendizaje_usando_SysML.
- Ballesteros, V. A., Rodríguez, O. I., Lozano, S. y Nisperuza, J. L. (2020). El aprendizaje móvil en educación superior: una experiencia desde la formación de ingenieros. *Revista Científica*, 38(2), 243-257. Recuperado de <https://doi.org/10.14483/23448350.15214>.
- Criollo, G. G. (2017). *Materiales didácticos y su incidencia en el proceso de aprendizaje del idioma Inglés para los estudiantes de educación básica media de la Unidad Educativa Réplica Nicolás Infante Díaz*. (Informe de proyecto final de licenciatura). Universidad Técnica de Babahoyo, Quevedo. Recuperado de <http://dspace.utb.edu.ec/handle/49000/3455>.
- Franco, D., García, D. G., Guevara, C. F. y Erazo, J. C. (2020). Scratch para la enseñanza de lenguaje de programación en primero de bachillerato. *Revista Arbitrada Interdisciplinaria Koinonía*, 5(5), 398-414.
- Gómez, M. del C., Cervantes, J. y González, P. (2019). *Fundamentos de Ingeniería de Software*. Naucalpan, México: Litoprocess. Recuperado de <http://ilitia.cua.uam.mx:8080/jspui/bitstream/123456789/1000/1/Fundamentos%20de%20Ingenier%c3%ada%20de%20Software.pdf>.
- Hinojo, F. J., Aznar, I. y Romero, J. M. (2018). Dispositivos móviles para el aprendizaje: análisis de la investigación doctoral sobre mobile learning en España. *Texto Livre*, 11(3), 154-175.
- Kuz, A. y Ariste, M. C. (2021). Un análisis desde la programación estructurada del lenguaje Scratch como entorno lúdico educativo. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, (33), 14-21.
- Massachusetts Institute of Technology. (s. f.). *Scratch*. Recuperado de <https://scratch.mit.edu/>.
- Paredes, I. (2021). *Modelo de programación basado en las mejores prácticas de la programación orientado a objetos, orientado a aspectos y orientado a eventos*

- para la optimización del proceso de desarrollo de software. Caso: Consultoría y Diseño de Sistemas Informáticos S. A. C. (Tesis de licenciatura). Universidad Señor de Sipán, Pimentel.
- Pérez, H., Roig, R. y Jaramillo, L. (2020). Uso de SCRATCH en el aprendizaje de Programación en Educación Superior. *Cátedra*, 3(1), 28-45. Recuperado de <https://doi.org/10.29166/10.29166/catedra.v3i1.2006>.
- Rodríguez, A., Rocío, E., Zambrano, V. y Rodríguez, G. (2019). TIC y aplicaciones móviles en la educación superior; del dicho al reto. *Revista Atlante: Cuadernos de Educación y Desarrollo*. Recuperado de <https://www.eumed.net/rev/atlante/2019/01/tics-educacion-superior.html>.
- Rodríguez, Ó. I., Ballesteros, V. A. y Lozano, S. (2020). Tecnologías digitales para la innovación en educación: una revisión teórica de procesos de aprendizaje mediados por dispositivos móviles. *Pensamiento y Acción*, (28), 83-103.
- Shiguay, G. A. (2019). *Las Tecnologías de la información y comunicación en el aprendizaje de algoritmos y diagramas de flujo en el curso de Informática V en los estudiantes del área de tecnología e informática del quinto grado de educación secundaria de la Institución Educativa Emblemática Ricardo Bentín Sánchez, Rímac, 2018*. (Tesis de maestría). Universidad Nacional de Educación Enrique Guzmán y Valle, Lima.
- Torbado, M. (2021). *Gamificación de Fundamentos de la Programación: juegos serios para el aprendizaje de estructuras de código iterativas, métodos y funciones*. (Tesis de licenciatura). Universidad de Valladolid, Valladolid.
- Xinogalos, S. and Tryfou, M. M. (2021). Using Greenfoot as a Tool for Serious Games Programming Education and Development. *International Journal of Serious Games*, 8(2), 67-86.
- Zamora, R. (2019). El m-learning, las ventajas de la utilización de dispositivos móviles en el proceso autónomo de aprendizaje. *ReHuSo*, 4(3), 29-38. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=7047179>.
- Zatarain, R. (2018). Reconocimiento afectivo y gamificación aplicados al aprendizaje de lógica algorítmica y programación. *Revista Electrónica de Investigación Educativa*, 20(3), 115-125. Recuperado de <https://doi.org/10.24320/redie.2018.20.3.1636>.

Rol de Contribución	Autor (es)
Conceptualización	Irene Aguilar Juárez (igual) Bernardo Alexis Rojas Espinoza (igual)
Metodología	Bernardo Alexis Rojas Espinoza (principal) Irene Aguilar Juárez(apoya)
Software	Bernardo Alexis Rojas Espinoza (principal)
Validación	Bernardo Alexis Rojas Espinoza (principal) Irene Aguilar Juárez(apoya)
Análisis Formal	Irene Aguilar Juárez (principal)
Investigación	Irene Aguilar Juárez (principal) Bernardo Alexis Rojas Espinoza (apoya)
Recursos	Irene Aguilar Juárez (principal)
Curación de datos	Bernardo Alexis Rojas Espinoza (principal) Irene Aguilar Juárez(apoya)
Escritura - Preparación del borrador original	Irene Aguilar Juárez Joel Ayala de la Vega (apoya)
Escritura - Revisión y edición	Joel Ayala de la Vega (apoya) Irene Aguilar Juárez (principal)
Visualización	Irene Aguilar Juárez
Supervisión	Irene Aguilar Juárez
Administración de Proyectos	Irene Aguilar Juárez
Adquisición de fondos	Bernardo Alexis Rojas Espinoza Joel Ayala de la Vega Irene Aguilar Juárez