



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN**

TEMA:

Desarrollo de un prototipo de aplicación móvil gamificada para el aprendizaje activo y el desarrollo de habilidades básicas de programación en el módulo de Python del programa de alfabetización digital, dirigido a participantes del curso ofertado por la UCSG.

AUTORES:

**Acuchi Chafra, Cristian Ronaldo
Ortega Oleas, Darío Fernando**

Proyecto de tecnología de información previo a la obtención del título de INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

TUTOR:

Ing. Cornejo Gómez, Galo Enrique

**Guayaquil, Ecuador
03 de marzo de 2026**



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

CERTIFICACIÓN

Certificamos que el presente proyecto de tecnología de información fue realizado en su totalidad por el Sr. **Acuchi Chafra, Cristian Ronaldo** y el Sr. **Ortega Oleas, Darío Fernando**, como requerimiento para la obtención del título de **Ingeniero en Ciencias de la Computación**.

TUTOR (A)

f. _____

Ing. Cornejo Gómez, Galo Enrique

Guayaquil, 03 de marzo del 2026



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Nosotros, **Acuchi Chafra, Cristian Ronaldo**
Ortega Oleas, Darío Fernando

DECLARAMOS QUE:

El proyecto de tecnología de información **Desarrollo de un prototipo de aplicación móvil gamificada para el aprendizaje activo y el desarrollo de habilidades básicas de programación en el módulo de Python del programa de alfabetización digital, dirigido a participantes del curso ofertado por la UCSG, Requerimiento para la obtención del título de Ingeniero en Ciencias de la Computación** ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Proyecto de T.I. referido.

Guayaquil, 03 de marzo del 2026

f. _____
Acuchi Chafra, Cristian Ronaldo

f. _____
Ortega Oleas, Darío Fernando



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

AUTORIZACIÓN

Nosotros, **Acuchi Chafra, Cristian Ronaldo**
Ortega Oleas, Darío Fernando

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del proyecto de tecnología de información, **Desarrollo de un prototipo de aplicación móvil gamificada para el aprendizaje activo y el desarrollo de habilidades básicas de programación en el módulo de Python del programa de alfabetización digital, dirigido a participantes del curso ofertado por la UCSG**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 03 de marzo del 2026

LOS AUTORES:

f. _____

Acuchi Chafra, Cristian Ronaldo

f. _____

Ortega Oleas, Darío Fernando



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

REPORTE ANTIPLAGIO

	CERTIFICADO DE ANÁLISIS magister
Proyecto ti Final	
0% Textos sospechosos	
Nombre del documento: Proyecto ti Final.docx ID del documento: f151c5b1cccdceb13533f1c5625c536f5b1b132 Tamaño del documento original: 3,87 MB	Depositante: Galo Enrique Cornejo Gómez Fecha de depósito: 19/2/2026 Tipo de carga: interface fecha de fin de análisis: 19/2/2026

Firma:



Ing. Galo Gómez Cornejo, Mgs.
Tutor del Proyecto de T.I.
Carrera de Ingeniería en Ciencias de la Computación

Agradecimiento

Deseo expresar mi más profundo agradecimiento a mi madre, cuyo amor incondicional y esfuerzo incansable han sido el motor que me impulsó a superar cada reto; su fe en mí fue la luz en los momentos de mayor incertidumbre. De manera muy especial, dedico este logro a la memoria de mi padre, quien, aunque ya no está físicamente conmigo, sigue guiando mis pasos con el legado de sus enseñanzas y el orgullo que siempre sintió por mi camino. A mis hermanos, les agradezco por ser mi red de apoyo constante y por compartir conmigo la alegría de este sueño cumplido que también es de ustedes.

Asimismo, extiendo mi gratitud a los amigos que la vida me regaló durante estos cinco años de carrera; gracias por las largas jornadas de estudio, las risas compartidas y por convertirse en una familia elegida que hizo este trayecto inolvidable. Finalmente, agradezco a mis profesores y a la institución por brindarme las herramientas necesarias para mi formación profesional. A todos aquellos que, con un consejo o una palabra de aliento, contribuyeron a que hoy pueda culminar esta etapa con éxito: gracias por ser parte fundamental de mi historia.

Cristian Acuhi Chafila.

Dedicatoria

Dedico este trabajo de titulación con todo mi amor a mi madre, por su apoyo incondicional y por ser el pilar que sostuvo cada uno de mis sueños. A la memoria de mi padre, cuya presencia espiritual y enseñanzas han sido mi guía constante en este largo camino. A mis hermanos, por su compañía y por celebrar conmigo cada pequeño paso hacia esta meta. De manera muy especial, dedico este logro a todas las personas que conocí durante estos cinco años de carrera; a mis amigos y compañeros, gracias por las jornadas de estudio, las risas y por hacer de la universidad una etapa inolvidable de crecimiento compartido.

Asimismo, expreso mi sincero agradecimiento al Ing. Galo Cornejo, por su dedicada guía como tutor y por brindarnos la orientación técnica necesaria para llevar este proyecto a feliz término. Finalmente, a mi alma mater, la Universidad Católica de Santiago de Guayaquil, por abrirme sus puertas y formarme no solo como un profesional competente, sino como una persona íntegra lista para servir a la sociedad.

Cristian Acuhi Chafía.

Agradecimiento

A la Universidad Católica de Santiago de Guayaquil, por haberme brindado una formación de excelencia y los valores necesarios para enfrentar mi futuro profesional.

Mi gratitud se extiende a mis docentes, quienes con su guía y exigencia técnica me inspiraron a alcanzar este nivel de rigor académico. Un agradecimiento especial al Ing. Galo Cornejo que fue nuestro tutor en este gran proyecto. Gracias por abrirme las puertas del conocimiento y por ser el escenario de este gran crecimiento personal.

Dario Fernando Ortega Oleas

Dedicatoria

A mi madre y mi padre, por darme las raíces y las alas para alcanzar esta meta. A mi abuela y mi tía, por cobijarme con su inmensa sabiduría y cariño. A mi novia, por caminar a mi lado con una paciencia y un amor inquebrantables a lo largo de este proceso.

Dario Fernando Ortega Oleas.



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN**

TRIBUNAL DE SUSTENTACIÓN

f. _____

ING. ANA CAMACHO CORONEL, MGS

DIRECTORA DE CARRERA

f. _____

ING. JOSE ERAZO AYON, MGS

DOCENTE DE LA CARRERA

f. _____

ING. ROBERTO GARCIA SANCHEZ, MGS

OPONENTE

Índice General

Resumen	XVII
Abstract.....	XVIII
Introducción.....	2
Capítulo I: El Problema	3
Evaluación del Problema	7
Objetivos.....	9
<i>Objetivo General</i>	9
<i>Objetivos Específicos</i>	9
Alcance del Problema	10
Justificación e importancia.....	11
Preguntas De Investigación	13
Capítulo II: Marco Teórico	14
1. Fundamentos del Aprendizaje Activo en la Educación Superior.....	14
2. Dificultades en el Aprendizaje de la Programación	14
3. Gamificación en la Educación	15
4. Aplicaciones Móviles Educativas	16
5. Gamificación en la Enseñanza de Programación	17
6. Impacto de la Gamificación en el Aprendizaje Activo	18
7. Antecedentes de Proyectos Similares	19
Conclusión del Marco Teórico.....	20
Capítulo III: Metodología de la Investigación.....	21
1. Enfoque de la Investigación.....	21
2. Tipo de Investigación	21
2.1. Clasificación y enfoque de la investigación	21
3. Diseño de la Investigación	21
4. Población y Muestra	22
4.1. Población	22
4.2. Muestra	22
5. Técnicas e Instrumentos de Recolección de Datos.....	22
6. Procedimiento de la Investigación	23
6.1. Fase de análisis.....	23

6.2.	Fase de diseño	23
6.3.	Fase de implementación	24
6.4.	Fase de evaluación.....	24
6.5.	Fase de análisis de resultados.....	24
7.	Métodos de Análisis de Datos.....	24
7.1.	Consideraciones Éticas.....	24
8.	Análisis de los Resultados	25
8.1.	Encuesta Inicial.....	25
8.2.	Encuesta de los Estudiantes de Fundamentos de la Programación	25
	Metodología de Desarrollo.....	28
1.	Justificación de la metodología del desarrollo.....	28
2.	Desarrollo evolutivo y escalabilidad del prototipo	29
3.	Procedimiento secuencial para el aprendizaje	29
3.1	Análisis de requerimientos y necesidades	29
3.2	Enfoque de diseño de interfaz (UI).....	30
3.3	Implementación del prototipo funcional	30
3.4	Validación del entorno operativo	30
3.5	Optimización y refinamiento del sistema.....	31
4.	Definición del stock tecnológico.....	31
	Capítulo IV: Propuesta Tecnológica.....	33
1.	Introducción	33
2.	Objetivo	33
2.1.	Objetivo General	33
2.2.	Objetivos Específicos.....	33
3.	Responsables	34
4.	Desarrollo	34
4.1.	Desarrollo de la aplicación.....	34
4.2.	Descripción de la Lógica de Gamificación	41
4.3.	Diseño de Interfaz y Experiencia de Usuario (UI/UX).....	44
4.3.1.	Interfaz	46
4.4.	Persistencia y Seguridad de los Datos	51
4.5.	Requerimientos Técnicos y Entorno de Ejecución	53

Conclusiones.....	55
Recomendaciones	57
Bibliografía.....	59
Anexos	62
Anexo 1	62
Anexo 2	63

Índice de Figuras

Figura 1. <i>Diagrama de flujo de la app</i>	41
Figura 2. <i>Ciclo de retroalimentación del juego.</i>	42
Figura 3. <i>Visualización del sistema de recompensa estética en gamificación educativa</i>	43
Figura 4. <i>Arquitectura visual y componentes de interacción.</i>	45
Figura 5. <i>Imagen del inicio de la app con archivos.json atractivos para el usuario.</i>	46
Figura 6. <i>Imagen que muestra un tip acerca de python.</i>	46
Figura 7. <i>Mapa de módulos.</i>	47
Figura 8. <i>Perfil del usuario</i>	47
Figura 9. <i>Tienda</i>	48
Figura 10. <i>Información antes de iniciar un test.</i>	48
Figura 11. <i>Resultado de los puntos</i>	49
Figura 12. <i>Test de los módulos.</i>	49

Índice de Tablas

Tabla 1. Tabla de criterios y contextos.....	7
Tabla 2. Síntesis de resultados estadísticos de las encuestas aplicada a los estudiantes.....	26
Tabla 3. Ventajas, desventajas y características de las tecnologías utilizadas en la app.....	35
Tabla 4. Comparativa de componentes de infraestructura base y su rol en el sistema.....	37
Tabla 5. Características de las funciones principales de la aplicación educativa.....	39
Tabla 6. Mecánicas de gamificación y su equivalente en el proceso de aprendizaje.....	43
Tabla 7. Principios de usabilidad aplicados en la interfaz del sistema.....	45
Tabla 8. Estrategias de persistencia y seguridad de información.....	52
Tabla 9. Especificaciones técnicas mínimas y recomendadas para la instalación.....	54

Índice de Anexos

Anexo 1	62
Anexo 2	63

Resumen

El objetivo de este proyecto es desarrollar un prototipo de aplicación gamificada para el programa de alfabetización digital de la UCSG. Queremos que el aprendizaje de Python sea un proceso dinámico basado en los principios del conectivismo, en lugar de una tarea tediosa. Para que esto funcione, se están integrando mecánicas de juego bajo los estándares de UX, siguiendo una metodología de software incremental que prioriza, por encima de todo, la arquitectura funcional. El contenido de este trabajo detalla la implementación de un mapa de niveles interactivo, lecciones con retroalimentación inmediata y un sistema de progresión que utiliza la persistencia de datos locales para realizar un seguimiento de la experiencia (XP). También describe la integración de elementos de control, como el desbloqueo de módulos a través de estados lógicos. Además, incluye estilos visuales dinámicos para indicar el éxito del usuario y refuerzo háptico con vibraciones para señalar errores de navegación. Profundiza en la gestión técnica de las actividades de aprendizaje. Esto garantiza que el soporte del sistema responda de manera eficiente a las interacciones de los estudiantes. En conclusión, se determina que el prototipo gamificado reduce la brecha digital en los programas de alfabetización. Al suavizar la curva de aprendizaje de Python, este método garantiza una comprensión más orgánica del lenguaje. Además, proporciona un marco robusto y ampliable que admite la eventual incorporación de cursos de programación de alto nivel a la oferta académica de la institución.

Palabras Clave: Software, Arquitectura funcional, Escalabilidad, UX (Experiencia de Usuario), Conectivismo.

Abstract

The goal of this project is to develop a prototype gamified app for the UCSG digital literacy program. We want learning Python to be a dynamic process based on connectivism principles, rather than a chore. To make this work, we are integrating game mechanics under UX standards, following an incremental software methodology that prioritizes, above all else, functional architecture.

The content of this work details the implementation of an interactive level map, lessons with immediate feedback, and a progression system that uses local data persistence to track experience (XP). It also describes the integration of control elements such as unlocking modules through logical states. It also includes dynamic visual styles to indicate user success and haptic reinforcement with vibrations to signal navigation errors. It delves into the technical management of learning activities. This ensures that the system support responds efficiently to student interactions. In conclusion, it is determined that the gamified prototype reduces the digital divide in literacy programs. By smoothing out the learning curve for Python, this method ensures a more organic grasp of the language. Furthermore, it provides a robust, expandable framework that supports the eventual addition of high-level programming courses to the institution's academic lineup.

Key words: Software, Functional Architecture, Scalability, UX (User Experience), Connectivism.

Introducción

Enseñar programación a quienes recién empiezan es uno de los mayores retos de la alfabetización digital actual. No basta con entregar información; para aprender Python y entender el pensamiento computacional, se necesitan métodos que realmente motiven y logren que el estudiante se sienta parte del proceso. Aquí es donde entra la gamificación, que aprovecha elementos de los juegos, como los niveles, las recompensas y el feedback constante, para que aprender sea algo entretenido, dinámico y mucho más fácil de mantener en el tiempo.

Bajo esta idea, surge este proyecto para crear una aplicación móvil gamificada diseñada para el módulo de Python de la Universidad Católica de Santiago de Guayaquil (UCSG). La meta es que personas de cualquier edad o nivel de experiencia puedan entender la lógica de la programación de una forma activa y cercana.

La aplicación se estructura de manera secuencial, empleando entornos lúdicos y desafíos prácticos que facilitan la adquisición de competencias en Python, reduciendo la carga cognitiva del usuario. Mediante un sistema de puntuación y logros, se pretende incentivar la curiosidad técnica y la autonomía en el aprendizaje. En última instancia, la propuesta busca proveer una herramienta tecnológica optimizada y accesible que fortalezca la autoeficacia y el desarrollo de habilidades digitales en el contexto tecnológico actual.

Capítulo I: El Problema

Aprender a programar en Python dentro del programa de Alfabetización Digital de la UCSG está siendo un verdadero reto para muchos participantes. El problema no es la capacidad de los estudiantes, sino que los métodos de enseñanza tradicionales a menudo se sienten distantes y poco motivadores, lo que termina generando desinterés. Al no contar con herramientas que inviten a participar de forma activa, el desarrollo de habilidades digitales se vuelve un proceso lento y pesado. Por eso, se ve una necesidad clara: crear una aplicación móvil que use el juego para que aprender a programar sea una experiencia emocionante, clara y, sobre todo, muy efectiva.

Análisis del Problema

Ubicación del problema en un contexto hoy por hoy, el celular es la herramienta principal para aprender casi cualquier cosa. En este mundo tecnológico, Python se ha convertido en el lenguaje estrella porque es sencillo de entender y fundamental en áreas como la inteligencia artificial. Sin embargo, a pesar de su importancia, todavía cuesta encontrar aplicaciones que aprovechen de verdad el potencial de los celulares para que las personas practiquen solas de forma divertida. Aquí es donde aparece una gran oportunidad: diseñar un prototipo que use retos, recompensas y respuestas inmediatas para que el usuario sienta que realmente está progresando y disfrute el proceso.

Este proyecto se construye con la idea de ser abierto y accesible para todos, utilizando software libre para que la aplicación pueda crecer y mejorar con el tiempo. Se está enfocando muchísimo en que sea muy fácil de usar y que funcione perfectamente en dispositivos Android, pensando siempre en la comodidad de quienes la van a usar. Todo este esfuerzo se está realizando en la Universidad Católica de Santiago de Guayaquil para el periodo 2025–2026, buscando crear una solución que realmente marque la diferencia en la comunidad.

Para guiar este trabajo, se priorizo dos puntos clave:

Variable independiente: El prototipo de la aplicación móvil gamificada que se está creando.

Variable dependiente: Qué tan bien funciona y qué tan útil resulta para que los estudiantes realmente aprendan a programar en Python.

En pocas palabras, el vacío tecnológico que se detecto es que no existe una herramienta propia que use el juego y la movilidad para enseñar programación de una manera humana y práctica. Al proponer esta solución, no solo se busca escribir código, sino innovar en la educación digital con una herramienta que entienda las necesidades de las personas y las motive a seguir creciendo en el mundo de la tecnología.

Causas y Consecuencias del Problema

Causas

1. Faltan herramientas hechas a medida: Actualmente no hay una aplicación pensada específicamente para quienes participan en el programa de Alfabetización Digital de la UCSG. Casi todo lo que encontramos en internet es muy general o está en otros idiomas, lo que hace que sea difícil aplicarlo a la realidad.
2. Poco uso del juego en las apps educativas locales: Aunque se sabe que aprender jugando es mucho más fácil, todavía no se aprovecha esta técnica en los proyectos universitarios de la zona. Muchas veces esto pasa porque no se sabe bien cómo usar las herramientas técnicas necesarias para que el juego funcione dentro de la aplicación.
3. Desaprovechamos el software libre: se tiene a la mano herramientas gratuitas y poderosas como Flutter, React Native o Kotlin, pero todavía no se han usado para crear un proyecto propio de la universidad que enseñe Python de una manera entretenida.
4. No existen soluciones personalizadas que funcionen en todo lugar: No se cuenta con un desarrollo propio que junte en un solo lugar el aprendizaje, el control de los avances y los consejos inmediatos, y que además funcione bien en cualquier tipo de celular o dispositivo.

Consecuencias o proyecciones

Si se deja pasar este problema y no creamos una solución propia, nos enfrentaríamos a estos retos en el futuro:

1. La persistencia en la dependencia de proveedores externos obliga a la institución a utilizar plataformas de terceros, lo cual restringe la autonomía para adaptar los entornos de aprendizaje a las necesidades específicas de los estudiantes. Esta limitación técnica condiciona la escalabilidad de los programas académicos y compromete la soberanía sobre la gestión de contenidos curriculares y el control operativo de los procesos de enseñanza-aprendizaje.
2. Menos espacio para la innovación: Si no se desarrolla una propia tecnología, la universidad pierde la oportunidad de destacar. Esto afecta a la carrera de Computación, ya que se vería limitada a usar software ajeno en lugar de demostrar que puede liderar la creación de herramientas educativas modernas
3. Se perderían funciones que motivan: Sin una app diseñada por sus alumnos, no podríamos aplicar funciones geniales como el seguimiento del avance paso a paso o los premios dinámicos. Estas son precisamente las herramientas que hacen que el usuario sienta que progresa y quiera seguir aprendiendo.
4. Dejar pasar una tendencia clave: La gamificación es el futuro de la educación digital. Si no se explora ahora desde la ingeniería, se estaría desperdiciando un área de investigación muy valiosa que permitiría crear sistemas de aprendizaje mucho más interactivos y cercanos a la gente.

Delimitación del Proyecto

Campo

Este proyecto nace en el corazón de la Ingeniería en Ciencias de la Computación con una misión muy clara: usar la tecnología para transformar la educación. No se trata solo de desarrollar software, sino de crear herramientas móviles que realmente conecten con las personas y les abran puertas hacia el conocimiento digital de una manera interactiva, moderna y cercana.

Área

Se enfoca en el desarrollo de aplicaciones móviles utilizando software libre, pero con un enfoque profundamente humano. La prioridad es que la tecnología deje de ser una barrera y se convierta en una aliada. Por eso, se aprovechó la gamificación y el diseño pensado en las personas para que aprender sea algo natural, fluido y, sobre todo, una experiencia agradable y motivadora dentro del mundo digital.

Aspecto

Se busca construir algo que funcione de verdad: un prototipo que acompañe al estudiante en cada paso de su progreso. Al incluir niveles, retos y recompensas constantes, se quiere que el usuario se sienta motivado y nunca se sienta solo en su aprendizaje. Además, al usar herramientas de código abierto, se asegura de que esta herramienta pueda evolucionar, adaptarse a diferentes tipos de celulares y ser fácil de mantener para que siempre esté al día.

Tema:

Desarrollo de un prototipo de aplicación móvil gamificada, basada en software libre, para apoyar el aprendizaje activo y la práctica de habilidades básicas de programación en el módulo de Python del programa de Alfabetización Digital de la Universidad Católica de Santiago de Guayaquil (UCSG).

Formulación del Problema

Hoy en día, existe la tecnología y las herramientas gratuitas al alcance de la mano para crear soluciones que realmente nos faciliten la vida. Sin embargo, en la UCSG todavía nos falta una aplicación propia que use el juego como motivación para que aprender Python sea algo práctico y emocionante dentro del curso de Alfabetización Digital.

Esto nos hace pensar: ¿cómo se puede aprovechar el software libre y las dinámicas de juego para crear una app que sea útil, rápida y que respete el ritmo de cada estudiante? El verdadero reto es que, ahora mismo, quienes están empezando a programar no tienen una plataforma sencilla y diseñada especialmente para ellos.

Por lo tanto, durante el periodo 2025-2026, se llevará a cabo en la universidad el diseño, lanzamiento y fase de pruebas del prototipo. Los actores principales del estudio son los estudiantes de Python, quienes validarán la aplicación y proporcionarán retroalimentación directa sobre su experiencia de usuario y el proceso de aprendizaje.

En última instancia, el propósito del estudio consiste en evaluar el desempeño del prototipo, fundamentando el análisis en criterios de usabilidad, eficiencia de rendimiento y robustez técnica. El objetivo primordial es consolidar esta herramienta como un recurso pedagógico optimizado que facilite el aprendizaje y la práctica del lenguaje Python en la UCSG, promoviendo un entorno educativo interactivo y eficaz para los estudiantes.

Evaluación del Problema

De los diez criterios definidos para la evaluación del problema, se escogerán siete, los cuales se indican en la tabla siguiente:

Tabla 1. *Tabla de criterios y contextos*

#	Criterios	Contexto
1	Delimitado	El problema se encuentra claramente definido en términos de espacio (Universidad Católica de Santiago de Guayaquil), tiempo (periodo 2025–2026) y población (participantes del módulo de Python del programa de Alfabetización Digital).
2	Claro	El problema está formulado de manera precisa y comprensible, ya que se enfoca directamente en la ausencia de un prototipo propio de aplicación móvil gamificada basada en software libre para

#	Criterios	Contexto
		<p>el aprendizaje de Python. La redacción permite identificar fácilmente qué se va a desarrollar, para quién y con qué propósito tecnológico.</p>
3	Relevante	<p>El problema es relevante para la comunidad universitaria, ya que impulsa la generación de soluciones tecnológicas propias dentro de la UCSG. Contribuye también al fortalecimiento del desarrollo de apps, el software libre y la utilización de trendy's como la gamificación en la acción institucional de la tecnología.</p>
4	Original	<p>Se hace la propuesta del desarrollo de una aplicación móvil gamificada de software libre para el aprendizaje de Python, no desarrollada antes en la institución.</p>
5	Concreto	<p>El planteamiento del problema está bien pensado y preciso y trata en específico el desarrollo de software. No se concierne a cuestiones pedagógicas ni sociales, sino que se ocupa de sólo la producción de una solución tecnológica operativa, lo que lo hace elegible para una tesis de Ingeniería en Ciencias de la Computación.</p>

#	Criterios	Contexto
6	Factible	El problema puede ser abordado dentro del tiempo y los recursos disponibles, utilizando herramientas de software libre, frameworks de desarrollo móvil y los conocimientos adquiridos en la carrera.
7	Variables	El problema permite identificar con claridad la variable independiente, que es el prototipo de aplicación móvil gamificada, y la variable dependiente, que es su funcionalidad y desempeño tecnológico para el aprendizaje de Python.

Nota: Elaboración Propia

Objetivos

Objetivo General

Crear un prototipo de aplicación móvil que use el juego como estrategia de aprendizaje, apoyándose en herramientas de software libre. El propósito es que los estudiantes del módulo de Python en el programa de Alfabetización Digital de la Universidad Católica de Santiago de Guayaquil (2025-2026) cuenten con un recurso práctico que les facilite dar sus primeros pasos en la programación de manera entretenida.

Objetivos Específicos

- Identificar qué funciones y características técnicas debe tener la aplicación para que realmente ayude a los estudiantes a aprender las bases de Python.
- Analizar y elegir las mejores herramientas y marcos de trabajo de código abierto para construir un prototipo sólido y moderno.

- Implementar las funciones principales de la aplicación usando software libre, integrando retos y actividades interactivas diseñadas para practicar código.
- Diseñar la estructura interna y el diseño de las pantallas pensando siempre en que sea fácil, cómodo y agradable de usar para el alumno.
- Evaluar el funcionamiento del prototipo en un entorno controlado para asegurar que sea rápido, estable y sencillo de navegar antes de su uso real.

Alcance del Problema

Este proyecto consiste en diseñar y construir un prototipo de aplicación móvil que use el juego para ayudar a los estudiantes de Alfabetización Digital de la UCSG a practicar Python durante el periodo 2025-2026. El objetivo es crear un software útil que aproveche las ventajas del mismo y que sea fácil de usar.

El vínculo entre el consumidor y la marca puede verse fortalecido a través del uso de un diseño óptimo que sirva como herramienta de ventas. Incluiré toda la carpeta técnica donde se encuentran los requerimientos, el modelo de la estructura del sistema y los resultados obtenidos de las pruebas.

Es importante aclarar el alcance que se da únicamente al prototipo. Esto significa que no se busca crear una versión comercial, subirla a tiendas de aplicaciones ni conectarla con los sistemas internos o bases de datos de la universidad. Tampoco se incluye el mantenimiento a largo plazo ni estudios sobre cómo influye en el aprendizaje después de terminar el proyecto.

Para el desarrollo del proyecto, el sustento teórico se fundamenta en los contenidos programáticos del curso de Python y en las especificaciones técnicas de los dispositivos móviles vigentes. El proceso de validación técnica y operativa cuenta con el respaldo del programa de Alfabetización Digital y la Facultad de Ingeniería, asegurando el cumplimiento de los estándares de calidad y funcionalidad del software.

En síntesis, el proyecto se centra exclusivamente en el ciclo de vida del desarrollo de software. El objetivo consiste en la entrega de un prototipo funcional y debidamente documentado, delimitando el alcance fuera de procesos administrativos o despliegues comerciales masivos.

Justificación e importancia

El presente trabajo surge de la necesidad de aplicar los conocimientos adquiridos durante la formación en Ingeniería en Ciencias de la Computación en un entorno de ejecución real. El proyecto integra el desarrollo móvil, las filosofías de software libre y técnicas de gamificación, con el propósito de generar una herramienta operativa que optimice el proceso de aprendizaje. A través de la construcción de este prototipo, se evidencia la capacidad de emplear tecnologías actuales para el desarrollo de soluciones digitales innovadoras que modernizan los métodos educativos. Finalmente, se pretende que este estudio se consolide como una referencia técnica y fuente de consulta para futuros desarrollos dentro de la institución.

Conveniencia

El desarrollo del proyecto representa la integración de competencias profesionales, abarcando desde la fase de planificación conceptual hasta el despliegue de la aplicación funcional. Esta propuesta constituye un aporte significativo para la UCSG, al proveer una herramienta institucional desarrollada bajo estándares de código abierto. La adopción de este modelo garantiza la soberanía tecnológica de la universidad, permitiendo el mantenimiento evolutivo y la escalabilidad del sistema de manera autónoma, sin restricciones de licenciamiento o dependencia de terceros.

Relevancia social

Desde una perspectiva de impacto social, el proyecto promueve la democratización del acceso a la tecnología. Mediante el uso de software libre, se fomenta la creación de herramientas accesibles que contribuyan a reducir la brecha digital dentro de la comunidad universitaria. El objetivo trasciende la formación técnica, buscando el desarrollo de profesionales con capacidad para diseñar soluciones tecnológicas alineadas con las demandas y necesidades actuales de la sociedad.

Implicaciones prácticas

En términos prácticos, el resultado consiste en una aplicación operativa que permite la interacción y la ejecución de pruebas de campo. El proyecto trasciende la fundamentación

teórica, constituyéndose como un caso de estudio sobre la convergencia entre el diseño centrado en el usuario (UCD) y una arquitectura técnica robusta. Asimismo, permite analizar de manera empírica la implementación de mecánicas de gamificación aplicadas al proceso de enseñanza-aprendizaje, demostrando la viabilidad del software de código abierto para sustentar soluciones tecnológicas de alta calidad.

Valor teórico

En términos de contribución académica, el presente trabajo constituye un recurso de consulta para futuras investigaciones en el área del desarrollo móvil. Se presenta una documentación detallada sobre el ciclo de vida del software, integrando principios de ingeniería de software y diseño de experiencia de usuario (UX). De este modo, el proyecto se consolida como una base técnica y metodológica de referencia para estudiantes y docentes, facilitando el desarrollo de soluciones tecnológicas similares o la expansión de la línea de investigación propuesta.

Utilidad

La herramienta será de gran ayuda para realizar prácticas, medir el rendimiento del software y entender mejor cómo los usuarios interactúan con la tecnología. Al estar basada en software libre, su código es como una semilla que puede crecer: otros pueden tomarlo, mejorarlo o adaptarlo para enseñar otros lenguajes de programación o para nuevos proyectos de la institución.

Beneficiarios

Los beneficiarios directos son los estudiantes de la carrera de Computación, quienes dispondrán de un caso de estudio real sobre el ciclo de vida del desarrollo de software profesional. Asimismo, la UCSG fortalece su línea de innovación tecnológica mediante la implementación de soluciones propias. Finalmente, el proyecto constituye un aporte para futuros desarrolladores y docentes, estableciendo una base técnica que fomenta el aprendizaje continuo y la creación de nuevas herramientas en diversos entornos.

Preguntas De Investigación

¿En qué medida el uso de mecánicas de juego en una aplicación móvil ayuda a que los estudiantes de la UCSG entiendan Python de forma más sencilla que con las clases convencionales?

¿Cómo ayuda una interfaz basada en retos y juegos a que los participantes del programa de alfabetización digital de la UCSG recuerden mejor lo aprendido sobre programación en Python?

Capítulo II: Marco Teórico

1. Fundamentos del Aprendizaje Activo en la Educación Superior

El aprendizaje activo se entiende como un proceso transformador en el que el estudiante deja de ser un receptor pasivo de información para asumir un rol protagónico en su propia formación académica. En vez de escuchar explicaciones, el alumno pasa a participar en forma dinámica, reflexionando de forma crítica, resolviendo problemas y trabajando con pares. Esta estrategia del aprendizaje busca generar un conocimiento significativo y robustecer con ello la autonomía intelectual (Parra Panduro & Cherre Antón, 2025).

A diferencia de la educación que tematiza un contenido específico para que los estudiantes lo asimilen, el aprendizaje activo se enfoca en que ese conocimiento adquirido tenga todo su sentido en la vida real. Según Bello, M., et al. (2022), esta metodología promueve el pensamiento crítico y la toma de decisiones fundamentadas y el desarrollo de competencias profesionales en la educación superior.

Incluso, la incorporación de instrumentos digitales, debates estructurados, estudios de caso y simulaciones facilita una experiencia formativa más profunda y duradera (Adriana Castillo Rosas, 2022). La práctica y la contextualización son importantes en el aprendizaje de cualquier materia. Esto es algo que se discute con frecuencia. El uso de plataformas interactivas permite experimentar con código real, analizar errores y recibir retroalimentación inmediata, lo que facilita la comprensión de conceptos abstractos. De acuerdo con Cambridge Assessment International Education (2021), cuando la teoría se integra con la práctica en entornos digitales interactivos, el aprendizaje se vuelve más significativo y alineado con las demandas del entorno profesional actual.

2. Dificultades en el Aprendizaje de la Programación

Se reconoce que el proceso de aprender a programar constituye uno de los desafíos más significativos en la formación de futuros ingenieros. La programación no solo exige la

comprensión de conceptos técnicos, sino también el desarrollo de un pensamiento lógico estructurado y la capacidad de analizar problemas de manera abstracta.

La necesidad de dominar simultáneamente el razonamiento algorítmico y la sintaxis específica de un lenguaje como Python suele generar confusión y sentimientos de frustración en las etapas iniciales del aprendizaje.

Según distintos estudios, los problemas no sólo se deben a la complejidad de los contenidos, sino también a metodologías de enseñanza demasiado teóricas, donde el alumno puede practicar poco y no tiene feedback inmediato (Beltrán Morales, 2021). La percepción del error como fracaso y no como oportunidad de aprendizaje genera un incremento de la ansiedad académica con una desaprobación de la continuidad.

La gamificación se transforma en una alternativa que promete revolucionar esta experiencia. Mediante dinámicas del juego, como el reto paulatino, la recompensa y la retroalimentación continua, descomponen lo complejo y lo hacen menos difícil. De acuerdo al autor (2023), este enfoque permite conseguir que el estudiante esté más comprometido y adquiera una actitud más positiva respecto a la programación.

De esta manera, el aprendizaje deja de percibirse como una actividad rígida y se convierte en un proceso interactivo, donde la práctica constante, el ensayo y error, y la motivación intrínseca permiten construir habilidades sólidas desde las primeras etapas de formación.

3. Gamificación en la Educación

Gamificar no significa jugar por jugar; implica aplicar de manera estratégica mecánicas propias de los juegos, como puntos, retos, recompensas y sistemas de progreso, en contextos formales con el fin de motivar y comprometer a las personas. En el ámbito educativo, la gamificación busca transformar el aprendizaje en una experiencia dinámica e interactiva, donde el estudiante asume un rol activo y se siente motivado a superar metas, avanzar por niveles y alcanzar objetivos concretos dentro de su proceso formativo.

Múltiples estudios han mostrado que elementos como insignias digitales, niveles de dificultad progresivos, clasificaciones y retroalimentación inmediata promueven la competencia sana y aumentan la autorregulación del estudiante.

Según Prieto-Andreu (2022) se potencia la motivación intrínseca al percibir satisfacción por los objetivos alcanzados, complementándose con incentivos externos de alta visibilidad. Bajo este enfoque, el proceso de aprendizaje trasciende la obligatoriedad académica convencional, transformándose en un desafío que fomenta la participación y el compromiso del estudiante.

El aspecto de juego tiene mucha importancia en el fortalecimiento del aprendizaje, ya que genera entusiasmo, compromiso y sensación de logro. Recientes estudios aseguran que gamificar la asignatura ayuda a proliferar en la mejora de rendimiento y motivación, particularmente en las unidades de enseñanza que son complejas (Lu, 2025). Al convertir actividades tradicionalmente monótonas en retos atractivos, se favorece la participación constante y la construcción progresiva del conocimiento, fortaleciendo así la permanencia y el interés del estudiante en su proceso de aprendizaje.

4. Aplicaciones Móviles Educativas

Dado que el uso del teléfono móvil forma parte de la vida cotidiana, el impacto de las aplicaciones educativas en los procesos de enseñanza y aprendizaje ha sido considerable. El aprendizaje móvil o m-learning ofrece una flexibilidad única, permitiendo que cada estudiante avance a su propio ritmo, sin depender estrictamente de horarios o espacios físicos determinados (INTEF, 2022).

Esta característica resulta especialmente relevante en la educación superior, donde los estudiantes combinan actividades académicas con responsabilidades personales y laborales.

Las aplicaciones móviles educativas agrupan en un solo espacio digital diferentes recursos, como vídeos explicativos, ejercicios, cuestionarios de autoevaluación y feedback. Esta combinación permite tomar contacto con los contenidos de forma que su comprensión se haga

de modo progresivo, favoreciendo, a su vez, un ejercicio constante que consolide los conocimientos (Loayza Maturrano, 2022).

La posibilidad de que el estudiante reciba respuestas automáticas ante aciertos o errores le permite a identificar, casi en el mismo momento, en qué se equivoca y corregir su aprendizaje.

Los recursos de matemáticas online más comunes son las calculadoras, que ayudan a los estudiantes a resolver problemas complejos de manera más rápida. Otras plataformas permiten la visualización gráfica y la realización de prácticas interactivas. En el contexto universitario, estas herramientas se convierten en un complemento estratégico a las clases presenciales, ya que brindan un espacio adicional para reforzar conceptos, practicar habilidades y profundizar en temas específicos sin la presión de la evaluación formal. Con la incorporación de la gamificación, las aplicaciones incrementan el compromiso del estudiante al añadirle puntos, niveles, recompensas, etc. Así se consigue que el estudiante se sienta más motivado. De este modo, el uso de aplicaciones móviles no solo amplía el acceso al conocimiento, sino que también promueve la autonomía, la constancia y el aprendizaje activo dentro y fuera del aula (Jaramillo Mediavilla et al, 2025).

5. Gamificación en la Enseñanza de Programación

Se ha comprobado que utilizar estrategias basadas en juegos para enseñar programación produce resultados altamente favorables en distintos niveles educativos. La programación, al requerir precisión lógica y dominio de estructuras sintácticas, puede resultar intimidante para quienes se inician en este campo. En este contexto, la gamificación se convierte en una herramienta pedagógica que reduce la ansiedad inicial y transforma el aprendizaje en una experiencia más accesible, estructurada y estimulante para los estudiantes principiantes (Cruz-García et al, 2021).

Las plataformas CodeCombat y Scratch prueban que usar dinámicas lúdicas favorece la participación activa y potencia la motivación intrínseca del estudiante. Mediante misiones, niveles progresivos, retos interactivos y recompensas virtuales, estas herramientas vuelven la

escritura de código un reto dinámico donde cada logro es un avance concreto y medible (García Rodríguez, 2022).

En el ámbito académico, la gamificación ayuda a que tareas que pueden parecer repetitivas se conviertan en una experiencia de superación constante. La acumulación de puntos, el desbloqueo de contenidos, el feedback instantáneo y los sistemas de progreso visible generan una sensación de avance continuo que nutre la confianza en uno mismo. La curiosidad por encontrar nuevas soluciones, optimizar algoritmos o experimentar con diferentes estructuras lógicas también se estimula (Londoño Vásquez & Rojas López, 2020).

El empleo de estas estrategias para el curso de Alfabetización Digital representa una oportunidad interesante para el fortalecimiento de la lógica de programación desde los primeros años de vida, logrando así un aprendizaje activo, participativo, progresivo y sostenido en el tiempo (Zhan et al., 2022).

6. Impacto de la Gamificación en el Aprendizaje Activo

¿Por qué resulta tan efectiva la gamificación en contextos educativos? Su éxito puede explicarse a partir de la Teoría de la Autodeterminación, la cual plantea que las personas experimentan una motivación más profunda y sostenible cuando se satisfacen tres necesidades psicológicas básicas: competencia, autonomía y relación.

En este contexto, la motivación del usuario se incrementa cuando se percibe la capacidad de superar desafíos técnicos, se dispone de autonomía en la toma de decisiones y se establecen vínculos de interacción social dentro de la plataforma. Las dinámicas de gamificación implementadas proporcionan objetivos claros y alcanzables, permitiendo la progresión a ritmos individualizados y generando espacios de interacción que fortalecen el compromiso del estudiante con el proceso de aprendizaje (Martínez J, 2021).

En ámbitos virtuales, el uso de esta teoría ha mostrado ser efectivo para aumentar el nivel de participación. Según Félix da Silva (2022), el uso de mecánicas de juego bien estructuradas en la educación permite que los estudiantes se comprometan más, sean más persistentes ante los obstáculos y estén más dispuestos a aprender. Un desarrollo continuo que puede llevar a la sensación de competencia y a un esfuerzo constante.

Los elementos lúdicos ayudan al alumnado a mantener la motivación y a minimizar el estrés. Cuando un error se convierte en una experiencia de aprendizaje que ocurre en un entorno suficientemente seguro, la ansiedad se reduce y el temor al fracaso se minimiza. Esta dinámica, según Villamar Gavilanes (2024), facilita la conformación de confianza y resiliencia académica.

De este modo, la gamificación no solo incrementa la motivación, sino que fortalece el aprendizaje activo al promover exploración, práctica constante y autorregulación del estudiante.

7. Antecedentes de Proyectos Similares

Se ha observado que diversos proyectos educativos han integrado con éxito la gamificación como estrategia para la enseñanza de programación en distintos contextos académicos. Estas iniciativas han demostrado que la incorporación de dinámicas de juego favorece la comprensión de conceptos técnicos y fortalece la motivación del estudiante (Morales Zúñiga & Ruiz Ledesma, 2025).

Mediante la implementación de plataformas digitales que incorporan desafíos periódicos, sistemas de puntuación, niveles de progresión e insignias virtuales, se ha conseguido que conceptos abstractos como la lógica algorítmica, las estructuras de control y la sintaxis fundamental resulten más accesibles. Este enfoque metodológico optimiza la curva de aprendizaje, facilitando la comprensión de fundamentos técnicos complejos para los usuarios noveles en el área de la programación (Beltrán Morales et al, 2021).

Un caso significativo fue el uso de una herramienta gamificada para los cursos introductorios. En él, se pudo exhibir que además de mejorar el rendimiento académico, se sumó el interés y la interacción del estudiante con la materia. (Pérez, 2020) Este tipo de resultados nos sugiere que la gamificación impacta no sólo en sus calificaciones, sino también en la actitud frente al aprendizaje.

Con el fin de complementar un aprendizaje matricial en el ámbito universitario, se puede utilizar una estrategia didáctica con el uso de rompecabezas constructores. A tal efecto, investigaciones recientes evidencian que aumenta en los estudiantes la activa participación en clase, así como el interés para resolver situaciones problemáticas complejas (Peñañiel

Villavicencio, 2025). Los estudiantes dejan de ver los ejercicios como tareas repetitivas y los afrontan como desafíos progresivos que desean superar.

Estos antecedentes evidencian que las dinámicas de juego constituyen una alternativa pedagógica viable y efectiva. Asimismo, respaldan la factibilidad de desarrollar un prototipo basado en software libre que promueva una experiencia formativa más interactiva, motivadora y alineada con las necesidades actuales de la educación tecnológica.

Conclusión del Marco Teórico

El análisis de los fundamentos teóricos revisados permite afirmar que el diseño de una aplicación móvil con dinámicas de juego constituye una estrategia pertinente y fundamentada para fortalecer el aprendizaje de Python en el curso de Alfabetización Digital. La combinación del aprendizaje activo, la gamificación y la aplicación de herramientas TIC educativas se propone como un enfoque metodológico que se ajusta actualmente a la educación superior, especialmente en el caso de las carreras técnicas y de programación (Ballesteros Ballesteros et al, 2020).

El aprendizaje activo se produce cuando el estudiante está permanentemente involucrado y favorece que haya una construcción significativa del conocimiento. Por su parte, la gamificación incluye componentes que pueden ser motivacionales y que pueden aumentar el nivel de compromiso, persistencia y confianza frente a los retos académicos (Pardo Montero et al, 2024). Por su parte, las aplicaciones móviles amplían el acceso al aprendizaje al ofrecer flexibilidad espacial y temporal, permitiendo que los estudiantes practiquen de manera autónoma y progresiva. La convergencia de estos tres componentes crea un entorno formativo dinámico, interactivo y adaptado a las características de los estudiantes contemporáneos (López-Noguero et al, 2023).

Asimismo, los antecedentes revisados demuestran que la implementación de mecánicas de juego en la enseñanza de programación genera mejoras tanto en el rendimiento académico como en la percepción positiva de la asignatura. Este estudio empírico confirma la importancia de la realización de un prototipo en un software libre que tenga niveles, retos y devoluciones instantáneas. Se puede desprender de lo mencionado anteriormente que la propuesta no sólo

responde a una necesidad tecnológica institucional, sino que también se apoya en bases pedagógicas para fortalecer competencias digitales prácticas, motivadoras y actuales en el aula universitaria.

Capítulo III: Metodología de la Investigación

1. Enfoque de la Investigación

Para este estudio se seleccionó un enfoque mixto que permite analizar ambas dimensiones del fenómeno. Por una parte, se emplean datos precisos para medir el impacto de la aplicación en la mejora de las habilidades de programación. Por otra, el interés se centra en recopilar las opiniones y vivencias reales de los estudiantes durante su uso. Se optó por este camino debido a que el objetivo no consiste únicamente en entregar una herramienta tecnológica, sino en comprobar su utilidad y aceptación dentro de un entorno educativo real.

2. Tipo de Investigación

2.1. Clasificación y enfoque de la investigación

- **Investigación aplicada:** Se busca resolver un problema real y práctico en las clases de Python dentro del curso de Alfabetización Digital, creando una herramienta tecnológica que sea de ayuda directa.
- **Investigación descriptiva:** El interés se centra en explicar el funcionamiento de la aplicación y los efectos que esta genera en el proceso de aprendizaje de los alumnos.
- **Investigación de desarrollo tecnológico:** El centro del trabajo es diseñar y construir un prototipo funcional de aplicación móvil usando herramientas de software libre.

3. Diseño de la Investigación

- El diseño es no experimental, ya que la intención no es manipular variables a propósito, sino observar cómo interactúan los estudiantes con el prototipo en un entorno real y controlado.

- Asimismo, la investigación es de corte transversal, debido a que la recolección de datos y el levantamiento de información se ejecutan en un periodo de tiempo único y determinado. Este proceso se concentra específicamente durante la fase de pruebas piloto del prototipo funcional, permitiendo evaluar el estado de las variables y el rendimiento del sistema en un momento puntual, sin requerir un análisis longitudinal de carácter plurianual

4. Población y Muestra

4.1. Población

El corazón de este estudio son los estudiantes que cursan Fundamentos de la Programación en la UCSG. La selección de la muestra se fundamenta en el perfil del usuario objetivo: estudiantes de niveles iniciales en tecnología que enfrentan la curva de aprendizaje de la lógica en Python. Las necesidades identificadas en este grupo proporcionan el sustento funcional al proyecto. Asimismo, se integró la perspectiva de los docentes, cuya experiencia pedagógica resulta determinante para asegurar que los contenidos de la aplicación cumplan con los objetivos de aprendizaje propuestos

4.2. Muestra

Con el fin de evaluar el prototipo y recopilar información relevante, se optó por trabajar con una muestra no probabilística por conveniencia. Esta decisión se fundamentó en la necesidad de mantener un contacto directo con estudiantes que cursaran la materia de forma activa; de este modo, la retroalimentación obtenida sobre la dificultad de las dinámicas y la claridad de los conceptos de alfabetización digital resulta más auténtica y funcional. En total, el estudio contó con la participación de 25 estudiantes, quienes aportaron datos a través de encuestas y pruebas de usabilidad para identificar sus necesidades y el impacto de la aplicación en el aprendizaje. Asimismo, se mantuvo un diálogo constante con el docente de la asignatura para asegurar que los contenidos de la herramienta estuvieran alineados con el programa oficial del módulo de Python.

5. Técnicas e Instrumentos de Recolección de Datos

Para la recolección de información se emplearán las siguientes técnicas e instrumentos:

- **Encuestas:** Aplicadas a los estudiantes participantes antes y después del uso del prototipo, con el fin de evaluar el nivel de motivación, percepción del aprendizaje activo y utilidad de la aplicación.

Instrumento: Cuestionario estructurado con escala tipo Likert.

- **Observación directa:** Permitirá analizar el comportamiento de los estudiantes durante el uso de la aplicación, su interacción con los retos y el cumplimiento de las actividades semanales.

Instrumento: Guía de observación.

- **Pruebas prácticas:** Se utilizarán ejercicios de programación para medir el desarrollo de habilidades básicas antes y después de la implementación del prototipo.

Instrumento: Lista de cotejo o rúbrica de evaluación.

6. Procedimiento de la Investigación

Con el propósito de asegurar el éxito del proyecto, la ejecución se llevará a cabo de manera sistemática, siguiendo las etapas que se detallan a continuación:

6.1. Fase de análisis

- Primero, se trató de entender qué es lo que realmente necesitan los estudiantes para aprender Python en sus clases de Alfabetización Digital.
- También se investigó qué dicen los expertos sobre cómo usar el juego y la tecnología para que aprender sea más fácil y entretenido.

6.2. Fase de diseño y arquitectura

- Se establecerán los requerimientos estéticos y operativos de la aplicación, determinando la arquitectura de información y el flujo de navegación del sistema.
- Se desarrollará el sistema de incentivos, que incluye la estructura de recompensas, desafíos temporales y niveles de progresión, con el objetivo de fomentar el compromiso y la motivación del usuario.
- Se elaborarán los *wireframes* y diagramas de flujo funcionales para la visualización

previa de la interfaz y la lógica de negocio antes de iniciar la etapa de desarrollo.

6.3. Fase de implementación

- Se construirá el prototipo usando herramientas de software libre Integración de contenidos académicos (ejercicios, videos, puzles).
- Se cargará todo el contenido educativo, desde ejercicios y videos hasta puzles interactivos.

6.4. Fase de evaluación

- Se pondrá a prueba la aplicación con un grupo de estudiantes para ver cómo les va.
- Se les pedirá su opinión mediante encuestas y se verá qué tanto aprendieron con algunas prácticas.

6.5. Fase de análisis de resultados

- Se analizará con calma los datos y los comentarios de los alumnos para entender el impacto real del proyecto.
- Se realizará un análisis de los aciertos técnicos, las áreas de mejora y la incorporación de nuevos requerimientos para optimizar el rendimiento y la escalabilidad del prototipo de forma incremental.

7. Métodos de Análisis de Datos

Los datos cuantitativos recolectados mediante encuestas y pruebas prácticas serán analizados utilizando estadística descriptiva, empleando tablas y gráficos para facilitar su interpretación.

Los datos cualitativos obtenidos a partir de la observación y comentarios de los estudiantes serán analizados mediante análisis de contenido, identificando patrones, percepciones y opiniones relevantes relacionadas con el uso de la aplicación gamificada.

7.1. Consideraciones Éticas

La investigación respetará los lineamientos éticos esenciales, asegurando que los

alumnos participen por decisión propia y que sus datos personales se manejen de forma estrictamente confidencial, recopilada; y, el uso de los datos únicamente con fines académicos. Los participantes serán informados sobre los objetivos del estudio y podrán retirarse del mismo en cualquier momento sin consecuencias académicas

8. Análisis de los Resultados

8.1. Encuesta Inicial

La encuesta inicial permite identificar las deficiencias en la metodología tradicional de enseñanza de Python dentro del programa de alfabetización digital. Los hallazgos revelaron que los métodos convencionales suelen carecer de elementos motivadores, lo cual genera una barrera de frustración significativa para los estudiantes principiantes. De manera análoga a los sistemas de información que carecen de mecanismos de registro, la lógica y la sintaxis de programación no se consolidan debido a la ausencia de una práctica interactiva inmediata. Los participantes manifestaron que el estudio de la teoría, sin el acompañamiento de desafíos o ejercicios aplicados, dificulta la comprensión de estructuras de control como bucles y condicionales. Estos resultados confirmaron la necesidad de implementar una herramienta tecnológica que transforme el aprendizaje en un proceso activo y aplicado.

8.2. Encuesta de los Estudiantes de Fundamentos de la Programación

Tras la implementación y evaluación del prototipo de la aplicación, se realizaron entrevistas y seguimientos con los estudiantes de la UCSG para analizar la percepción del usuario y el fortalecimiento de sus competencias técnicas. En relación con la interactividad y los elementos de gamificación, los participantes destacaron que el sistema de puntuación y niveles transformó la percepción de la programación, convirtiéndola en un desafío competitivo y dinámico. Respecto a la sintaxis de Python, la mayoría de los usuarios logró identificar errores de código con mayor celeridad debido a la retroalimentación inmediata del sistema, la cual actúa como una guía constante durante la escritura de scripts. Finalmente, se confirmó la alta usabilidad de la herramienta; incluso los estudiantes con menor autoconfianza tecnológica navegaron por las lecciones de manera autónoma, reduciendo significativamente la

dependencia de asistencia externa.

- **Resultados de la encuesta previa**

La dificultad percibida para aprender a programar se situó en un promedio de 3,88/5; un 42 % calificó la tarea como muy difícil y un 24 % como difícil. Más de la mitad de los alumnos, el 52 %, reconoce que escuchar un término técnico en inglés les genera confusión, el 36 % siente curiosidad y el 12 % miedo. El 64% de los encuestados se siente más cómodo aprendiendo en un libro que con dibujos o juegos. Asimismo, porcentaje similar asegura que prefiere aprender en su teléfono y no en una computadora de escritorio. El 76% de los encuestados opina que resolver rompecabezas resulta más divertido que codificar en una hoja. La puntuación de las mecánicas ronda un 4,06/5, mientras que la estética de los colores y sonidos se ubica en 3,76/5. Respecto al primer logro que les gustaría alcanzar, el 70 % quiere que la computadora salude con su nombre, el 21 % prefiere realizar operaciones matemáticas y el 9 % elaborar una lista de compras. Al cometer errores, el 82 % prefiere recibir una pista y el 18 % simplemente quiere saber que se equivocó. Y el 85 % cree que mejorar sus competencias digitales repercute positivamente en su vida académica y laboral.

- **Resultados de la encuesta posterior:**

El 12 % de los encuestados no logró completar la instalación, mientras que el 88 % sí pudo. Mientras utilizaban la app, el 94 % nunca tuvo cierre imprevisto y el 6 % solo una vez. Las puntuaciones medias (4,0/5) y desviación típica (0,69) de los entrevistados evidencian su alta facilidad de uso. El 88 % de los usuarios validan el tamaño de textos e iconos, y el 88 % tienen una opinión favorable sobre la paleta de colores. El 100 % reconoció de inmediato la función principal de la aplicación al abrirla.

Tabla 2. *Síntesis de resultados estadísticos de las encuestas aplicada a los estudiantes.*

Encuesta	Pregunta	Resultado principal
----------	----------	---------------------

Encuesta previa	Dificultad para aprender a programar	42 % nivel 5, 24 % nivel 4, 15 % nivel 3; promedio 3,88/5
Encuesta previa	Reacción a términos en inglés	52 % se confunde, 36 % siente curiosidad, 12 % siente miedo
Encuesta previa	Preferencia de aprendizaje	64 % prefiere libros, 36 % dibujos/juegos
Encuesta previa	Dispositivo preferido	64 % smartphone, 36 % computadora
Encuesta previa	Preferencia por rompecabezas	76 % prefiere puzles, 24 % escribir en hoja
Encuesta previa	Valoración de recompensas	Promedio 4,06/5
Encuesta previa	Valoración de estética	Promedio 3,76/5
Encuesta previa	Primer objetivo	70 % saludar con su nombre, 21 % operaciones matemáticas, 9 % lista de compras
Encuesta previa	Respuesta ante errores	82 % prefiere pistas, 18 % conocer error
Encuesta previa	Importancia de las competencias digitales	85 % considera que ayudan, 15 % no
Encuesta posterior	Instalación de la app	88 % sin problemas, 12 % con errores
Encuesta posterior	Estabilidad de la app	94 % sin cierres, 6 % una vez
Encuesta posterior	Facilidad de uso de botones	Promedio 4,0/5

Encuesta posterior	Adecuación del tamaño de textos	88 % adecuado, 6 % pequeño, 6 % grande
Encuesta posterior	Colores de la interfaz	88 % agradables, 12 % no
Encuesta posterior	Función principal identificada	100 % de inmediato

Nota: Elaboración propia

Metodología de Desarrollo

1. Justificación de la metodología del desarrollo

La naturaleza dinámica de una aplicación móvil que combina un motor de juego con un sistema educativo requiere un enfoque flexible. Por ello, se ha seleccionado el modelo de Prototipado. Esta metodología permite que el software crezca de forma orgánica, donde cada iteración no solo corrige errores de la anterior, sino que añade capas de complejidad técnica y pedagógica.

- **Justificación Técnica de la Selección**

En proyectos de Tecnologías de la Información orientados al aprendizaje (m-Learning), el éxito depende de la simbiosis entre el rendimiento del software y la retención del usuario. La elección del prototipado responde a tres necesidades críticas identificadas durante la planificación:

- ✚ **Validación del Motor Gráfico:** Antes de implementar los 30 módulos, fue imperativo desarrollar un prototipo funcional del SnakeEngine para validar la estabilidad de los 60 FPS (cuadros por segundo) en dispositivos de gama media.
- ✚ **Ajuste de la Curva de Dificultad:** El proceso de prototipado se utilizó también en pruebas de campo con estudiantes de segundo ciclo, con el objetivo de determinar si el tiempo de respuesta que requería la serpiente interfería con el tiempo de procesamiento cognitivo de las preguntas de programación.

- ✚ **Integración de la Persistencia:** Se desarrollaron prototipos específicos para testear la robustez de la clase `SharedPreferences`, asegurando que la escritura asíncrona de datos no generara fugas de memoria o corrupción de los puntos XP (Developers, 2024).

- **Fases de Ejecución de la Metodología**

Para este proyecto, el prototipado se dividió en cuatro fases fundamentales que estructuraron el flujo de trabajo:

- ✚ **Fase 1:** Recolección de Requisitos y Análisis: Se definieron las competencias de programación necesarias para los estudiantes de segundo ciclo. Aquí se diseñó la lógica del Banco de Preguntas y los requerimientos funcionales mínimos de la interfaz.
- ✚ **Fase 2:** Diseño Rápido y Construcción del Prototipo Base: Se implementó la arquitectura básica en Kotlin, enfocándose en la mecánica de colisiones y el renderizado en `SurfaceView`.
- ✚ **Fase 3:** El prototipo fue sometido a pruebas de usabilidad (ux). El feedback que se obtuvo permite identificar que algunos colores de la serpiente dificultan la visibilidad del texto en las preguntas. Por eso, se rediseñaron bajo el criterio de Material Design 3.
- ✚ **Fase 4:** Ingeniería de Producto Final: Tras validar la mecánica de juego y la lógica de aprendizaje, se procedió a la implementación de la Tienda de Skins y el sistema de bloqueo secuencial de módulos, garantizando una arquitectura de software limpia y escalable.

2. Desarrollo evolutivo y escalabilidad del prototipo

No se va a intentar hacerlo todo de golpe. Se usará un prototipo evolutivo, lo que significa que se empezará con las funciones más básicas y se irán mejorando paso a paso. En cada vuelta, se añaden nuevas características basadas en lo que se aprenden de las pruebas anteriores. Así, el resultado final no será algo rígido, sino el fruto de un proceso de maduración constante que permite aprovechar mejor el tiempo y el esfuerzo.

3. Procedimiento secuencial para el aprendizaje

3.1 Análisis de requerimientos y necesidades

Esta fase inicial se fundamenta en la identificación y el levantamiento de requerimientos, proceso orientado a determinar las necesidades académicas de los estudiantes en la asignatura de Fundamentos de Programación y sus expectativas respecto a la herramienta móvil. En esta etapa, se definen los requerimientos funcionales críticos del sistema, tales como la implementación de mecánicas de gamificación y la organización de contenidos curriculares por periodos semanales. Asimismo, se establecen los atributos de calidad de software (requerimientos no funcionales), priorizando la eficiencia en el rendimiento y la usabilidad de la interfaz, con el fin de garantizar una experiencia de usuario intuitiva que facilite el proceso de aprendizaje.

3.2 Enfoque de diseño de interfaz (UI)

Con las necesidades claras, se creará los primeros bocetos. Son prototipos visuales sencillos que permiten ver cómo se navegará entre pantallas y dónde irá cada elemento. Es la forma de "ver" la aplicación antes de escribir una sola línea de código complejo.

3.3 Implementación del prototipo funcional

En esta etapa se ejecuta la construcción de una versión operativa del sistema, integrando los módulos fundamentales para el cumplimiento de los objetivos técnicos. El desarrollo comprende la implementación del sistema de autenticación de usuarios, la estructuración de contenidos académicos distribuidos de forma semanal y la integración de mecánicas de gamificación mediante un sistema de puntuación. Para garantizar la sostenibilidad y accesibilidad del proyecto, se emplean herramientas de código abierto y servicios de nivel gratuito (*free tier*), asegurando una arquitectura técnica robusta sin incurrir en costos de licenciamiento.

3.4 Validación del entorno operativo

Una vez alcanzada la estabilidad funcional del prototipo, se procede a la ejecución de una prueba piloto con una muestra representativa de estudiantes. El objetivo de esta fase es evaluar la usabilidad del sistema, la comprensión de las mecánicas de gamificación integradas y el nivel de compromiso (*engagement*) generado hacia el proceso de aprendizaje. Los datos

recolectados y las observaciones de los usuarios finales constituyen una fuente crítica de retroalimentación, permitiendo diagnosticar la situación operativa actual y realizar los ajustes técnicos necesarios antes del despliegue definitivo.

3.5 Optimización y refinamiento del sistema

A partir de los hallazgos obtenidos en las pruebas de usabilidad, se procederá a la fase de optimización de la aplicación. Este proceso incluye el ajuste del diseño de la interfaz, la mejora del rendimiento de las funciones lógicas y el refinamiento de los elementos de gamificación. El objetivo principal es garantizar una experiencia de usuario fluida y eficiente, asegurando que el flujo de aprendizaje sea intuitivo para todos los participantes.

4. Definición del stack tecnológico

Para garantizar la robustez, el rendimiento gráfico y la escalabilidad del proyecto, se seleccionó un conjunto de herramientas de desarrollo nativo.

A diferencia de soluciones genéricas, esta selección permite un acceso directo a los recursos del hardware del dispositivo, optimizando la ejecución del motor de juego.

- **Entorno de Desarrollo Integrado (IDE)**

Se utilizó Android Studio (versión Jellyfish o superior) como la estación de trabajo principal. Este IDE proporciona las herramientas de depuración, el editor de diseño XML para la interfaz y el sistema de construcción Gradle, fundamentales para gestionar las dependencias del proyecto y la compilación del binario APK.

- **Lenguaje de Programación**

El código fuente de la aplicación fue escrito en Kotlin. El lenguaje fue elegido por su seguridad de nullos (Null Safety), por su sintaxis con un número mínimo de líneas y por su total compatibilidad con las APIs modernas de Android, para implementar hilos secundarios del motor de juego.

- **Diseño de Interfaz y Gráficos**

En vez de utilizar herramientas de prototipado externas, se hizo un uso directo de Android XML Layouts y Material Design 3. Esto permite una integración inmediata entre los componentes visuales y la lógica de programación, asegurando que la interfaz sea adaptativa (responsive) a diferentes tamaños de pantalla sin perder calidad.

- **Motor de Juego y Renderizado**

Para el desarrollo de la mecánica de la serpiente, se empleó la API nativa de Canvas y SurfaceView. Esta decisión técnica permite un renderizado de alto rendimiento a 60 FPS, gestionando los gráficos de manera eficiente sin la sobrecarga que implican los frameworks de terceros.

5. Impacto proyectado de la solución tecnológica

Al terminar, no solo se tendrá una aplicación, sino una prueba real de que aprender programación puede ser algo activo y divertido. Este prototipo será el corazón de un sistema educativo más moderno y servirá de base para que, en el futuro, se pueda implementar a una escala mucho mayor en toda la universidad.

Capítulo IV: Propuesta Tecnológica

1. Introducción

El propósito de esta iniciativa es crear un prototipo de aplicación móvil que utilice el juego para fortalecer el aprendizaje activo de los estudiantes de segundo ciclo de Computación en la UCSG. Se sabe que aprender a programar puede ser un reto difícil, y esta propuesta nace precisamente para atacar problemas como la falta de motivación o esa sensación de no recibir ayuda inmediata cuando uno se queda trabado en un código. Al unir la movilidad del celular con dinámicas divertidas de gamificación, se quiere entregar una herramienta que no solo enseñe, sino que invite al estudiante a participar y practicar por su cuenta. Además, para que el proyecto sea una realidad accesible para todos, se construirá usando únicamente software libre.

2. Objetivo

2.1. Objetivo General

Crear un prototipo de aplicación móvil que use el juego como estrategia de aprendizaje, apoyándose en herramientas de software libre. El propósito es que los estudiantes del módulo de Python en el programa de Alfabetización Digital de la Universidad Católica de Santiago de Guayaquil (2025-2026) cuenten con un recurso práctico que les facilite dar sus primeros pasos en la programación de manera entretenida.

2.2. Objetivos Específicos

- Identificar qué funciones y características técnicas debe tener la aplicación para que realmente ayude a los estudiantes a aprender las bases de Python.
- Analizar y elegir las mejores herramientas y marcos de trabajo de código abierto para construir un prototipo sólido y moderno.
- Implementar las funciones principales de la aplicación usando software libre, integrando retos y actividades interactivas diseñadas para practicar código.
- Diseñar la estructura interna y el diseño de las pantallas pensando siempre en que sea fácil, cómodo y agradable de usar para el alumno.

- Evaluar el funcionamiento del prototipo en un entorno controlado para asegurar que sea rápido, estable y sencillo de navegar antes de su uso real.

3. Responsables

Este proyecto está en manos de Cristian Ronaldo Acuchi Chafra y Dario Fernando Ortega Oleas, quienes se encargan de todo el proceso, desde el análisis y diseño hasta el desarrollo y la documentación. Se contó con el apoyo de un docente tutor, quien guio paso a paso para que el contenido sea académicamente sólido y se cumplan todos los objetivos planteados.

4. Desarrollo

El desarrollo de esta propuesta se llevará a cabo mediante una metodología de prototipado evolutivo, avanzando de forma organizada por las siguientes etapas:

4.1. Desarrollo de la aplicación

En el panorama actual, el desarrollo de soluciones móviles se ha visto impulsado por marcos de trabajo que optimizan el rendimiento y la compatibilidad. Como señala (Falade, 2024) la introducción de entornos como Flutter o React Native ha permitido a los desarrolladores implementar aplicaciones de alto rendimiento utilizando una única base de código.

Para este proyecto, se decidió usar una arquitectura nativa con Kotlin porque se necesitó que el hardware respondiera al instante. Esta elección es fundamental ya que, a diferencia de las aplicaciones de gestión comunes, el motor gráfico del minijuego requiere una fluidez y estabilidad que solo el desarrollo nativo puede garantizar.

En cuanto al manejo de la información, se buscó que fuera lo más eficiente posible. Aunque hay sistemas que utilizan herramientas como FastAPI para mover datos en tiempo real (Falade, 2024), se añadió un backend local usando SharedPreferences y persistencia de datos. Esto permite que la tienda y el progreso del usuario funcionen perfectamente sin depender de una conexión a internet, logrando que la aplicación se pueda usar para aprender en cualquier lugar.

Uno de los elementos innovadores del sistema es la integración de una economía digital gamificada. Bajo esta premisa, el progreso académico no se limita a la lectura de contenido,

sino que se traduce en unidades de valor (puntos XP). Al igual que los sistemas de reconocimiento de voz facilitan la interacción con dispositivos electrónicos al eliminar la fricción de las interfaces táctiles (Falade, 2024), la aplicación reduce la resistencia al estudio mediante el refuerzo positivo. Los usuarios no son solo seres pasivos que consumen información, sino que se esfuerzan académicamente en personalizar su espacio. Esto, a su vez, afianza su identidad.

El proyecto se robustece en una estructura de módulos que se interconectan. Con el sistema de control de flujo implementado, el aprendizaje es secuencial al bloquear desafíos avanzados hasta validar competencias base. Esta tecnología jerárquica y el motor gráfico cuya personalización se ejecuta por medio de parámetros hexadecimales, hace de la aplicación un producto tecnológico. No solo hace más dinámico el proceso de enseñanza, sino que insufla modernidad a la elaboración de proyectos formativos con el uso del Machine Learning y la programación aplicada.

Tabla 3. *Ventajas, Desventajas y características de las Tecnologías utilizadas en la app.*

Tecnología	Características Principales	Ventajas	Desventajas
Kotlin (Nativo)	Lenguaje de tipado estático, moderno e interoperable con Java.	Alto rendimiento y gestión eficiente de memoria RAM.	Curva de aprendizaje más compleja que lenguajes de scripting.
SurfaceView	Clase especializada para renderizado en hilos secundarios.	Proporciona 60 FPS fluidos sin bloquear la interfaz principal.	Mayor complejidad en la gestión del ciclo de vida (SurfaceHolder).

Tecnología	Características Principales	Ventajas	Desventajas
SharedPreferences	Sistema de almacenamiento clave-valor para persistencia ligera.	Acceso instantáneo a configuraciones y progreso local.	No apto para grandes volúmenes de datos o relaciones complejas.

Nota: Elaboración Propia

4.1.1. Funcionalidades de la App

La aplicación móvil desarrollada propone una ruptura con los métodos de enseñanza técnica tradicionales al integrar un entorno de aprendizaje basado en juegos (GBL). Esta transición hacia un modelo de gamificación estructural permite transformar la instrucción pasiva en un proceso de interacción dinámica, donde el usuario asume un rol activo en la resolución de problemas lógicos. Mediante la ejecución de algoritmos en un entorno lúdico controlado, el sistema facilita la asimilación de conceptos complejos de programación, optimizando la curva de aprendizaje y fortaleciendo el compromiso del estudiante con el contenido curricular

A diferencia de los sistemas de gestión de aprendizaje convencionales (LMS), caracterizados por una arquitectura estática y lineal, la solución implementada despliega un entorno de gamificación inmersiva que optimiza la adquisición de competencias en programación. Mediante la integración de un motor gráfico y lógica procedimental, el sistema suprime el modelo de recepción pasiva, posicionando al usuario en un rol de ejecución activa. Esta configuración técnica facilita la validación de conocimientos en tiempo real mediante el procesamiento de desafíos cognitivos y entornos de simulación interactivos, asegurando una retroalimentación inmediata sobre el flujo lógico del código.

Uno de los componentes más destacados del sistema es la implementación de una economía virtual y un módulo de personalización. Este mecanismo permite el reconocimiento

de los logros académicos de los estudiantes, transformando los aciertos en las evaluaciones en puntos de experiencia (XP). A diferencia de los métodos de evaluación tradicionales, cuya calificación es estática, este sistema permite la redención inmediata de los puntos obtenidos en una tienda virtual para la personalización de avatares con diversos estilos gráficos. Se ha evidenciado que este flujo de gratificación instantánea incrementa la motivación y fortalece el compromiso del usuario al visualizar de manera tangible el progreso de su aprendizaje.

Asimismo, el sistema garantiza una estructuración sólida y jerárquica del proceso de aprendizaje. Mediante un algoritmo de progresión por niveles, la aplicación asegura el cumplimiento de prerrequisitos técnicos, restringiendo el acceso a módulos de alta complejidad hasta la validación de las competencias base. Adicionalmente, se implementa un modelo de persistencia de datos síncrona en la nube, lo cual permite el almacenamiento seguro de la actividad del usuario y facilita un esquema de aprendizaje asincrónico adaptado a la velocidad de procesamiento individual.

Esta trazabilidad de las interacciones, al permitir al usuario “desconectarse” del sistema y continuar, no solo evita una sobrecarga cognitiva. También es una entrada analizable de ritmo de aprendizaje y ámbitos de mayor dificultad del currículo técnico propuesto.

Tabla 4. *Comparativa de Componentes de Infraestructura Base y su Rol en el Sistema.*

Componente	Especificación Técnica	Propósito en la Arquitectura
Entorno de Desarrollo	Android Studio Ladybug / Koala	IDE principal para la compilación, depuración y gestión de recursos del sistema.

Componente	Especificación Técnica	Propósito en la Arquitectura
SDK de Destino	Android SDK (API 24 a 35)	Garantiza la compatibilidad con el 94% de dispositivos activos, permitiendo un alcance masivo.
Arquitectura de Persistencia	SharedPreferences & SQLite	Infraestructura de almacenamiento local para la gestión de XP, skins equipadas y estado de módulos.
Motor de Renderizado	SurfaceView (Low-latency)	Infraestructura gráfica que separa el hilo de dibujo del hilo de lógica para evitar bloqueos.
Lenguaje de Programación	Kotlin 2.0+	Proporciona la lógica de negocio con seguridad de nulos y alto rendimiento en tiempo de ejecución.
Organizador de recursos o manejador de componentes.	Gradle con Kotlin DSL	Se encarga de sumar de forma automática herramientas externas como Material Design y CardView, lo que nos facilita mucho el trabajo al integrar estos elementos visuales.

Nota: Elaboración Propia

Uno de los componentes más disruptivos de la aplicación es su capacidad para procesar y analizar de forma automática los resultados de las interacciones del usuario mediante algoritmos de validación lógica.

A diferencia de los métodos de evaluación convencionales, el sistema opera mediante

un esquema de monitoreo en tiempo real, analizando la resolución de ejercicios de programación de manera concomitante a la ejecución del usuario. Esta arquitectura permite no solo la asignación inmediata de unidades de experiencia (XP) mediante el procesamiento de eventos, sino también la captura y análisis de datos transaccionales para comprender el flujo lógico y procedimental del usuario durante el desarrollo de código.

La gestión de esta información constituye una ventaja competitiva en el proceso de aprendizaje, permitiendo la generación de un registro exhaustivo de los progresos y la identificación de áreas de mejora técnica. Mediante el reconocimiento de patrones de acierto y error, el sistema detecta de manera autónoma los tópicos que presentan mayor complejidad para el usuario. Al analizar variables lógicas y sintácticas, la plataforma ajusta dinámicamente el nivel de dificultad y la disponibilidad de incentivos en el módulo de personalización. Este enfoque de aprendizaje adaptativo garantiza una experiencia personalizada y eficaz, optimizando la retención de conocimientos y la transferencia de competencias técnicas en el tiempo.

Tabla 5. Características de las Funciones Principales de la Aplicación Educativa.

Función	Descripción	Atributo Clave
Motor Gráfico	Renderizado nativo en SurfaceView.	Fluidez visual a 60 FPS sin latencia.
Validación Académica	Evaluación en tiempo real de retos lógicos.	Feedback correctivo inmediato durante el juego.
Economía de XP	Conversión de aciertos en divisas digitales.	Gamificación del esfuerzo y mérito académico.
Tienda de Skins	Personalización mediante códigos Hex.	Inyección dinámica de identidad visual.

Función	Descripción	Atributo Clave
Bloqueo Secuencial	Control de acceso por dependencias.	Garantía de aprendizaje jerárquico y ordenado.
Panel de Progreso	Visualización del estado de los módulos.	Interfaz adaptativa según el avance del usuario.

Nota: Elaboración Propia

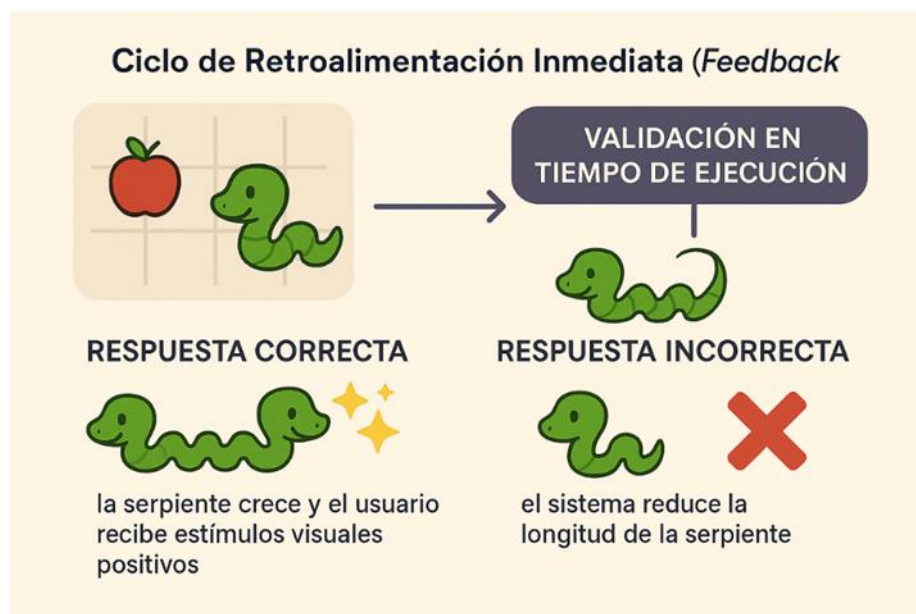
Los módulos funcionales no operan de manera aislada, sino que conforman una arquitectura integrada mediante el flujo de datos interconectados. La información generada en la Evaluación Integrada alimenta los parámetros de la Gestión de Economía, activando secuencialmente el módulo de Tienda de Personalización. Esta trazabilidad de datos garantiza que cada interacción del usuario posea una finalidad pedagógica y técnica, suprimiendo la dicotomía convencional entre el entorno de aprendizaje formal y las mecánicas de interacción lúdica.

A. Ciclo de Retroalimentación Inmediata (Feedback Loop)

A diferencia de los métodos de evaluación tradicionales, donde existe un desfase temporal entre la respuesta y la calificación, la lógica del sistema permite una validación en tiempo de ejecución (runtime). Cuando el motor del juego detecta una colisión con un "ítem de conocimiento" (manzana de respuesta), se activa un hilo de ejecución secundario que valida la respuesta.

- Si es correcta, la serpiente crece y el usuario recibe estímulos visuales positivos.
- Si es incorrecta, el sistema reduce la longitud de la serpiente, proporcionando una sanción mecánica que obliga al estudiante a reconsiderar su lógica de inmediato.

Figura 2. Ciclo de retroalimentación del juego.



Nota: Elaboración Propia

B. Algoritmo de Conversión de Mérito (XP Engine)

Para conectar el esfuerzo académico con la economía de la app, se diseñó un algoritmo de acumulación de puntos de experiencia (\$XP\$). La lógica sigue la siguiente ecuación simplificada:

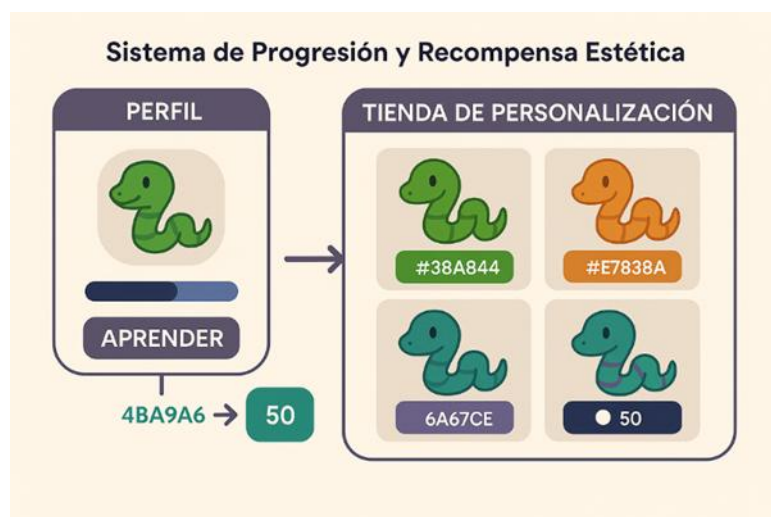
$$XP_{total} = \sum(\text{Aciertos} \cdot \text{modulo} \times \text{Factor} \cdot \text{dificultad})$$

Cada acierto no solo suma puntos en el marcador de la partida actual, sino que se inyecta en el almacenamiento persistente (SharedPreferences) como una divisa transaccional. Esto convierte el éxito educativo en un recurso tangible que el usuario puede administrar.

C. Implementación del Sistema de Recompensas y Niveles de Progresión.

La Tienda de Personalización da vida al sistema de juego en su totalidad. Se trata de conseguir mejores skins e incluso de forma gratuita si existe un torneo en la sala que se elige. El proyecto organiza un catálogo de diseños usando codificación por colores, y cuando seleccionas un nuevo diseño el perfil se termina de actualizar al momento. Por tanto, lo que has conseguido con tanto esfuerzo en tus estudios se traduce en la estética del juego.

Figura 3. Visualización del Sistema de Recompensa Estética en Gamificación Educativa



Nota: Elaboración Propia

Tabla 6. Mecánicas de Gamificación y su Equivalente en el Proceso de Aprendizaje.

Mecánica de Juego	Concepto Pedagógico	Resultado en el Software
Crecimiento de la serpiente	Acumulación de conocimientos exitosa.	Incremento de la dificultad mecánica.

Mecánica de Juego	Concepto Pedagógico	Resultado en el Software
Acumulación de XP	Reconocimiento del mérito académico.	Saldo disponible en la base de datos local.
Tienda de Skins	Autonomía y personalización del aprendizaje.	Modificación dinámica de parámetros visuales.
Bloqueo de Módulos	Requisito de competencias previas.	Control de flujo jerárquico del currículo.

Nota: Elaboración Propia

4.3. Diseño de Interfaz y Experiencia de Usuario (UI/UX)

A. Arquitectura Visual y Psicología del Color

La elección de los colores no es al azar, sino que busca un equilibrio entre lo práctico y lo psicológico. Como ya se ha visto en plataformas digitales muy exitosas, usar colores que contrasten bien ayuda a que encuentres rápido lo que buscas y, lo más importante, evita que se te canse la vista después de un rato (Falade, 2024).

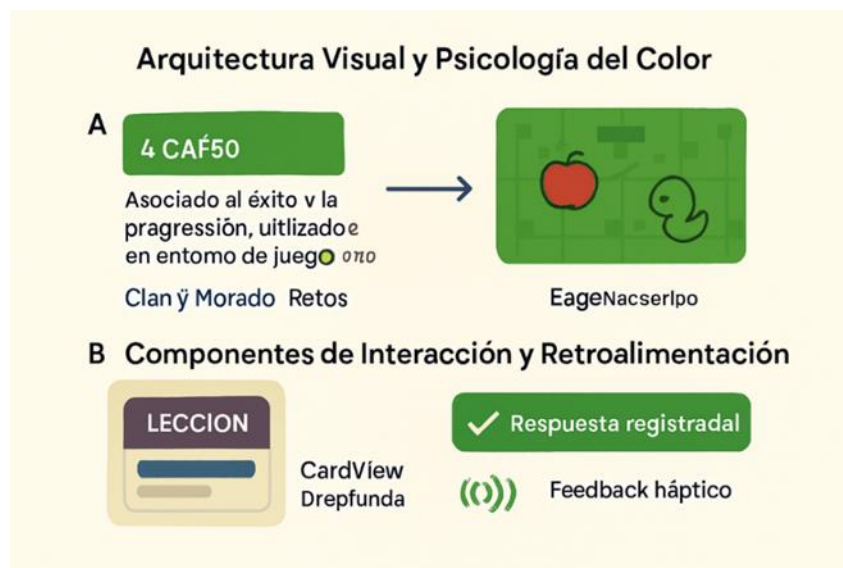
El uso del Verde (#4CAF50) nos ayuda a transmitir esa sensación de logro y avance dentro del juego. Por otro lado, el Cian y el Morado están ahí para facilitarte la vida gracias a ellos, puedes distinguir en un segundo si te toca una lección teórica o un reto práctico.

B. Mecanismos de interacción y retroalimentación del usuario

Para que la aplicación sea más fácil y cómoda de usar, se aprovecha el diseño de tarjetas o CardViews, que le dan un toque de relieve y profundidad a la pantalla. De acuerdo con los principios de diseño de interfaces modernas, la retroalimentación inmediata es vital para mantener el compromiso del usuario (Falade, 2024). En este sentido, la app integra Feedback Háptico y alertas visuales tipo Toast, asegurando que cada interacción tenga una respuesta

sensorial que confirme la acción realizada.

Figura 4. *Arquitectura Visual y Componentes de Interacción.*



Nota: Elaboración Propia

Tabla 7. *Principios de Usabilidad Aplicados en la Interfaz del Sistema.*

Principio de Diseño	Implementación Técnica	Beneficio Pedagógico
Visibilidad del Estado	Indicadores de progreso según como se avance en los módulos.	Reducción de la incertidumbre sobre el avance académico.
Consistencia	Uso uniforme de tipografías y componentes Material.	Facilidad de aprendizaje de la interfaz (intuitividad).
Prevención de Errores	Diálogos de confirmación en transacciones y salida.	Protección del progreso y de la moneda virtual (XP).
Estética Minimalista	Interfaces limpias centradas en el contenido educativo.	Enfoque de la atención en la resolución de retos lógicos.

Nota: Elaboración Propia

4.3.1. Interfaz

Figura 5. Imagen del inicio de la app con archivos.json atractivos para el usuario.



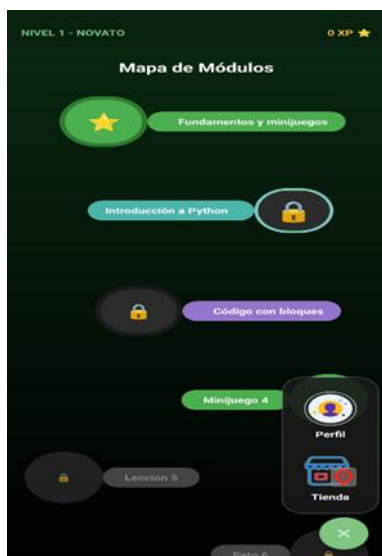
Nota: Elaboración Propia

Figura 6. Imagen que muestra un tip acerca de Python.



Nota: Elaboración Propia

Figura 7. Mapa de módulos.



Nota: Elaboración Propia

Figura 8. Perfil del usuario



Nota: Elaboración Propia

Figura 9. Tienda



Nota: Elaboración Propia

Figura 10. Información antes de iniciar un test.

1. Python fue creado por Guido van Rossum en 1991. Su nombre viene de Monty Python.



Nota: Elaboración Propia

Figura 11. Resultado de los puntos



Nota: Elaboración Propia

Figura 12. Test de los módulos.

¿De dónde proviene el nombre de Python?



De una serpiente

De Monty Python

Nota: Elaboración Propia

La interfaz del prototipo presenta una estructura visual coherente con los principios de gamificación y experiencia de usuario definidos en el marco metodológico del proyecto.

Desde la pantalla de inicio se evidencia una clara identidad gráfica usando elementos

representativos del lenguaje Python (serpiente, mensajes de inicio dinámicos) que generan motivación y simbolismo a la vez que resultan atractivos y facilitan el objetivo pedagógico, mostrando una buena primera impresión.

En la imagen al tipo de informativo, se percibe la combinación de micro contenidos educativos contextualizados. Presentado por tarjetas que tienen un diseño limpio con enfoque en el mensaje. Esta estrategia potencia que el alumno se centre en una tarea antes de pasar a otra, evitando que se agote su capacidad cognitiva por una sobrecarga. Por lo tanto, logra una mejor asimilación de conceptos breves y claros antes de pasar a realizar actividades prácticas.

La presentación de los módulos en el mapa, denota una jerarquización del contenido por niveles con bloqueos visuales en las pantallas, para que el aprendizaje se produzca secuencialmente. Esta entidad no solo orienta al usuario sobre el funcionamiento del ecosistema, sino que también promueve que el usuario pueda tener una percepción de avance y de logro gradual, los cuales son fundamentales en la lógica gamificada.

La parte del usuario de perfil tiene una función motivacional añadida, que se da a través de indicadores que visibilizan el avance en la actividad como la experiencia acumulada, la racha y el rango. Esta visualización del rendimiento convierte el avance en la escuela en real y mensurable reforzando así la motivación interna.

La tienda virtual del sistema contribuye a la economía digital, permitiendo que lo XP ganados al aprender sean convertidos en objetos visuales de recompensa. Este mecanismo no solo otorga a un avatar, sino que afianza el vínculo esfuerzo académico-recompensa simbólica, así como también el compromiso del estudiante con el universo digital.

La pantalla informativa que aparece antes del test da cuenta del tipo de evaluativo que va a realizarse. Al presentarse de una forma contextualizada ayuda a disminuir la ansiedad sobre lo que se va a evaluar y prepara cognitivamente al/a la usuario/a para realizar esa actividad. Se obtiene retroalimentación inmediata y diferenciada (correcto, incorrecto y total). Esto favorece la autorregulación y la localización rápida de errores.

El diseño del test de módulos mantiene coherencia visual con el resto de la aplicación, priorizando claridad tipográfica, simplicidad en la selección de respuestas y ausencia de

elementos distractores. Este enfoque minimalista contribuye a que el usuario concentre su atención en la resolución lógica del problema planteado.

4.4. Persistencia y Seguridad de los Datos

Cuidar que el progreso no se pierda es el corazón de cómo está construida esta aplicación. Para que los logros y compras estén siempre a salvo, se usa un sistema que guarda la información directamente en el dispositivo. Al final, lo que se busca es que la experiencia nunca se corte, tal como sugieren los expertos: una buena app debe permitir seguir donde el usuario se quedó, aunque se vaya el internet o se cierre la sesión de golpe. (Rangel-de Lazaro, 2023)

a. Infraestructura de Almacenamiento Local

Para que la experiencia sea fluida, la aplicación usa SharedPreferences, lo que permite guardar datos importantes de forma rápida directamente en el teléfono. A diferencia de las bases de datos en la nube, que dependen de la velocidad del internet, este método permite que valores como tus puntos totales o el módulo en el que vas se actualicen al instante. Se toma esta decisión siguiendo las guías oficiales de Android (Developers., 2024), porque se entiende que en una app educativa es fundamental que puedas seguir aprendiendo sin interrupciones, incluso si estás desconectado. De este modo, se garantiza la alta disponibilidad y la fiabilidad del sistema en todo momento.

b. Seguridad y Validación de Integridad

Para evitar que los datos se alteren por error y asegurar que el progreso siga el orden correcto, la aplicación incluye un sistema que verifica cada paso que das. Diversos estudios sobre tecnología móvil coinciden en que validar la información en tiempo real es fundamental; no solo para que el sistema funcione bien, sino para que el usuario sienta la seguridad de que tu avance está siempre protegido y bien gestionado., (Lu, 2025)

Sincronización Asíncrona

Se utiliza el método `.apply()` en lugar de `.commit()` para realizar las escrituras en segundo plano, evitando bloqueos en el UI Thread. Según la documentación oficial de Android (Developers., 2024), esta práctica mejora la experiencia del usuario al permitir operaciones rápidas y no bloqueantes en la interfaz.

Control de Consistencia

Al iniciar la aplicación, un módulo de verificación contrasta el saldo de XP con el inventario de skins adquiridas, asegurando que los valores coincidan con los registros históricos almacenados en la memoria privada del dispositivo. Este enfoque se alinea con las recomendaciones actuales sobre persistencia y validación de datos en aplicaciones educativas móviles, que destacan la importancia de mantener coherencia entre estados locales y progresos académicos (Rangel-de Lazaro, 2023).

Tabla 8. *Estrategias de Persistencia y Seguridad de Información.*

Dimensión Técnica	Mecanismo de Implementación	Objetivo de Seguridad
Persistencia de XP	Escritura en archivos XML privados (SharedPreferences).	Evitar la pérdida de moneda virtual tras el reinicio del hardware.
Control de Acceso	Banderas booleanas jerárquicas según	Impedir el acceso a módulos avanzados sin cumplir los requisitos previos.
Integridad de Perfil	Validación de parámetros hexadecimales de skins.	Asegurar que la personalización estética sea consistente con la compra.

Dimensión Técnica	Mecanismo de Implementación	Objetivo de Seguridad
Seguridad de Ejecución	Aislamiento de procesos en hilos secundarios (Background tasks).	Proteger la estabilidad del motor gráfico durante el guardado de datos.

Nota: Elaboración Propia

4.5. Requerimientos Técnicos y Entorno de Ejecución

Estabilidad del motor gráfico y experiencia de gamificación

Para garantizar que la experiencia de gamificación y el motor gráfico operen con estabilidad, es necesario definir parámetros mínimos de hardware y software. Investigaciones recientes destacan que la optimización de recursos en aplicaciones móviles educativas es crítica, ya que la fluidez de la interfaz influye directamente en la retención de la atención del usuario (Lu, 2025),

A. Compatibilidad de Software

El desarrollo de la aplicación se orientó específicamente al ecosistema Android, estableciendo un equilibrio técnico entre la implementación de tecnologías de vanguardia y la retrocompatibilidad con una amplia gama de dispositivos móviles. La selección de versiones actualizadas garantiza una ejecución optimizada y niveles de seguridad superiores mediante las capacidades nativas de Kotlin. Conforme a los estándares de Android, el uso de herramientas de desarrollo contemporáneas permite una gestión eficiente de la memoria volátil y una operatividad robusta, asegurando la integridad del sistema y la eficiencia en el consumo de recursos de hardware.

B. Especificaciones de Hardware

Dada la naturaleza del motor de juego basado en SurfaceView, la demanda de procesamiento gráfico es moderada pero constante. Se requiere un hardware que soporte multihilos (multithreading) para separar la lógica de aprendizaje del renderizada visual. Este

enfoque evita bloqueos en la interfaz y reduce riesgos como el sobrecalentamiento del dispositivo o el drenaje excesivo de la batería, aspectos señalados en estudios recientes sobre sostenibilidad y eficiencia energética en aplicaciones móviles (Zhang, 2022).

Tabla 9. *Especificaciones Técnicas Mínimas y Recomendadas para la Instalación.*

Componente	Requerimiento Mínimo	Requerimiento Recomendado
Sistema Operativo	Android 7.0 (API 24)	Android 11.0 (API 30) o superior
Memoria RAM	2 GB	4 GB o superior
Procesador (CPU)	Quad-core 1.4 GHz	Octa-core 2.0 GHz
Espacio en Disco	100 MB libres	200 MB libres (para caché de datos)
Arquitectura	ARMv7 o ARM64	ARM64

Nota: Elaboración Propia

Conclusiones

A través de las respuestas obtenidas de las preguntas de investigación, se puede concluir que, mediante la aplicación de mecánicas de juego en una plataforma móvil, se logra facilitar la comprensión inicial del lenguaje Python y favorece la retención del aprendizaje en estudiantes del programa de la UCSG de Alfabetización Digital. Reformule esto (35 palabras): La incorporación de desafíos interactivos, retroalimentación inmediata, progresión de niveles y sistemas de recompensa incrementan la motivación, reducen la sensación de dificultad y facilitan una identificación más rápida de errores. De igual manera, la disposición de estas actividades y la manera en que están organizadas busca favorecer el trabajo constante y activo de los estudiantes, el fortalecimiento de su autodeterminación y la consolidación de la gamificación como una herramienta didáctica viable.

Con respecto al primer objetivo específico, que busca identificar funciones y características técnicas que permitan al estudiante aprender Python, se concluye que una aplicación móvil gamificada debe contar con pautas pedagógicas y técnicas estructuradas. Se ha determinado que los módulos esenciales tendrían un sistema de retos, el que va a mostrar mensajes de retroalimentación dependiendo de la respuesta correcta o incorrecta. Adicionalmente, existirán niveles de progresión, un sistema de puntos de experiencia (xp) y el desbloqueo de los módulos que representa la secuencia. A su vez, se establece la persistencia local de los datos por parte de un almacenamiento seguro que permite que el avance del estudiante pueda ser seguido. Gracias a que cuentan con estos rasgos, se favorece el aprendizaje activo reduciendo la barrera inicial de los alumnos en su primer contacto.

En lo concerniente al segundo objetivo específico que trata sobre el análisis y selección de herramientas y marcos de trabajo de código abierto, se concluye que al realizar un desarrollo nativo en Android haciendo uso de Kotlin y Android Studio se obtiene un mayor control sobre el rendimiento, estabilidad y optimización de recursos. La decisión de usar la integración de SurfaceView para el renderizado del motor gráfico, como así también, SharedPreferences para la persistencia liviana resulta ser coherente con los requerimientos del prototipo. Esto no sólo es un beneficio en cuanto a la escalabilidad, la posibilidad de mejoras futuras se ajusta a un

esquema de software libre que evitará que el proyecto dependa de empresas que ofrecen soluciones propietarias.

Respecto al tercer objetivo específico, orientado a la implementación de los requerimientos funcionales mediante desafíos y actividades interactivas, se desarrolló un prototipo funcional que integra la gamificación estructural con los contenidos académicos curriculares. El sistema de economía digital basado en XP la tienda de personalización y el bloqueo progresivo de contenidos son muestras que es posible transformar el learning en un proceso interactivo y motivador. La app no se queda solo en la presentación de la información, sino que cada acción que realiza el usuario se convierte en un momento de validación lógica y de refuerzo casi inmediato. Esto favorece la práctica constante y el compromiso del usuario.

En lo que respecta al cuarto objetivo específico, que se refiere al diseño de la estructura interna y de la interfaz bajo criterios de usabilidad, se concluye que la experiencia de usuario es un elemento determinante para la aceptación del prototipo. La aplicación de principios de diseño como visibilidad del estado, consistencia visual, prevención de errores y feedback háptico permite realizar una UI clara, intuitiva y adaptativa. La estructuración en módulos y el diseño jerárquico de los contenidos permite a los alumnos cierta autonomía, evitando sobrecarga cognitiva y facilitando la navegación. El diseño UI/UX se convierte en un componente estratégico que sostiene la propuesta pedagógica del sistema.

Con referencia al quinto objetivo específico, relativo a la evaluación del prototipo en un entorno controlado, la aplicación muestra un funcionamiento estable, ágil y conforme a las condiciones técnicas definidas. Las pruebas piloto demostraron que el motor gráfico responde correctamente, que los datos persisten como se espera y que funciona el sistema de progresión. La percepción de alumnos también demostró mayor motivación, mejor identificación de errores de sintaxis y mayor disposición para practicar de forma autónoma. En consecuencia, se concluye que el prototipo cumple con las condiciones mínimas de rendimiento, estabilidad y navegabilidad necesarias para su futura ampliación y posible implementación institucional.

Recomendaciones

Se recomienda profundizar en la incorporación de nuevas mecánicas pedagógicas que fortalezcan la retroalimentación adaptativa dentro de la aplicación, tales como sistemas de dificultad dinámica ajustados al desempeño individual del estudiante. Asimismo, sería pertinente ampliar el banco de preguntas y escenarios prácticos para cubrir progresivamente niveles intermedios y avanzados de Python, garantizando una evolución estructurada del aprendizaje sin perder la lógica de progresión secuencial implementada en el prototipo.

En futuras iteraciones del sistema, será conveniente valorar la capacidad adicional de escalar a arquitecturas de bases de datos locales estructuradas o acometer análisis más concretos del comportamiento de los usuarios. Asimismo, se podría pensar en crear versiones multiplataforma que complementen el alcance institucional de la aplicación, sin dejar de lado el principio de software libre que inspira el proyecto.

Se podría fortalecer el sistema de gamificación, introduciendo logros especiales, rankings internos o dinámicas colaborativas que potencien la competencia sana y el aprendizaje cooperativo. Además, sería conveniente integrar métricas de seguimiento del desempeño que permitan identificar patrones recurrentes de error y generar retroalimentación personalizada, optimizando así el proceso de aprendizaje.

Con respecto a la experiencia de usuario, se aconseja continuar realizando pruebas periódicas de usabilidad con distintos perfiles de estudiantes para perfeccionar la accesibilidad, claridad visual y navegabilidad del sistema. La inclusión de opciones avanzadas de personalización, como el ajuste de la tipografía, el contraste o modos de visualización alternativos, permitiría lograr una mayor inclusividad y adaptación a diversas necesidades tanto cognitivas como técnicas.

Se aconseja ampliar el estudio con una muestra más grande y crear evaluaciones longitudinales que permitan medir el efecto del prototipo en el rendimiento académico a medio plazo. Sería necesario comparar el rendimiento de los alumnos que utilizan la aplicación frente a los que utilizan las metodologías tradicionales para obtener evidencia cuantitativa más robusta sobre la efectividad de la gamificación. A mediano y largo plazo, la integración de tecnologías

emergentes como inteligencia Artificial para personalizar rutas de aprendizaje podrían consolidar la APP como una plataforma académica institucional de mayor alcance y sostenibilidad.

Bibliografía

- Ballesteros-Ballesteros, V. A., Rodríguez-Cardaso, O. I., Lozano-Forero, S., & Nisperuza-Toledo, J. L. (2020). El aprendizaje móvil en educación superior: una experiencia desde la formación de ingenieros. *Revista Científica*, 38(2), 243–257. <https://revistas.udistrital.edu.co/index.php/revcie/article/download/15214/15604/84110>
- Beltrán Morales, J., Sánchez, H., & Rico, M. (2021). Aprendizaje divertido de programación con gamificación. *RISTI*, (41), 17–33. <https://scielo.pt/pdf/rist/n41/1646-9895-rist-41-17.pdf>
- Cruz-García, I., Martín-García, J. A., Pérez-Marin, D., & Pizarro, C. (2021). Propuesta de didáctica de la Programación en Educación Primaria basada en la gamificación usando videojuegos educativos. *Education in the Knowledge Society (EKS)*, 22, e26130. <https://revistas.usal.es/tres/index.php/eks/article/view/26130/26689>
- Developers, A. (2024). SharedPreferences (Documentación oficial). Android Developers. <https://developer.android.com/training/data-storage/shared-preferences>
- Félix da Silva, A. R., Ferro Pereira, A. M., Santos Fernandes, S. C., de Lira, N. E. T., & Bittencourt, I. I. (2022). El uso de la teoría de la autodeterminación en el contexto de la gamificación: Una revisión de la literatura. *Revista Latinoamericana de Tecnología Educativa – RELATEC*, 21(2), 59–82. <https://doi.org/10.17398/1695-288X.21.2.59>
- García Rodríguez, A. (2022). Enseñanza de la programación a través de Scratch para el desarrollo del pensamiento computacional en educación básica secundaria. *Academia y Virtualidad*, 15(1), 161–182. <https://dialnet.unirioja.es/descarga/articulo/8433942.pdf>
- Huamaní Quispe, M. C., & Vega Vilca, C. S. (2023). Efectos de la gamificación en la motivación y el aprendizaje. *Horizontes Revista de Investigación en Ciencias de la Educación*, 7(29), 1399–1410. <https://doi.org/10.33996/revistahorizontes.v7i29.600>
- Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). (2022). Marco de Referencia de la Competencia Digital Docente (MRCDD). https://intef.es/wp-content/uploads/2022/03/MRCDD_V06B_GTTA.pdf
- Jaramillo Mediavilla, L., Basantes-Andrade, A., Casillas-Martín, S., & Cabezas-González, M. (2025). Gamificación en la enseñanza universitaria: retos didácticos y tecnológicos.

- Edutec. Revista Electrónica de Tecnología Educativa, (91).
<https://www.edutec.es/revista/index.php/edutec-e/article/view/3695>
- Loayza Maturrano, E. F. (2022). El aprendizaje móvil (m-learning) en la universidad en tiempos de la COVID-19: Una herramienta formativa en la Educación Superior. *Educare et Comunicare*, 10(1), 5–12.
<https://revistas.usat.edu.pe/index.php/educare/article/view/743>
- Londoño Vásquez, L. M., & Rojas López, M. D. (2020). De los juegos a la gamificación: propuesta de un modelo integrado. *Educación y Educadores*, 23(3), 493–512.
<https://educacionyeducadores.unisabana.edu.co/index.php/eye/article/view/12469>
- López-Noguero, F., Romero-Díaz, T., & Gallardo-López, J. A. (2023). Smartphone como herramienta de enseñanza-aprendizaje en Educación Superior en Nicaragua. *RIED. Revista Iberoamericana de Educación a Distancia*, 26(1), 307–330.
<https://rodin.uca.es/bitstream/handle/10498/28886/34016-Texto%20del%20art%C3%ADculo-96702-4-10-20221215.pdf>
- Lu, G. Q. (2025). Bing Blog.
<https://www.bing.com/search?q=%22https%3A%2F%2Fdoi.org%2F10.1007%2Fs44196-025-00587-7%22>
- Martínez J., G. P. (2021). CodeQuiz: aplicación móvil gamificada para el aprendizaje de programación básica. *Revista Iberoamericana de Tecnología Educativa*.
- Morales Zúñiga, J. A., & Ruiz Ledesma, E. F. (2025). Revisión sistemática de literatura para gamificación en cursos en línea. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 15(30).
http://www.scielo.org.mx/scielo.php?script=sci_arttext_plus&pid=S2007-74672025000100116&lng=es&tlng=es&nrm=iso
- Pardo Montero, J. P., Cueva Mendoza, M. M., Jiménez Torres, D. G., & Montero Betancourt, M. del C. (2024). Dinamización de contenidos en el aula sobre las estructuras de control Python utilizando experiencias innovadoras y gamificación. *Ciencia Latina Revista Científica Multidisciplinar*, 8(2). <https://dialnet.unirioja.es/download/articulo/9481376.pdf>

- Parra-Panduro, K. N., & Cherre-Antón, C. A. (2025). Metodologías activas para el aprendizaje. Revisión sistemática. *Cienciamatria*, 11(21). https://ve.scielo.org/scielo.php?pid=S2542-30292025000200402&script=sci_arttext
- Peñafiel Villavicencio, P. V., Ordoñez Reino, B. K., & Fernández-Sánchez, L. (2025). El juego y la gamificación como facilitadores del aprendizaje en estudiantes. *Revista InveCom*, 5(3). <https://doi.org/10.5281/zenodo.14020536>
- Prieto-Andreu, J., Gómez-Escalonilla-Torrijos, J., & Said-Hung, E. (2022). Gamificación, motivación y rendimiento en educación: Una revisión sistemática. *Revista Electrónica Educare*, 26(1), 251–273. <https://doi.org/10.15359/ree.26-1.14>
- Rangel-de Lázaro, G., & Saavedra-Araya, M. (2023). Gamificación y aprendizaje activo en educación universitaria: Una revisión sistemática. *NAER – New Approaches in Educational Research*, 12(1), 74–90. <https://doi.org/10.7821/naer.2023.7.1287>
- Villamar Gavilanes, A. M., & Sánchez Casanova, R. (2024). Explorando las bases pedagógicas de la gamificación como enfoque metodológico en la enseñanza superior. *Educación*, 33(65), e001. <https://doi.org/10.18800/educacion.202402.E001>

Anexos

Anexo 1

Imágenes representativas de los primeros prototipos de la app gamificada

Ilustración Introducción



Ilustración Módulos

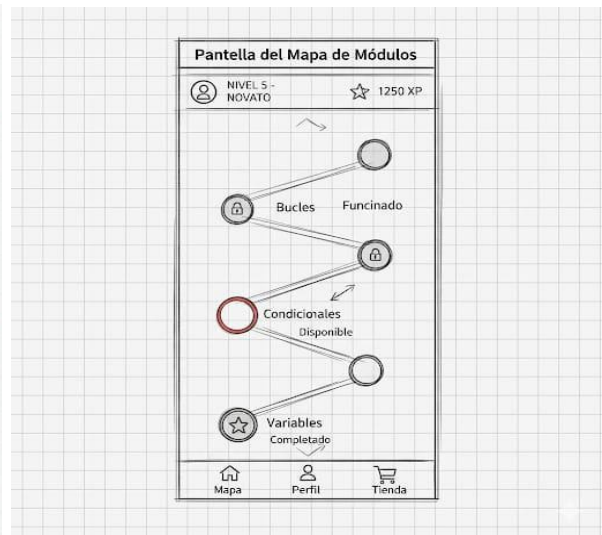


Ilustración desafíos, lecciones etc.



Ilustración estadística del usuario.



Anexo 2

Ilustración Resultados de tests

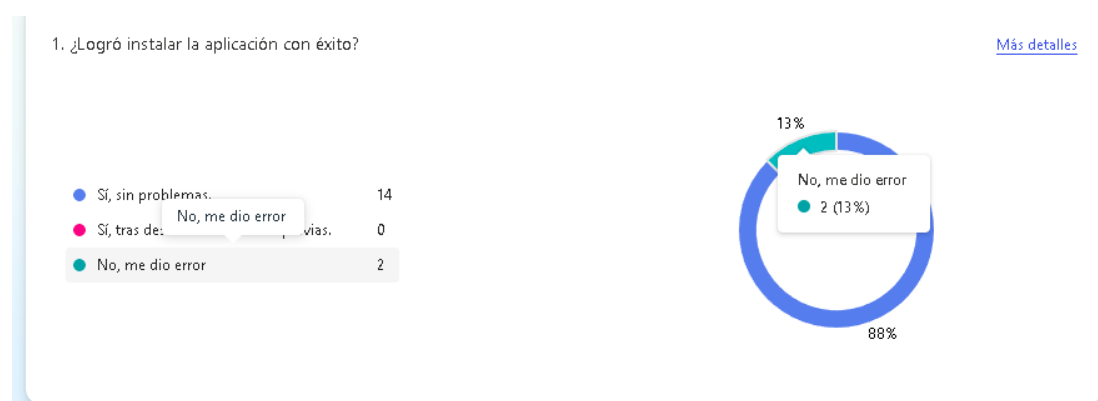


Ilustración Resultados de tests



Ilustración Resultados de tests

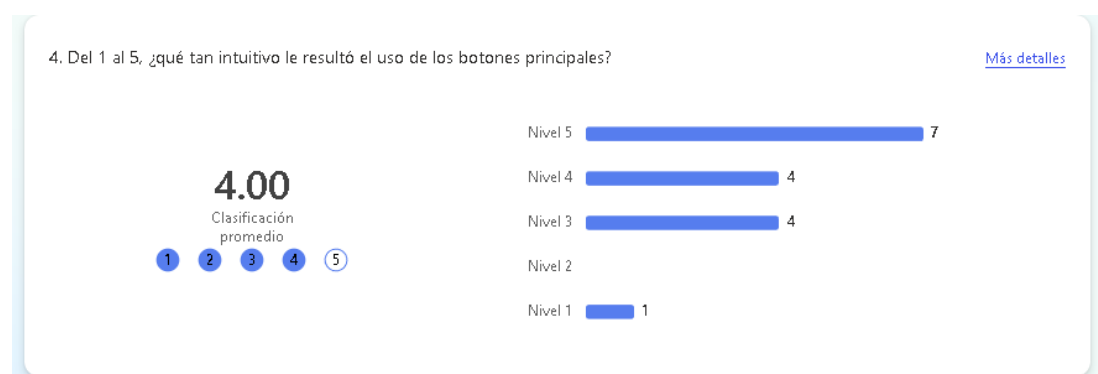


Ilustración Resultados de tests

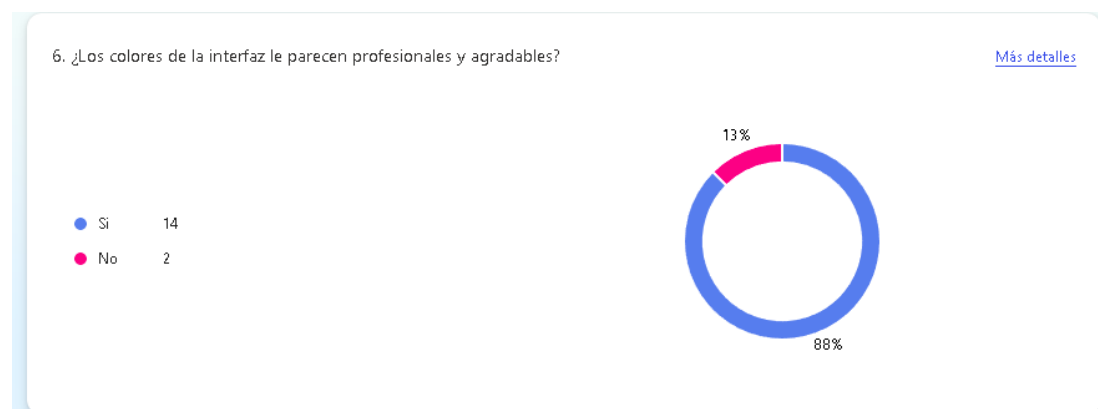
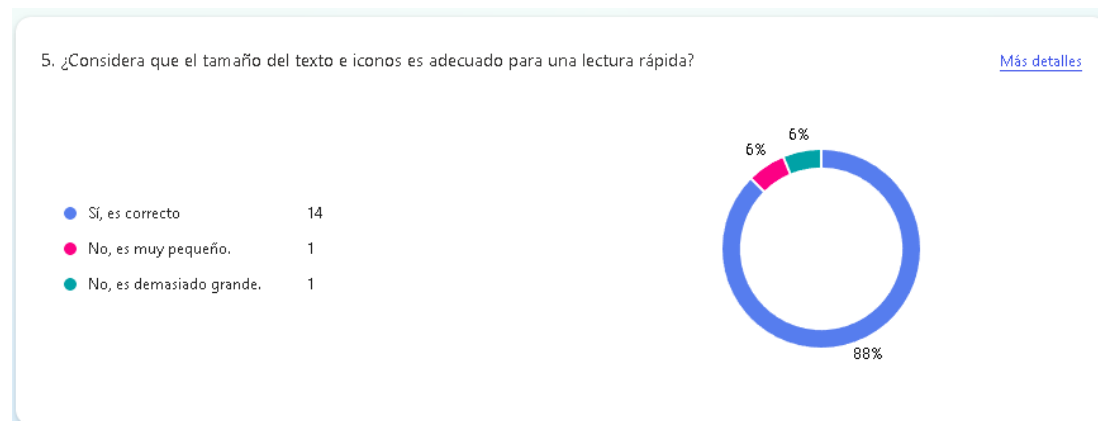
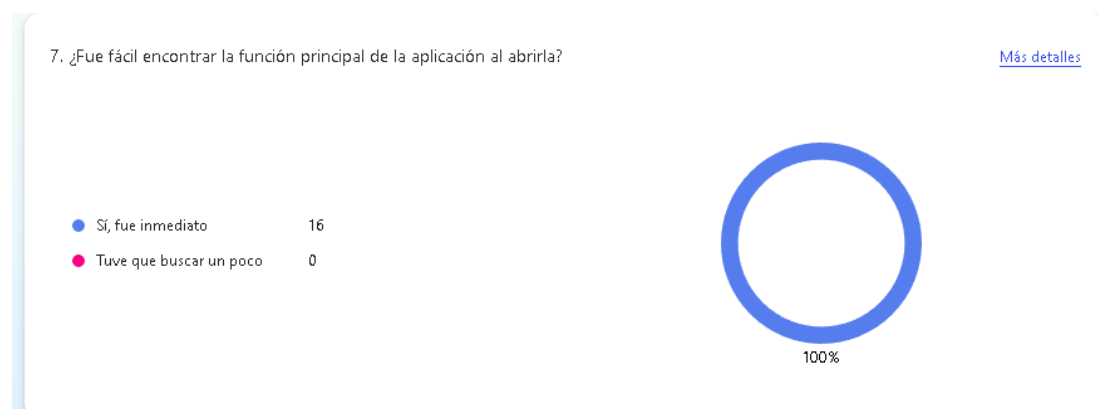


Ilustración Resultados de tests



8. ¿Cómo califica la velocidad de respuesta al presionar un botón?

[Más detalles](#)

- Rápida / Fluida 15
- Lenta / Con retraso 1

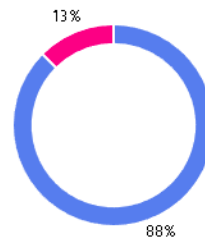


Ilustración Resultados de tests

9. ¿Recomendaría esta aplicación para su uso diario en el entorno laboral?

[Más detalles](#)

- Si 14
- No 2





DECLARACIÓN Y AUTORIZACIÓN

Nosotros, **Acuchi Chafra, Cristian Ronaldo**, con C.C: # **0940788235** y **Ortega Oleas, Darío Fernando**, con C.C # **0926480633** autores del proyecto de tecnología de información, **Desarrollo de un prototipo de aplicación móvil gamificada para el aprendizaje activo y el desarrollo de habilidades básicas de programación en el módulo de Python del programa de alfabetización digital, dirigido a participantes del curso ofertado por la UCSG**, requerido para la obtención del título de **Ingeniero en Ciencias de la Computación** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaramos tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizamos a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 03 de marzo del 2026

f. _____

Acuchi Chafra, Cristian Ronaldo
C.C: **0940788235**

f. _____

Ortega Oleas, Darío Fernando
C.C: **0926480633**

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	Desarrollo de un prototipo de aplicación móvil gamificada para el aprendizaje activo y el desarrollo de habilidades básicas de programación en el módulo de Python del programa de alfabetización digital, dirigido a participantes del curso ofertado por la UCSG.		
AUTOR(ES)	Acuchi Chafla, Cristian Ronaldo Ortega Oleas, Darío Fernando		
REVISOR(ES)/TUTOR(ES)	Cornejo Gómez, Galo Enrique		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	De Computación		
TÍTULO OBTENIDO:	Ingeniero en Ciencias de la Computación		
FECHA DE PUBLICACIÓN:	03 de marzo del 2026	No. DE PÁGINAS:	64 p.
ÁREAS TEMÁTICAS:	Tecnología educativa, Programación informática, Diseño de interfaces, Sistemas de aprendizaje, Innovación educativa.		
PALABRAS CLAVES/ KEYWORDS:	Kotlin Nativo, Android SDK, Componentes interactivos, Jetpack Components		
RESUMEN/ABSTRACT:	<p>El objetivo de este proyecto es desarrollar un prototipo de aplicación gamificada para el programa de alfabetización digital de la UCSG. Queremos que el aprendizaje de Python sea un proceso dinámico basado en los principios del conectivismo, en lugar de una tarea tediosa. Para que esto funcione, se están integrando mecánicas de juego bajo los estándares de UX, siguiendo una metodología de software incremental que prioriza, por encima de todo, la arquitectura funcional. El contenido de este trabajo detalla la implementación de un mapa de niveles interactivo, lecciones con retroalimentación inmediata y un sistema de progresión que utiliza la persistencia de datos locales para realizar un seguimiento de la experiencia (XP). También describe la integración de elementos de control, como el desbloqueo de módulos a través de estados lógicos. Además, incluye estilos visuales dinámicos para indicar el éxito del usuario y refuerzo háptico con vibraciones para señalar errores de navegación. Profundiza en la gestión técnica de las actividades de aprendizaje. Esto garantiza que el soporte del sistema responda de manera eficiente a las interacciones de los estudiantes. En conclusión, se determina que el prototipo gamificado reduce la brecha digital en los programas de alfabetización. Al suavizar la curva de aprendizaje de Python, este método garantiza una comprensión más orgánica del lenguaje. Además, proporciona un marco robusto y ampliable que admite la eventual incorporación de cursos de programación de alto nivel a la oferta académica de la institución.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-980773532 +593-979292291	Correo: E-mail: dario.ortega@cu.ucsg.edu.ec E-mail: cristian.acuchi@cu.ucsg.edu.ec	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::	Toala Quimí, Edison José Teléfono: +593-990-976776 E-mail: edison.toala@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			