



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

**FACULTAD DE CIENCIAS DE LA INGENIERÍA**

**CARRERA DE SOFTWARE**

Trabajo de Integración  
Curricular previa la obtención  
del Grado Académico de  
Ingeniero en Software

**Proyecto De Investigación:**

**“APLICACIÓN WEB PARA INCENTIVAR EL APRENDIZAJE DE  
FUNDAMENTOS DE PROGRAMACIÓN DE ESTUDIANTES DE EDUCACIÓN  
BÁSICA MEDIA-SUPERIOR”**

**Autor:**

Aviles Chalaco Diana Stefania

**Director del Proyecto de Investigación:**

Ing. Orlando Erazo, PhD

**QUEVEDO – LOS RÍOS – ECUADOR**

**2024**





## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **AVILES CHALACO DIANA STEFANIA**, declaro que la investigación aquí descrita es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este documento, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Diana Stefania Aviles Chalaco

C.I. 0957246630



## **CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN**

El suscrito, **Ing. Orlando Erazo, Ph.D.**, docente de la Universidad Técnica Estatal de Quevedo, certifica que el estudiante **Aviles Chalaco Diana Stefania**, realizó el Proyecto de Investigación de grado titulado “**Aplicación web para incentivar el aprendizaje de fundamentos de programación de estudiantes de educación básica media-superior**” previo a la obtención del título de Ingeniero en Software, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

---

Ing. Orlando Erazo, Ph.D.

**DIRECTOR DEL PROYECTO  
DE INVESTIGACIÓN**



## CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO

El suscrito, **Ing. Orlando Erazo Moreta, Ph.D.**, mediante el presente cumpla en presentar a usted, el informe del Trabajo de Integración Curricular titulado “**Aplicación web para incentivar el aprendizaje de fundamentos de programación de estudiantes de educación básica media-superior**” presentado por el estudiante **AVILES CHALACO DIANA STEFANIA**, egresado de la Carrera de Software, que fue revisado bajo mi dirección según resolución del Consejo Directivo de la Facultad de Ciencias de la Ingeniería, que se ha desarrollado de acuerdo al Reglamento de la Unidad de Integración Curricular de la Universidad Técnica Estatal de Quevedo y cumple con el requerimiento de análisis del sistema COMPILATIO el cual avala los niveles de originalidad en un 93% y similitud 7%, del trabajo investigativo. Valido este documento para que el estudiante siga con los trámites pertinentes, de acuerdo como lo establece el Reglamento.

 CERTIFICADO DE ANÁLISIS  
magister

Aplicación web para incentivar el aprendizaje de fundamentos de programación

**7%**  
Textos sospechosos

**4% Similitudes**  
1% similitudes entre comillas  
0% entre las fuentes mencionadas  
**3% Idiomas no reconocidos**

Nombre del documento: Aplicación web para incentivar el aprendizaje de fundamentos de programación.docx  
ID del documento: be59dd150f98ec4cf3ababaf91b9a47aa502732b  
Tamaño del documento original: 8,47 MB

Depositante: ORLANDO RAMIRO ERAZO MORETA  
Fecha de depósito: 9/5/2024  
Tipo de carga: interface  
fecha de fin de análisis: 9/5/2024

Número de palabras: 16.924  
Número de caracteres: 111.906

Ing. Orlando Erazo, Ph.D.

**DIRECTOR DEL PROYECTO DE INVESTIGACIÓN**



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIA DE LA INGENIERÍA**  
**CARRERA DE SOFTWARE**

**PROYECTO DE INVESTIGACIÓN:**

“Aplicación web para incentivar el aprendizaje de fundamentos de programación de estudiantes de educación básica media-superior”

Presentado al consejo directivo de facultad como requisito previo a la obtención del título de ingeniero en software.

Aprobado por:

---

**PRESIDENTE DEL TRIBUNAL**

Ing. Guerrero Ulloa Gleiston, Ph.D.

---

**MIEMBRO DEL TRIBUNAL**

Ing. Zambrano Vega Cristian, PhD.

---

**MIEMBRO DEL TRIBUNAL**

Ing. Jaramillo Chuqui Iván, PhD.

QUEVEDO – LOS RÍOS – ECUADOR

2024

## **AGRADECIMIENTO**

Agradezco primeramente a Dios por darme la fortaleza y guía en cada paso de este camino.

A mis padres, David Aviles y Diana Chalaco, les agradezco por su amor incondicional y su constante apoyo. A mis hermanos, Samuel y Cristhian Aviles, gracias por siempre estar ahí para mí.

Un agradecimiento especial a mi mejor amiga Vivian Napa, cuya amistad ha sido un pilar en mi vida.

A mis compañeros de clase, gracias por los momentos compartidos y el apoyo mutuo en nuestra jornada académica.

Por último, pero no menos importante, a mis profesores, cuya sabiduría y dedicación han sido fundamentales en mi formación. Gracias por su paciencia, su enseñanza e inspiración.

A todos, gracias de corazón.

## **DEDICATORIA**

Dedico este trabajo a mis queridos padres, David Aviles y Diana Chalaco, por su amor incondicional y apoyo constante.

A mis hermanos, Samuel y Cristhian Aviles, por estar siempre pendientes de mí y por ser mi soporte.

**Diana**

## RESUMEN EJECUTIVO Y PALABRAS CLAVES

La programación se ha convertido en una habilidad esencial que no sólo promueve carreras técnicas, sino que también mejora el pensamiento lógico y las habilidades de resolución de problemas en diversos campos. A pesar de su importancia, enseñar programación a aprendices nuevos tiene sus desafíos, especialmente a la hora de mantener su interés en los conceptos que se explican. Para abordar estos problemas, se propone una aplicación web en forma de juego serio, diseñada para hacer que el aprendizaje de la programación sea más entretenido y educativo para los estudiantes de educación básica media-superior. Se llevaron a cabo entrevistas con estudiantes y docentes para determinar las características del juego. Además, se crearon prototipos que se probaron con usuarios representativos utilizando un enfoque de diseño participativo. Finalmente, se llevó a cabo una fase de pruebas para evaluar la usabilidad de la aplicación. Los resultados sugieren que la aplicación tiene el potencial de aumentar la motivación al aprendizaje de la programación, lo que sienta una base sólida para futuras investigaciones y avances en el campo de la educación tecnológica.

**Palabras clave:** Educación en programación, usabilidad del software, juegos serios, sala de escape.

## **ABSTRACT AND KEYWORDS**

Programming has become an essential skill that not only promotes technical careers but also enhances logical thinking and problem-solving skills in various fields. Despite its importance, teaching programming to new learners has its challenges, especially when it comes to maintaining their interest in the concepts being explained. To address these issues, a web application in the form of a serious game is proposed, designed to make learning programming more entertaining and educational for upper-middle basic education students. Interviews were conducted with students and teachers to determine the features of the game. In addition, prototypes were created and tested with representative users using a participatory design approach. Finally, a testing phase was carried out to evaluate the usability of the application. The results suggest that the application has the potential to increase motivation for learning programming, laying a solid foundation for future research and advancements in the field of technological education.

**Keywords:** Programming Education, Serious Games, Software Usability, Escape Room.

## TABLA DE CONTENIDO

PORTADA .....	i
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS .....	ii
CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN.....	iii
CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO.....	iv
CERTIFICADO DE APROBACIÓN POR TRIBUNAL DE SUSTENTACIÓN .....	v
AGRADECIMIENTO .....	vi
DEDICATORIA.....	vii
RESUMEN EJECUTIVO Y PALABRAS CLAVES .....	viii
ABSTRACT AND KEYWORDS .....	ix
CÓDIGO DUBLÍN.....	xvii
INTRODUCCIÓN.....	1
CAPÍTULO I.....	3
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN .....	3
1.1.    Problema de investigación.....	4
1.1.1.    Planteamiento del problema .....	4
1.1.2.    Formulación de problema.....	5
1.1.3.    Sistematización del problema.....	5
1.2.    Objetivos.....	6
1.2.1.    Objetivo General.....	6
1.2.2.    Objetivos Específicos .....	6
1.3.    Justificación.....	6
CAPÍTULO II.....	8
FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN .....	8
2.1.    Programación y pensamiento computacional .....	9
2.1.1.    Programación en educación básica.....	9

2.1.2.	Juegos serios y programación.....	11
2.2.	Salas de escape educativas .....	11
2.3.	Diseño participativo.....	13
2.4.	Metodología de desarrollo basado en prototipo .....	13
CAPÍTULO III .....		15
METODOLOGÍA DE LA INVESTIGACIÓN.....		15
3.1.	Localización.....	16
3.2.	Tipos de investigación.....	16
3.3.	Métodos de investigación .....	16
3.3.1.	Método Inductivo .....	16
3.3.2.	Método Analítico.....	16
3.4.	Fuentes de recopilación de información.....	16
3.5.	Diseño de la investigación.....	17
3.5.1.	Obtención de requerimientos.....	17
3.5.2.	Diseño de un prototipo .....	19
3.5.1.	Evaluación de usabilidad .....	20
3.6.	Instrumentos de investigación .....	24
3.7.	Tratamiento de datos .....	24
3.8.	Recursos humanos y materiales.....	24
3.8.1.	Recursos de hardware.....	24
3.8.2.	Recursos de Software .....	25
CAPÍTULO IV .....		26
RESULTADOS Y DISCUSIÓN .....		26
4.1.	Resultados de la obtención de requerimientos .....	27
4.1.1.	Entrevista no estructurada .....	27
4.1.2.	Uso de una inteligencia artificial ChatGPT.....	30
4.2.	Diseño del prototipo .....	31

4.2.1.	Diagrama de casos de uso.....	34
4.2.2.	Diagrama de clases .....	35
4.2.3.	Diagrama de secuencia general .....	36
4.2.4.	Casos de uso extendidos .....	37
4.2.5.	Diagrama de Arquitectura .....	43
4.2.6.	Puntaje del juego .....	44
4.2.7.	Niveles.....	45
4.3.	Evaluación de usabilidad .....	46
4.3.1.	Resultados Grupo A.....	46
4.3.2.	Resultados Grupo B.....	49
4.4.	Discusión .....	51
CAPÍTULO V.....		53
CONCLUSIONES Y RECOMENDACIONES .....		53
5.1.	Conclusiones.....	54
5.2.	Recomendaciones .....	55
CAPÍTULO VI .....		56
BIBLIOGRAFÍA .....		56
CAPÍTULO VII.....		65
ANEXOS .....		65
7.1.	Anexo 1: Invitación a participar en entrevistas iniciales.....	66
7.2.	Anexo 2: Preguntas abiertas de la entrevista no estructurada a docentes.....	67
7.3.	Anexo 3: Preguntas abiertas de la entrevista no estructurada a estudiantes.....	68
7.4.	Anexo 4: Evidencia de entrevista .....	68
7.5.	Anexo 5: Preguntas realizadas a ChatGPT.....	69
7.6.	Anexo 6: Consentimiento de participación y la toma de imágenes.....	70
7.7.	Anexo 7: Cuestionario SUS.....	71
7.8.	Anexo 8: Participante de evaluación de usabilidad .....	72

7.9.	Anexo 9: Planificación académica .....	72
7.9.1.	Nivel 0 : Introducción a python - print() .....	72
7.9.2.	Nivel 1: Variables.....	74
7.9.3.	Nivel 2: Tipos de datos.....	78
7.10.	Anexo 10: Preguntas prueba de conocimiento .....	82

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Estudiantes seleccionados para la evaluación de usabilidad .....	20
<b>Tabla 2</b> Preguntas del cuestionario SUS.....	21
<b>Tabla 3</b> Pasos para realizar evaluación de usabilidad.....	22
<b>Tabla 4</b> Requerimientos del hardware .....	24
<b>Tabla 5</b> Recursos del software.....	25
<b>Tabla 6</b> Requerimientos funcionales .....	27
<b>Tabla 7</b> Requerimientos no funcionales .....	28
<b>Tabla 8</b> Características obtenidas de docentes y estudiantes.....	28
<b>Tabla 9</b> Características obtenidas de ChatGPT .....	30
<b>Tabla 10</b> Caso de uso: Iniciar Juego.....	37
<b>Tabla 11</b> Caso de Uso: Mostrar Mapa de niveles .....	38
<b>Tabla 12</b> Caso de Uso: Seleccionar Nivel .....	40
<b>Tabla 13</b> Caso de uso: Completar Nivel .....	42
<b>Tabla 14</b> Resultados del cuestionario SUS - Grupo A .....	46
<b>Tabla 15</b> Resultados de las preguntas abiertas - Grupo A.....	47
<b>Tabla 16</b> Resultados de la prueba de conocimientos - Grupo A.....	48
<b>Tabla 17</b> Resultados del cuestionario SUS - Grupo B.....	49
<b>Tabla 18</b> Resultados de las preguntas abiertas - Grupo B .....	50
<b>Tabla 19</b> Resultados de la prueba de conocimientos - Grupo B.....	51

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Escala Likert de emojis .....	21
<b>Figura 2</b> Diseño inicial de las habitaciones de la sala de escape .....	32
<b>Figura 3</b> Diseño del prototipo nivel medio .....	33
<b>Figura 4</b> Diseño inicial de la integración de Problemas de Programación.....	33
<b>Figura 5</b> Casos de uso del sistema.....	34
<b>Figura 6</b> Diagrama de clases.....	35
<b>Figura 7</b> Diagrama de secuencia.....	36
<b>Figura 8</b> Diagrama de secuencia para iniciar juego.....	38
<b>Figura 9</b> Diagrama de secuencia del mapa de niveles .....	39
<b>Figura 10</b> Diagrama de secuencia para jugar niveles .....	41
<b>Figura 11</b> Diagrama de secuencia del menú de opciones.....	43
<b>Figura 12</b> Diagrama de arquitectura .....	44

## ÍNDICE DE ANEXOS

<b>Anexo 1</b> Evidencia de entrevista inicial .....	68
<b>Anexo 2</b> Preguntas realizadas a ChatGPT.....	69
<b>Anexo 3</b> Consentimiento de participación .....	70
<b>Anexo 4</b> Cuestionario SUS.....	71
<b>Anexo 5</b> Participante de evaluación de usabilidad.....	72

## CÓDIGO DUBLÍN

<b>Título:</b>	Aplicación Web Para Incentivar El Aprendizaje De Fundamentos De Programación De Estudiantes De Educación Básica Media-Superior			
<b>Autor:</b>	Aviles Chalaco Diana Stefania			
<b>Palabras Clave:</b>	Educación en programación	Usabilidad del software	Juegos serios	Sala de Escape
<b>Fecha de publicación:</b>	Mayo, 2024			
<b>Director del proyecto:</b>	Ing. Erazo Moreta Orlando Ramiro, Ph.D.			
<b>Editorial:</b>	Quevedo – UTEQ “La María”, 2024			
<b>Resumen:</b>	<p>La programación se ha convertido en una habilidad esencial que no sólo promueve carreras técnicas, sino que también mejora el pensamiento lógico y las habilidades de resolución de problemas en diversos campos. A pesar de su importancia, enseñar programación a aprendices nuevos tiene sus desafíos, especialmente a la hora de mantener su interés en los conceptos que se explican. Para abordar estos problemas, se propone una aplicación web en forma de juego serio, diseñada para hacer que el aprendizaje de la programación sea más entretenido y educativo para los estudiantes de educación básica media-superior. Se llevaron a cabo entrevistas con estudiantes y docentes para determinar las características del juego. Además, se crearon prototipos que se probaron con usuarios representativos utilizando un enfoque de diseño participativo. Finalmente, se llevó a cabo una fase de pruebas para evaluar la usabilidad de la aplicación. Los resultados sugieren que la aplicación tiene el potencial de aumentar la motivación al aprendizaje de la programación, lo que sienta una base sólida para futuras investigaciones y avances en el campo de la educación tecnológica.</p>			
<b>Abstract:</b>	<p>Programming has become an essential skill that not only promotes technical careers but also enhances logical thinking and problem-solving skills in various fields. Despite its importance, teaching programming to new learners has its challenges, especially when it comes to maintaining their interest in the concepts being explained. To address these issues, a web application in the form of a serious game is proposed, designed to make learning programming more entertaining and educational for upper-middle basic education students. Interviews were conducted with students and teachers to determine the features of the game. In addition, prototypes were created and tested with representative users using a participatory design approach. Finally, a testing phase was carried out to evaluate the usability of the application. The results suggest that the application has the potential to increase motivation for learning programming, laying a solid foundation for future research and advancements in the field of technological education.</p>			
<b>Descripción:</b>	101 hojas: Dimensiones 21 cm x 29,7 cm + CD-ROM 6162			
<b>URI:</b>				

## INTRODUCCIÓN

Se espera que la demanda de profesionales con habilidades de programación aumente significativamente en un futuro próximo [1] lo que ha traído implicaciones. Se ha vuelto crucial abordar el aprendizaje de programación desde temprana edad, brindando a los estudiantes de educación básica la oportunidad de desarrollar habilidades digitales y de pensamiento computacional que les permita enfrentar los desafíos de la era digital de manera efectiva y aprovechar las oportunidades que ofrece. Así, las escuelas pueden sacar provecho del interés natural de los jóvenes por la tecnología para promover el aprendizaje de la programación, fomentando el pensamiento lógico y la creatividad [2].

Además, la digitalización está transformando rápidamente el mercado laboral, la vida y la sociedad en general. Durante la última década, se ha prestado cada vez más atención a la competencia digital de los niños, con el fin de prepararse para un futuro mercado laboral que será digitalizado y automatizado. Asimismo, se ha reconocido la importancia del conocimiento digital profundo para entender la sociedad del siglo XXI. Como resultado, el pensamiento computacional (CT, por sus siglas en inglés) se ha introducido en la educación, con el objetivo de brindar a los niños una comprensión básica de los algoritmos, la descomposición de problemas y el reconocimiento de patrones [3].

Por estas razones, en años recientes se ha venido realizando diversas acciones para trabajar en el Desarrollo del CT. Por ejemplo, Bell et al. [4] realizaron un campamento de programación diseñado para estudiantes de sexto a noveno grado. Durante el campamento los participantes utilizaron Scratch, e incorporaron elementos de arte y música en las actividades de programación. Las encuestas realizadas antes y después del campamento mostraron un aumento significativo en la autoeficacia y el interés en la programación. Esto sugiere que la integración de la programación y las artes creativas en la educación puede ser efectiva para involucrar y promover el interés en las ciencias de la computación. Por otra parte, el estudio realizado por Yadav et al. [5], aborda la importancia de la enseñanza del pensamiento computacional desde una edad temprana y proporciona una visión general de los desafíos y oportunidades que enfrentan los educadores al enseñar el pensamiento computacional. Además, el artículo destaca la importancia de la colaboración entre educadores, investigadores y la industria para desarrollar y mejorar la enseñanza del pensamiento computacional.

Por otro lado, aprender a programar requiere muchas competencias, como el razonamiento lógico, la solución de problemas y la comprensión de ideas abstractas. Es por esta causa que

a numerosos alumnos les resulta un desafío aprender a programar. Este aspecto puede disminuir su motivación al cursar materias introductorias de programación [6]. Además, existen otros obstáculos que deben superarse para aprovechar plenamente los beneficios educativos que brinda la programación. Uno de estos obstáculos es el diseño de actividades de programación desconectadas de la realidad y contexto de vida de los estudiantes, lo cual puede desmotivarlos y alejarlos del aprendizaje [7], [8]. Estos inconvenientes, junto con las técnicas de enseñanza utilizadas en décadas pasadas, han ocultado el potencial educativo que puede ofrecer la programación en las primeras etapas de la educación [9].

En este contexto, este proyecto de investigación plantea una aplicación web enfocada en un juego educativo de tipo Escape Room o Salas de Escape, con el propósito de fomentar el aprendizaje de los principios básicos de la programación en el nivel de educación básica media-superior. Esta herramienta tiene como objetivo promover la creatividad y el pensamiento lógico presentando a los estudiantes acertijos relacionados con la programación. Mediante la resolución de problemas a través de un juego de Salas de Escape, los estudiantes podrán practicar sus habilidades de codificación y desarrollar su capacidad para lidiar con situaciones complejas. De esta forma, se busca contribuir al proceso de enseñanza y aprendizaje de programación al incentivar el uso de lenguajes de programación entre estudiantes de educación básica media-superior.

## **CAPÍTULO I**

### **CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN**

## **1.1. Problema de investigación**

### **1.1.1. Planteamiento del problema**

Es innegable la creciente importancia de la programación. La sociedad se encuentra inmersa en una era digital donde las Tecnologías de la Información y Comunicación (TIC) desempeñan un papel fundamental en todos los ámbitos de nuestra vida. Como parte de ello, la programación es ampliamente reconocida como una habilidad esencial no solo en ingeniería, sino también en otros campos [1]. Sin embargo, actualmente no se limita solo a los adultos, sino que es crucial abordar este tema desde una edad temprana. Aunque la programación puede parecer compleja y desconocida para muchas personas, especialmente para aquellas en edades de educación básica que aún no tienen conocimientos en este ámbito, es fundamental iniciar el desarrollo del pensamiento computacional desde temprana edad porque esto puede tener un impacto significativo en el desarrollo de habilidades como el razonamiento lógico, la resolución de problemas y la creatividad [10].

A pesar del progreso en la integración de la tecnología en las escuelas, la enseñanza de la codificación no se ha generalizado, lo que pone a los estudiantes en desventaja en términos de habilidades digitales y limita sus oportunidades futuras en un mercado laboral en constante crecimiento, el cual está cada vez más enfocado en habilidades tecnológicas [1]. Es imprescindible abordar esta brecha y promover la enseñanza de programación desde temprana edad para garantizar que los estudiantes estén preparados para el futuro. Esto implica una atención especial a la inclusión de la codificación desde la escuela primaria, así como a la provisión de recursos adecuados y accesibles para los estudiantes. Al cerrar esta brecha y priorizar la enseñanza de programación, se puede equipar a los estudiantes con las habilidades necesarias para prosperar en una sociedad cada vez más digitalizada y competitiva.

Sin embargo, enseñar lenguajes de programación a niños de educación básica puede enfrentar desafíos considerables [6], [7], [11]. Los métodos de enseñanza inadecuados, la escasa interacción con los estudiantes y la falta de interés son los principales obstáculos para aprender a programar [12]. Para muchos estudiantes, aprender a codificar puede parecer algo abstracto, alejado de sus intereses y actividades cotidianas, por ello pueden percibir la programación como aburrida o poco interesante, lo que puede conducir a una falta de motivación y compromiso.

## **Diagnóstico**

La enseñanza de lenguajes de programación a estudiantes de primaria puede enfrentar importantes desafíos [6], [7], [11]. Para los estudiantes, el proceso de aprender habilidades de programación a menudo se percibe como una tarea difícil y frustrante, especialmente cuando los estudiantes tienden a perderse en la codificación, centrarse demasiado en los detalles y perder el enfoque en los conceptos o conceptos más amplios de la programación. cuando se eliminan del diseño general [13]. Los métodos de enseñanza incorrectos, la mala interacción con los estudiantes y la falta de interés son los mayores obstáculos para aprender a programar [12].

## **Pronóstico**

Utilizar juegos como herramientas educativas podría ser una estrategia efectiva para combatir el desinterés de los niños por aprender a programar. Se ha demostrado la eficacia del uso de aplicaciones de juegos en cualquier ciencia, especialmente en informática [14]. La incorporación de conceptos de programación en los videojuegos ha ganado popularidad en la industria del entretenimiento digital. Aunque se han creado numerosos juegos con fines educativos para enseñar programación, su diversidad y alineación con los planes de estudio aún permanecen en un territorio inexplorado [1]. El uso de salas de escape representa un concepto novedoso. A pesar de su capacidad para abrir oportunidades innovadoras en el ámbito educativo y transformar la forma de aprendizaje, aún existe una escasez de investigaciones que analicen sus tendencias, posibilidades y desafíos en la educación [15], y en particular en el aprendizaje de programación.

### **1.1.2. Formulación de problema**

¿Cómo debería estar diseñada una aplicación web que aporte al incentivo del aprendizaje de fundamentos de programación de estudiantes de educación básica media-superior?

### **1.1.3. Sistematización del problema**

¿Qué lineamientos debería tener la aplicación para que constituya un aporte al aprendizaje de fundamentos de programación?

¿Cómo debería ser construida una aplicación web que permita incentivar el aprendizaje de fundamentos de programación?

¿Qué acogida tiene la aplicación por parte de usuarios representativos?

## **1.2. Objetivos**

### **1.2.1. Objetivo General**

Diseñar una aplicación web basada en la temática de Salas de Escape que permita incentivar el aprendizaje de fundamentos de programación de estudiantes de educación básica media-superior.

### **1.2.2. Objetivos Específicos**

- Seleccionar las características que se deben incorporar en un software como el propuesto a través de entrevistas a profesionales relacionados al tema y con la participación de usuarios representativos.
- Diseñar un prototipo de aplicación web basado en las características previamente definidas con miras a facilitar el aprendizaje de fundamentos de programación.
- Determinar el grado de usabilidad de la aplicación mediante la colaboración de usuarios representativos, con el fin de obtener retroalimentación y realizar mejoras en el prototipo.

## **1.3. Justificación**

La programación y el pensamiento computacional son esenciales en la sociedad actual, ya que la tecnología está presente en todas las áreas de nuestras vidas. Estas habilidades permiten adaptarse a los avances tecnológicos y fomentan la innovación y la creatividad, siendo cada vez más demandadas en el mercado laboral [16]. Además, es importante enfocarse en garantizar una educación inclusiva, equitativa y de calidad, promoviendo oportunidades de aprendizaje a lo largo de toda la vida para todos los ciudadanos, tal como lo señala el Cuarto Objetivo de Desarrollo Sostenible [17]. Con este propósito, se ha propuesto el desarrollo de un software basado en tecnología web y Salas de Escape para incentivar el aprendizaje de fundamentos de programación entre estudiantes de educación básica-superior. Se reconoce la importancia de preparar a los estudiantes para el futuro, dotándolos de habilidades de alfabetización digital y fomentando el uso eficiente de la tecnología para alcanzar el éxito profesional [18]. Ya que aprender a codificar permite comprender cómo funciona el mundo digital y desarrollar soluciones innovadoras a los desafíos que se enfrentan diariamente, estas son habilidades necesarias en cualquier entorno laboral, especialmente en el ámbito digital [19].

La programación también se ha convertido en un tema destacado en la industria de los videojuegos [1]. El uso de juegos con propósitos educativos y de aprendizaje no es una

novedad, pero el aumento en la popularidad de los juegos digitales como forma de entretenimiento convencional ha generado el interés en cómo aprovechar su potencial para la educación [1], [20]. Los juegos abarcan una amplia variedad de campos (humanidades, ciencias, ingeniería, etc.) y contenidos (aprendizaje de idiomas, ciencia, historia, etc.), así como diferentes géneros de juego (casuales, disparos en primera persona, juegos en línea multijugador masivos [MMO], juegos de rol, etc.). Estos géneros se entrelazan y se conectan entre sí en diversas formas [20]. Además, distintos géneros de juegos respaldan diversos propósitos de aprendizaje. Por ejemplo, los juegos de rompecabezas son ideales para practicar problemas específicos de forma estructurada, lo que a su vez puede utilizarse para mejorar habilidades particulares. Estos juegos ofrecen un entorno controlado con niveles que se asemejan a ejercicios individuales [1]. Los niños de hoy se enfrentarán a un flujo continuo de nuevos problemas y desafíos inesperados en el futuro. Muchas cosas que aprenden hoy serán obsoletas mañana. Para prosperar, deben aprender a diseñar soluciones innovadoras a los problemas inesperados que sin duda surgirán en sus vidas [21].

En este contexto, surge la propuesta de utilizar las Salas de Escape como una herramienta innovadora para enseñar programación, ya que representan una alternativa pedagógica valiosa. En consonancia con la transición reciente de los enfoques educativos tradicionales en el aula hacia métodos de aprendizaje activo, se ha observado un crecimiento significativo en el uso de Salas de Escape y juegos grupales en el ámbito académico [15]. Las salas de escape son tecnologías educativas que inspiran el aprendizaje y la participación en desafíos. En estas estrategias, los estudiantes trabajan para usar su razonamiento, conocimiento y habilidades para resolver acertijos y desafíos relacionados con el contenido del plan de estudios, con el objetivo de completar el juego dentro del tiempo asignado [22].

Con este proyecto se espera cumplir con el objetivo principal de incentivar el aprendizaje de los fundamentos de programación entre estudiantes de educación básica-superior. El software ha sido diseñado de manera interactiva, amigable y atractiva, proporcionando una experiencia educativa lúdica y envolvente que fomenta el interés y la participación de los jóvenes en el mundo de la programación.

## **CAPÍTULO II**

### **FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN**

## **2.1. Programación y pensamiento computacional**

Un lenguaje de programación es un conjunto de reglas y estructuras sintácticas que permiten a los programadores escribir instrucciones que permiten a las computadoras comprender y realizar tareas específicas. Estos lenguajes proporcionan un conjunto de palabras clave, reglas gramaticales y convenciones de escritura que definen cómo deben construirse los programas. Un programa consiste en un conjunto de instrucciones destinadas a ser ejecutadas por una máquina informática con el fin de realizar tareas específicas y lograr un objetivo preestablecido.

Hay dos tipos de lenguajes de programación: lenguajes específicos de dominio (DSL) y lenguajes de propósito general (GPL) [23]. Los DSL se enfocan en resolver problemas particulares en un dominio específico, como bases de datos o aplicaciones web. Algunos ejemplos de estos lenguajes son el lenguaje de marcado de hipertexto (HTML), las hojas de estilo en cascada (CSS) y el lenguaje de consulta estructurado (SQL) [24]. Los lenguajes de propósito general (GPL) se utilizan para desarrollar software en varias áreas de aplicación. Ejemplos generales de lenguaje de proposiciones para C, C++, C#, Python y Java [25].

El pensamiento computacional es un medio para comprender y resolver problemas complejos mediante el uso de conceptos y técnicas informáticas, como la descomposición, el reconocimiento de patrones, la abstracción y los algoritmos [26]. Dado que puede relacionarse con las habilidades de los individuos para usar la tecnología en la vida cotidiana, el pensamiento computacional se vuelve aún más crítico para tener éxito en la sociedad [27].

### **2.1.1. Programación en educación básica**

El pensamiento computacional es una habilidad esencial para resolver de manera efectiva los desafíos de una sociedad compleja y tecnológica. Dada la falta de oportunidades en las escuelas para desarrollar estas habilidades, es necesario apoyar a los docentes para que las integren en sus métodos de enseñanza. Además de adquirir conocimientos tecnológicos, contenidos y técnicas pedagógicas, los educadores deben reconocer la importancia del pensamiento computacional en su enseñanza, ya que influirá en su práctica futura [27].

La programación basada en bloques se está volviendo cada vez más popular entre los jóvenes estudiantes que desean ingresar al campo de la informática [28]. Esta forma de programación se usa a menudo en el contexto de micromundos, donde los estudiantes resuelven acertijos y desafíos específicos. Herramientas como Scratch, Blockly y Alice han impulsado esta tendencia al permitir que millones de niños participen en la codificación arrastrando y soltando bloques a través de interfaces gráficas. Un ejemplo de su aplicación

está en el evento denominado “La Hora del Código”, que está dirigido a millones de estudiantes [29]. Esta iniciativa ha superado los 500 millones de "horas servidas" y ha llegado a estudiantes de todos los países del mundo [30]. Esto demuestra que la moda de la informática es un fenómeno global.

Scratch [31] es uno de los lenguajes de programación basados en bloques más populares. A partir de septiembre de 2022, había más de 94 millones de usuarios de Scratch, con más de 110 millones de proyectos Scratch compartidos entre ellos [32]. Debido al éxito de estas herramientas e iniciativas para involucrar a los principiantes en la programación, los entornos de programación basados en bloques se integran cada vez más en los planes de estudio de los cursos de informática de la escuela secundaria [33]. Los lenguajes de programación que utilizan bloques son muy empleados en la educación, tanto en escuelas primarias y secundarias como en instituciones de educación superior. El objetivo principal de estos lenguajes es promover habilidades de pensamiento computacional en los estudiantes [34].

Yendo un poco más allá, Zhonggen et al. [35] determinaron que los juegos educativos pueden mejorar significativamente los resultados del aprendizaje en varias áreas, incluido el rendimiento académico, las habilidades de resolución de problemas y el pensamiento crítico. Además, los juegos educativos pueden aumentar la motivación y el compromiso de los estudiantes, aunque los resultados son variables y dependen en gran medida del diseño del juego y del contexto de implementación.

De manera similar, Lindberg et al. [36] realizaron investigaciones en siete países y descubrieron que la integración de la programación en los planes de estudio escolares es una tendencia creciente. Este estudio sugiere que los juegos de programación, cuando se integran adecuadamente con los objetivos del curso, pueden ser una herramienta valiosa para apoyar el aprendizaje de conceptos de programación en diferentes niveles educativos.

También, Zhonggen [37] realizó una revisión sistemática de la literatura de la última década sobre el uso de juegos serios en educación mostrando que pueden tener un impacto significativo en el aprendizaje. El análisis muestra que los juegos serios son particularmente efectivos para mejorar la retención de conocimientos y la motivación de los estudiantes. Además, factores como la usabilidad del juego y la interacción social entre los estudiantes afectan significativamente la efectividad del aprendizaje basado en juegos serios.

### **2.1.2. Juegos serios y programación**

Se utilizan varios términos para describir la inclusión de juegos en la educación, tales como: juegos serios, entretenimiento educativo, gamificación, aprendizaje basado en juegos y aprendizaje lúdico [1], [20]. Ha habido debates considerables sobre cómo incorporar contenido educativo en los juegos educativos, lo que ha dado como resultado una variedad de enfoques y perspectivas. Por ejemplo, los juegos de conversación educativos generalmente brindan herramientas de aprendizaje donde los juegos se usan para lograr objetivos educativos específicos, mientras que en el aprendizaje lúdico los juegos son visibles e inherentemente parte del plan de estudios [38].

Los juegos serios son una valiosa herramienta que ha demostrado ser sumamente atractiva tanto en el ámbito educativo como en el de la formación, pero su utilidad es aún más amplia [39]. Mientras que los juegos en general suelen ser diseñados con el único propósito de brindar entretenimiento, los juegos serios van más allá al proporcionar un entorno atractivo que fomenta la adquisición de conocimientos, estimula el interés, la motivación y la mejora de los logros de aprendizaje de los estudiantes [40].

Los juegos serios se caracterizan por su enfoque en la resolución de problemas y en la integración de componentes de aprendizaje [41], y están diseñados para que los estudiantes aprendan y practiquen diversas habilidades, utilizando el juego como medio para alcanzar un propósito educativo específico [42]. De esta manera, los jugadores se enfrentan a desafíos interactivos que los instan a esforzarse por transformar al alumno de un receptor pasivo de información a un colaborador activo en el proceso educativo [43].

En una aplicación práctica de estos principios, Montes et al. [44] analizaron el uso de un juego serio en línea para enseñar conceptos básicos de programación y promover experiencias de juego entre los estudiantes de secundaria. Los resultados mostraron mejoras significativas en el rendimiento académico en el grupo que utilizó juegos serios, sin diferencias de género. Además, las percepciones de la experiencia de juego, evaluadas mediante el cuestionario Gamefulquest, fueron positivas, siendo ligeramente superiores para las mujeres que para los hombres.

## **2.2. Salas de escape educativas**

Las salas de escape virtuales educativas son un tipo de juegos serios que tienen como objetivo armonizar la diversión con el aprendizaje y el entretenimiento con la educación [45]. Se caracterizan como experiencias de equipo donde los participantes buscan pistas, resuelven acertijos y completan desafíos dentro de una o varias salas con el propósito de

alcanzar un objetivo concreto, que generalmente consiste en escapar de la habitación, todo ello dentro de un límite de tiempo establecido [46]. Esta mezcla bien integrada de contenido dentro de una narrativa o historia transforma el proceso de aprendizaje en una experiencia atractiva que fomenta el sentido de control y dominio [47].

Sonsoles et al. [48] aplicaron analítica de aprendizaje para investigar el comportamiento de los estudiantes en una sala de escape educativa centrada en la programación. Observaron que el uso efectivo de pistas estaba correlacionado con una mejora en el aprendizaje. Los resultados mostraron que la actividad de la sala de escape incrementó significativamente el conocimiento de los estudiantes, demostrando la efectividad de las salas de escape como herramientas educativas.

En un estudio relacionado, Lathwesen y Belova [49] analizaron 93 artículos para determinar si el mercado para estos juegos educativos está saturado o si aún existe potencial para su desarrollo. La revisión reveló una amplia aplicación en la formación médica, especialmente en enfermería, y una necesidad de salas de escape más adaptables y de mayor evidencia empírica sobre sus efectos en el aprendizaje. Los resultados destacan la continua relevancia de las salas de escape en educación STEM, pero también la importancia de investigar más para optimizar su diseño y validar su eficacia educativa.

Además, Makri et al. [50] realizaron una revisión sistemática de la literatura sobre salas de escape educativas digitales. Examinan cómo estos entornos digitales fomentan resultados de aprendizaje en entornos virtuales y destacan su eficacia para mejorar habilidades cognitivas, como el pensamiento crítico y la motivación. El estudio propone una agenda para futuras investigaciones, resaltando la necesidad de optimizar el diseño y la aplicación de estas herramientas pedagógicas.

Finalmente, dentro del contexto del proyecto de investigación, se optó por la denominación "Salas de Escape" en lugar de "Cuartos de Escape" se justifica debido a su mayor nivel de formalidad. El término "Salas de Escape" evoca una sensación de intriga y misterio, insinuando que los participantes se enfrentarán a desafíos más elaborados y cautivadores. La palabra "sala" sugiere un espacio más amplio y completo, lo que implica que la experiencia será más detallada y compleja. En contraste, el término "cuarto" conlleva la idea de un espacio más pequeño y limitado, lo que podría llevar a percibir la experiencia como menos desafiante o menos cuidadosamente diseñada.

### **2.3. Diseño participativo**

El diseño participativo es una metodología interdisciplinaria centrada en el ser humano que se utiliza para desarrollar productos TIC que buscan satisfacer las necesidades de todos los involucrados [51]. Los estudios han demostrado que este enfoque tiene un impacto positivo en la satisfacción de los valores y necesidades de los usuarios [52]. Por lo tanto, se recomienda utilizar el diseño participativo para satisfacer las necesidades de todos los actores involucrados [53].

El diseño participativo involucra a varias partes interesadas, como diseñadores, investigadores, profesores y estudiantes, que trabajan juntos como socios en el proceso de diseño. Debido a la diversidad de actores involucrados, la planificación participativa se caracteriza por la negociación, la mediación y el consenso para la toma de decisiones. Algunos autores han abogado por la planificación participativa para la inclusión de los usuarios finales en la educación por varias razones, como mejorar la cooperación y la equidad entre todos los participantes [54].

En una investigación realizada por Merkel y Kucharski [55] se analiza el uso del diseño participativo en gerontotecnología, clasificando los estudios en tres categorías según las tecnologías empleadas. Se destaca la importancia de considerar el motivo, quiénes, cuándo y cómo se involucran los futuros usuarios en el proceso de innovación. Se sugiere seguir una estrategia integral que inicie con las necesidades del usuario y concluya con una evaluación de los resultados logrados mediante los métodos aplicados.

Por otro lado, Reddy et al. [56] aplican el diseño participativo en el diseño de una intervención de farmacia comunitaria, destacando los pasos, ventajas y desafíos. Se incluye el proceso de seis sesiones adaptativas con personal de farmacia y pacientes mayores como interesados. Las ventajas del diseño participativo incluyen clarificar problemas, generar soluciones innovadoras y ajustar iterativamente las ideas de diseño. Aunque el diseño participativo requiere tiempo y es intensivo, proporciona información valiosa para los investigadores y llevó al desarrollo de un prototipo centrado en el paciente.

### **2.4. Metodología de desarrollo basado en prototipo**

Un prototipo es una versión preliminar o preliminar de un producto, sistema o diseño creado para probar, evaluar y validar conceptos, características y funciones antes de la producción o implementación final. La creación de prototipos es una excelente manera de pensar en una interfaz de usuario (UI) y permite evaluar una solución al comienzo del proyecto [57].

Existen diferentes categorías de prototipos, que se pueden clasificar en baja, media y alta fidelidad [58]. Según Walker [59], los prototipos de "alta fidelidad" son aquellos que más se asemejan al producto final, como los prototipos creados en HTML. Por otro lado, los prototipos de "baja fidelidad" son los menos similares, como prototipos en papel o bocetos. Los prototipos de baja fidelidad son valiosos debido a su simplicidad de producción, bajo costo y velocidad. Ayudan a identificar problemas en las primeras etapas del diseño y, a través del juego de roles, los usuarios pueden tener una idea realista de cómo será interactuar con el producto. Por otro lado, la creación de prototipos de alta fidelidad es útil para presentar ideas a las personas y probar aspectos técnicos o problemas específicos [58].

## **CAPÍTULO III**

### **METODOLOGÍA DE LA INVESTIGACIÓN**

### **3.1. Localización**

El presente proyecto de investigación se lo realizó en la Universidad Técnica Estatal de Quevedo (UTEQ), en el campus Central, ubicado en el Cantón Quevedo de la Provincia Los Ríos, Av. Quito km. 1.5 vía a Santo Domingo de los Tsáchilas.

### **3.2. Tipos de investigación**

Este proyecto se llevó a cabo un enfoque de investigación descriptiva. Dado que se efectuó una revisión detallada de varias fuentes relevantes, incluidos artículos académicos, revistas, libros e investigaciones previas, así como grupos focales, para identificar características específicas.

### **3.3. Métodos de investigación**

#### **3.3.1. Método Inductivo**

El proyecto sigue un enfoque inductivo. En este sentido, se partió de observaciones realizadas a través de entrevistas no estructuradas y de la retroalimentación de los usuarios durante el proceso de diseño participativo. A partir de este material empírico, se formularon generalizaciones y teorías, como por ejemplo mejoras en la usabilidad o en el diseño de la interfaz de usuario. De esta manera, se ha establecido una base sólida para optimizar el diseño del software.

#### **3.3.2. Método Analítico**

Dada la variedad de fuentes de información utilizadas (artículos científicos, publicaciones en revistas, tesis sobre temas relacionados) y la complejidad inherente al desarrollo y evaluación de software, este enfoque permitió descomponer y examinar detalladamente los datos recopilados, incluyendo tanto la información obtenida de ChatGPT como los datos recogidos durante el proceso de prototipado y la evaluación de la usabilidad.

### **3.4. Fuentes de recopilación de información**

Se utilizaron varias fuentes de información para respaldar el proyecto de investigación, incluidos artículos científicos, publicaciones de revistas, tesis sobre temas relacionados y entrevistas que contribuyeron significativamente a la investigación.

### **3.5. Diseño de la investigación**

#### **3.5.1. Obtención de requerimientos**

Se consideraron tres pasos para obtener los requisitos previos necesarios. En primer lugar, se llevó a cabo entrevistas no estructuradas. En segundo lugar, se utilizó inteligencia artificial generativa para complementar los requisitos. Finalmente, se siguió un enfoque de diseño participativo. A continuación, se describen estos pasos.

- **Entrevista no estructurada**

Se realizaron entrevistas no estructuradas con el objetivo de obtener las ideas de los participantes sobre la propuesta de utilizar juegos de escape como medio de aprendizaje de un lenguaje de programación. Si los entrevistados tenían experiencia con juegos similares centrados en el aprendizaje a través de salas de escape, se les pedía que dieran su opinión sobre ellos. Las respuestas de los participantes ayudaron a formular preguntas de seguimiento para explorar sus puntos de vista en detalle y comprender de mejor manera la viabilidad y el potencial de la propuesta.

Como proceso de selección, se tomó en consideración a aquellos participantes que eran ingenieros especializados en la rama de sistemas y docencia, características clave cuyo aporte fue esencial para explorar y analizar aspectos relevantes relacionados con la tecnología y la educación. Los participantes incluyeron a 4 ingenieras y 3 niños de entre 11 y 12 años con conocimiento previo en aplicaciones similares como Scratch.

La entrevista se llevó a cabo de manera remota a través de una videoconferencia en Google Meet. Durante la sesión, se inició con una breve presentación acerca de qué son las salas de escape, explicando su concepto y su creciente popularidad como actividad de entretenimiento y desafío intelectual. Posteriormente, se proporcionó un ejemplo básico de ejercicios que se suelen encontrar en estas salas, con el fin de ilustrar cómo funcionan y cómo se resuelven los enigmas y desafíos planteados en ellas.

Durante la presentación del ejemplo, se fomentó la participación de los entrevistados, invitándoles a plantear preguntas y expresar sus opiniones acerca de la propuesta. Se generó un diálogo enriquecedor en el que se discutieron las posibles estrategias para resolver los acertijos y se recogieron valiosas perspectivas sobre la experiencia de participar en una sala de escape.

- **Uso inteligencia artificial generativa**

Se usó ChatGPT como una herramienta para obtener ideas sobre cómo se considera que debe ser un juego de Salas de Escape que incentive el aprendizaje de un lenguaje de programación. En colaboración con ChatGPT [60], se le pidió a este modelo de lenguaje que genere sugerencias creativas para el diseño de dicho juego. Se exploraron varios aspectos, como temas, desafíos, mecánicas de juego y actividades relacionadas con la programación, que se pueden incorporar a la experiencia. Este enfoque puede generar nuevas ideas que inspiren juegos de escape atractivos para aprender lenguajes de programación.

No se consideraron alternativas como Bard de Google o Chat Bind debido a razones fundamentales. Al momento de este estudio, Bard se encontraba en una etapa experimental [61], lo que implica que sus respuestas pueden carecer de precisión y confiabilidad. Por otro lado, Chat Bind se centra en mostrar resúmenes de respuestas obtenidas de la web [62], lo que a menudo resulta en respuestas genéricas. En contraste, ChatGPT utiliza su modelo GPT-3.5 optimizado [60], respaldado por el aprendizaje reforzado con retroalimentación humana, lo que lo convierte en la elección ideal para aportar ideas al proyecto.

- **Enfoque de diseño participativo**

El proyecto se inclinó por un diseño participativo desde el principio, convirtiéndolo en una parte central del desarrollo de software. Se comenzó con un prototipo no funcional, que era la idea inicial del contenido del software, este abrió el camino para discusiones e intercambio de ideas con los usuarios, permitiendo recopilar comentarios valiosos y adaptar el diseño a sus necesidades y preferencias. Durante el proyecto, se implementó el primer nivel del software y se realizaron pruebas con un usuario representativo. Este ciclo de desarrollo y retroalimentación se repite con cada nuevo nivel, asegurando que cada nivel sea evaluado. Este enfoque permite un desarrollo que pone a los usuarios al frente, esperando como resultado un producto que no solo cumple con las preferencias de los usuarios, sino que también se ajusta a sus necesidades.

Los participantes en este estudio fueron niños de 12 y 14 años, todos varones. Estos niños fueron seleccionados por su familiaridad previa con tecnologías digitales y juegos interactivos. En cada iteración del juego o al agregar un nuevo nivel, se enviaba una versión actualizada a estos niños. Se les solicitaba que jugaran y, posteriormente, proporcionaran comentarios sobre su experiencia. Estos comentarios eran fundamentales para guiar las siguientes etapas del desarrollo del software. El feedback de los participantes ayudaba a identificar tanto los aspectos positivos del juego como las áreas de mejora, asegurando que cada iteración del software se alineara mejor con las necesidades de los usuarios finales.

### **3.5.2. Diseño de un prototipo**

Durante esta fase, se aplicó la metodología de prototipado, lo que permitió llegar a la versión final de la aplicación tras iterar a través de dos prototipos. El prototipo se construyó siguiendo los lineamientos de diseño establecidos en la etapa previa. El proceso de diseño y desarrollo implicó la creación de prototipos iterativos, comenzando con prototipos de fidelidad media y pasando gradualmente a prototipos de alta fidelidad. Esta estrategia permite probar funcionalidades en una etapa temprana, asegurando que el diseño final satisfaga las necesidades del usuario y los objetivos del proyecto.

El modelo de prototipado es un enfoque iterativo en el cual, a lo largo de este proceso, se construyen versiones preliminares del software, se prueban y se refinan hasta alcanzar un prototipo aceptable. Las fases clave de este modelo incluyen la definición de requisitos, creación del prototipo, retroalimentación por parte de los usuarios, y finalmente, la construcción del sistema final [63].

El primer paso en el desarrollo fue la creación de un prototipo de media fidelidad utilizando Genially [64]. Este prototipo permitió una visualización temprana de la estructura y el flujo de interacción del juego, facilitando la comprensión del concepto y la navegación general sin la necesidad de implementar detalles técnicos complejos. Genially se seleccionó por su capacidad para crear rápidamente maquetas interactivas que pueden simular la experiencia del usuario de forma abstracta.

El prototipo de media fidelidad se centró en definir las mecánicas del juego, los elementos de juego principales, y cómo los usuarios interactuarían con estos componentes. La retroalimentación recogida de las pruebas de este prototipo fue invaluable para identificar mejoras y ajustar el diseño antes de proceder a la implementación técnica más detallada.

Con la retroalimentación obtenida del prototipo de media fidelidad, se procedió a desarrollar un prototipo de alta fidelidad. Esta versión fue realizada utilizando HTML5 y el framework Phaser [65], elegido por su robustez y flexibilidad para el desarrollo de juegos web. Este paso representó un avance que permitió incorporar gráficos detallados, animaciones, interacciones complejas y la lógica de juego específica.

El desarrollo del prototipo de alta fidelidad en Phaser facilitó una exploración profunda de la mecánica del juego, incluyendo la gestión de estados, el control de sprites, las animaciones y la implementación de sistemas de físicas para simular interacciones realistas. Este nivel de

detalle hizo posible realizar pruebas de usabilidad más precisas, permitiendo ajustes finos en la interfaz de usuario y la experiencia de juego en general.

### 3.5.1. Evaluación de usabilidad

La evaluación de usabilidad se llevó a cabo en dos etapas y tuvo como objetivo observar la interacción de los niños con la Sala de Escape y recopilar sus opiniones sobre el juego.

#### 3.5.1.1. Preparación de la Evaluación

En el estudio participaron 20 estudiantes para realizar la evaluación de usabilidad para explorar aspectos adicionales como la satisfacción del usuario y la facilidad de aprendizaje. La cantidad de participantes está dentro de lo habitualmente aceptado en el área de estudios. Este grupo de estudiantes se dividió en dos rangos de edades teniendo en cuenta los diferentes niveles de familiaridad con la programación. El grupo A consistió en 7 niños y 3 niñas de entre 9 y 11 años de años, mientras que el grupo B estuvo compuesto por 6 niños y 4 niñas de entre 12 y 14 años. La **Tabla 1** mientras como se estructuraron estos grupos:

**Tabla 1** Estudiantes seleccionados para la evaluación de usabilidad

Grupo A		Grupo B	
Sexo	Edad	Sexo	Edad
M	11	F	14
M	9	F	13
F	11	F	14
F	10	F	12
M	10	M	13
M	9	M	12
M	10	M	14
M	11	M	12
M	11	M	14
F	11	M	13

**FUENTE: AUTOR**

Se prepararon los materiales necesarios para realizar la evaluación, incluido un cuestionario SUS (System Usability Scale) para evaluar la usabilidad del juego, ya que este cuestionario es ideal para obtener una evaluación rápida y efectiva [66]. Además, el cuestionario fue adaptado con base en un estudio previo realizado por Putnam et al. [67], para hacer las

preguntas más accesibles para niños de diferentes edades. Se simplificó el lenguaje y se utilizaron términos familiares para adaptarse al nivel de lectura y comprensión de los niños. También se empleó una escala Likert con emojis, inspirada en la 'escala de sonrisas', para facilitar aún más la comprensión. Además, se efectuó una prueba de conocimientos de programación en Python (función print(), variables y tipos de datos) para conocer el grado de aprendizaje. Por último, se incluyeron preguntas abiertas para recabar opiniones sobre el juego. Adicionalmente, se solicitó y obtuvo la autorización firmada por parte de un representante de cada estudiante antes de iniciar el estudio.

En la **2** se presentan las 10 preguntas realizadas en el cuestionario SUS. Estas preguntas fueron adaptadas para hacerlas más accesibles a los niños de diferentes edades.

**Tabla 2** Preguntas del cuestionario SUS

Nº	Preguntas
1	Me gustaría usar este juego muchas veces.
2	Encontré el juego muy complicado.
3	Creo que el juego es fácil de usar.
4	Necesité la ayuda de un adulto para utilizar este juego.
5	Todas las partes del juego trabajan bien juntas.
6	Existen partes del juego que no funcionaban como esperaba.
7	Otros niños podrían aprender a jugar este juego rápidamente.
8	Me sentí muy confundido/a al usar el juego.
9	Me sentí muy confiado/a utilizando el juego.
10	Creo que necesito aprender muchas cosas antes de poder jugar eficazmente este juego.

**FUENTE:** AUTOR

Además, en la **Figura 1** se muestra la escala Likert con emojis, inspirada en la 'escala de sonrisas', que se empleó para facilitar aún más la comprensión de los estudiantes.

**Figura 1** Escala Likert de emojis



**FUENTE:** AUTOR

Una vez completada la evaluación, se procedió al cálculo de la puntuación del Cuestionario SUS a partir de las respuestas de los participantes. Cada pregunta se ajusta según una fórmula determinada, restando 1 o respondiendo 5 y luego sumando todos los puntos ajustados. Para obtener el resultado final, multiplica la cantidad resultante por 2,5. Una puntuación alta indica buena usabilidad, mientras que una puntuación baja indica áreas de mejora.

Por otro lado, las respuestas a las preguntas abiertas proporcionaron una visión cualitativa de la experiencia de los participantes, buscando identificar problemas recurrentes, preferencias, dificultades y sugerencias sobre cómo mejorar el juego. Además, la última pregunta ayuda a evaluar el interés por el juego como herramienta de aprendizaje.

Por último, la prueba de conocimiento ayudó a identificar los niveles en los que los estudiantes tenían dificultades, permitiendo así determinar qué conceptos de programación son difíciles de entender y requieren mayor explicación.

La combinación de los resultados de la puntuación del cuestionario SUS, las respuestas de la prueba de conocimientos y la retroalimentación con preguntas abiertas permitió identificar áreas específicas que requerían mejoras.

### 3.5.1.2. Realización de la Evaluación

La **Tabla 3** describe el proceso de evaluación y los pasos tomados para garantizar una comprensión de la experiencia del usuario junto con la interacción de los niños con el juego.

**Tabla 3** Pasos para realizar evaluación de usabilidad

<b>Paso</b>	<b>Descripción</b>
<b>Preparación de materiales</b>	Antes de comenzar la evaluación, se prepararon todos los materiales necesarios, incluyendo el juego o la aplicación a evaluar y cualquier otro recurso relevante (como documentos de registro, cuestionarios, etc.).
<b>Introducción</b>	Se programó una reunión individual con cada estudiante de manera presencial, para explicar el propósito de la evaluación. Durante la reunión, conocieron los objetivos generales de la evaluación y se explicó cómo su participación individual contribuiría a ello.

<b>Tareas</b>	Se explicó a los estudiantes que su tarea consistía en iniciar el juego desde el nivel 0 y completar hasta el nivel 2. Con el objetivo de permitirles explorar y familiarizarse con las funcionalidades del juego por su cuenta, se les animó a interactuar libremente con el juego para comprender mejor la interfaz y mecánica sin intervención directa.
<b>Observaciones</b>	Mientras los niños jugaban, se les solicitó que compartieran su pantalla para poder observar directamente sus interacciones en el juego. Se centró la atención en el tiempo que cada uno tardaba en completar un nivel y en evaluar el nivel de dificultad que cada niño experimentaba durante el juego.
<b>Cuestionario SUS</b>	Después de completar el juego, se aplicó el cuestionario SUS para evaluar la percepción general de usabilidad del sistema y obtener una medida cuantitativa de la satisfacción y facilidad de uso percibida por los usuarios.
<b>Prueba de conocimientos</b>	Se realizó una prueba de conocimientos para evaluar el aprendizaje y la comprensión de los niños sobre los temas explicados dentro del juego.
<b>Preguntas abiertas</b>	Se realizaron preguntas abiertas para obtener retroalimentación cualitativa directa de los niños sobre su experiencia, incluyendo preguntas como: <ul style="list-style-type: none"> <li>• ¿Qué te gustó más del juego?</li> <li>• ¿Hubo algo que te resultó confuso o difícil? ¿Qué?</li> <li>• ¿Cambiarías o mejorarías algo en el juego? ¿Qué?</li> <li>• ¿Te gustaría seguir aprendiendo con este juego?</li> </ul>

---

**FUENTE: AUTOR**

### 3.6. Instrumentos de investigación

Los instrumentos utilizados para el proyecto de investigación fueron:

Entrevista individual

Inteligencia artificial ChatGPT

Diseño participativo

### 3.7. Tratamiento de datos

Para el tratamiento de los datos obtenidos en las entrevistas individuales, se optó por clasificarlos en requerimientos funcionales y no funcionales. Además, se analizaron en conjunto las características obtenidas para asegurar una mejor comprensión.

### 3.8. Recursos humanos y materiales

#### 3.8.1. Recursos de hardware

Los recursos de hardware que se utilizaron en el proyecto de investigación se detallan en la siguiente tabla:

**Tabla 4** Requerimientos del hardware

<b>Cantidad</b>	<b>Equipo</b>	<b>Características</b>
1	Computadora de escritorio	➤ AMD Ryzen 5 3400G 3.70 GHz, Memoria RAM 16 GB y 1 TB Disco Solido.
1	Impresora	➤ Epson L4150 Wifi

**FUENTE:** AUTOR

### 3.8.2. Recursos de Software

Los recursos de software que se utilizaron en el proyecto de investigación se detallan en la siguiente tabla:

**Tabla 5** Recursos del software

<b>Tipo</b>	<b>Descripción</b>
Sistema Operativo	➤ Windows 10: licencia propia
Software de trabajo	➤ Visual Studio Code
	➤ GitHub
	➤ Firebase
	➤ ChatGPT
	➤ Canva
Programas de oficina	➤ Paquete de office de Microsoft

**FUENTE:** AUTOR

**CAPÍTULO IV**  
**RESULTADOS Y DISCUSIÓN**

## 4.1. Resultados de la obtención de requerimientos

En esta fase, se completó la tarea de identificar y adquirir los requisitos y pautas de diseño esenciales para el desarrollo de una aplicación web. Para la obtención de estos requisitos, se implementaron dos enfoques distintos.

### 4.1.1. Entrevista no estructurada

Después de realizar entrevistas no estructuradas con los participantes seleccionados, se pudieron identificar las necesidades y expectativas de los usuarios en cuanto al diseño y la funcionalidad del software.

- **Requerimientos funcionales**

**Tabla 6** Requerimientos funcionales

<b>Requerimiento</b>	<b>Descripción</b>
<b>Registro de usuario</b>	Es importante mantener un registro del avance del estudiante en la aplicación.
<b>Contenido educativo</b>	La aplicación contiene una variedad de módulos de aprendizaje sobre fundamentos de programación. Cada módulo proporciona lecciones interactivas que incluyen ejemplos de código, lecturas, ejercicios prácticos, etc.
<b>Evaluación</b>	Los estudiantes pueden realizar cuestionarios y exámenes para medir su conocimiento en programación. Después de completar una evaluación, la aplicación proporciona retroalimentación inmediata para ayudar a los estudiantes a comprender sus errores y aprender de ellos.
<b>Incorporación de historia</b>	Una buena narrativa o una historia podrá ayudar a aumentar el interés por el uso de una aplicación educativa.
<b>Tutorial interactivo</b>	Implementar un tutorial interactivo al comienzo del juego que introduzca a los jugadores en la premisa del juego y les enseñe cómo interactuar con los objetos.
<b>Interacción con elementos</b>	Permitir a los jugadores interactuar con objetos a través de una interfaz de usuario intuitiva.

<b>Registro y seguimiento de progreso</b>	Los estudiantes pueden obtener puntos para alguna meta, de esta manera tendrán un incentivo para seguir usando la aplicación
<b>Sistema de pistas</b>	El sistema de pistas debe proporcionar a los jugadores información relevante para resolver los enigmas y avanzar en el juego.

**FUENTE: AUTOR**

- **Requerimientos no funcionales**

**Tabla 7** Requerimientos no funcionales

<b>Requerimiento</b>	<b>Descripción</b>
<b>Canje de puntos</b>	Con los puntos obtenidos al realizar un nivel, poder canjear elementos especiales.
<b>Corazones o vidas</b>	Contar con un sistema de vidas donde al finalizar todos los corazones deba realizar otras actividades o esperar a que recargue.
<b>Nivelación fácil, medio y difícil</b>	Tener distribución de niveles por habitación, iniciando por los fáciles hacia los de mayor complejidad.
<b>Logros</b>	Obtener logros especiales al realizar ciertas actividades en la aplicación que ser puedan caracterizar como únicas.

**FUENTE: AUTOR**

La **Tabla 8** resalta las características clave que han sido señaladas por los docentes y estudiantes. Estas características se enfocan en hacer que la programación sea divertida y accesible para los niños, incorporando desafíos adecuados a su edad.

**Tabla 8** Características obtenidas de docentes y estudiantes

<b>Características</b>	<b>Docentes</b>	<b>Estudiantes</b>
<b>Objetivos educativos</b>	<ul style="list-style-type: none"> <li>• Promover el trabajo en equipo</li> </ul>	<ul style="list-style-type: none"> <li>• Aprender lenguajes de programación a través de la práctica</li> </ul>

---

<b>Características del juego</b>	<ul style="list-style-type: none"> <li>• Fomentar la comprensión básica de conceptos de programación</li> <li>• Integración de problemas Python en la trama del juego</li> <li>• Evaluación de código simplificado</li> <li>• Variedad de desafíos</li> </ul>	<ul style="list-style-type: none"> <li>• Introducir conceptos de programación de manera lúdica y accesible</li> <li>• Interfaz intuitiva y atractiva</li> <li>• Elementos visuales atractivos y narrativa lúdica</li> <li>• Niveles de dificultad progresivos</li> <li>• Uso de animaciones para explicar conceptos</li> </ul>
<b>Interactividad</b>	<ul style="list-style-type: none"> <li>• Facilitar la participación de los estudiantes</li> <li>• Retroalimentación inmediata</li> </ul>	<ul style="list-style-type: none"> <li>• Incentivar la colaboración entre compañeros</li> <li>• Entorno interactivo para probar y experimentar con el código</li> </ul>
<b>Dificultad</b>	<ul style="list-style-type: none"> <li>• Adaptación de la dificultad para habilidades en desarrollo</li> </ul>	<ul style="list-style-type: none"> <li>• Ofrecer pistas y ayuda cuando sea necesario</li> <li>• Progresión gradual de desafíos con niveles de dificultad apropiados</li> </ul>
<b>Relevancia educativa</b>	<ul style="list-style-type: none"> <li>• Integración de historias y personajes atractivos para niños</li> <li>• Relación de conceptos de programación con experiencias cotidianas</li> </ul>	<ul style="list-style-type: none"> <li>• Enfoque en aventuras y desafíos que capturan la atención</li> </ul>

---

**FUENTE: AUTOR**

#### 4.1.2. Uso de una inteligencia artificial ChatGPT

Se procedió a realizar una serie de preguntas a ChatGPT con el objetivo principal de comprender los aspectos cruciales para tener en cuenta en el desarrollo de una aplicación educativa enfocada en incentivar el aprendizaje de fundamentos de programación incorporando Salas de Escape. Con base en las respuestas obtenidas, se profundizó en áreas críticas, tales como los temas que abordará la sala de escape, los desafíos propuestos, las mecánicas de juego y las actividades vinculadas a la programación.

En la **Tabla 9**, se muestran las características proporcionadas por ChatGPT con nuevas ideas que ayudan a la creación de juegos de Salas de Escape educativos orientados al aprendizaje de lenguajes de programación.

**Tabla 9** Características obtenidas de ChatGPT

<b>Aspectos</b>	<b>Características</b>
<b>Temas</b>	<ul style="list-style-type: none"><li>• Introducción a la Lógica de Programación</li><li>• Variables y Tipos de Datos</li><li>• Estructuras de Control</li><li>• Funciones y Procedimientos</li><li>• Arreglos y Listas</li></ul>
<b>Desafíos</b>	<ul style="list-style-type: none"><li>• Puzles de código</li><li>• Juegos de lógica</li><li>• Simulaciones interactivas</li><li>• Retos de algoritmos</li><li>• Juegos de preguntas y respuestas</li><li>• Programación por bloques</li><li>• Niveles de dificultad</li></ul>
<b>Mecánicas de juego</b>	<ul style="list-style-type: none"><li>• Puzles de programación</li><li>• Simulaciones interactivas</li><li>• Historias interactivas</li><li>• Competencias y desafíos</li><li>• Tutoriales Interactivos</li><li>• Sistema de recompensas</li></ul>

---

<b>Actividades relacionadas con la programación</b>	<ul style="list-style-type: none"><li>• Resolución de problemas simples</li><li>• Simulaciones interactivas</li><li>• Ejercicios de algoritmos</li><li>• Mini lecciones interactivas</li></ul>
---	--

---

**FUENTE: AUTOR**

## **4.2. Diseño del prototipo**

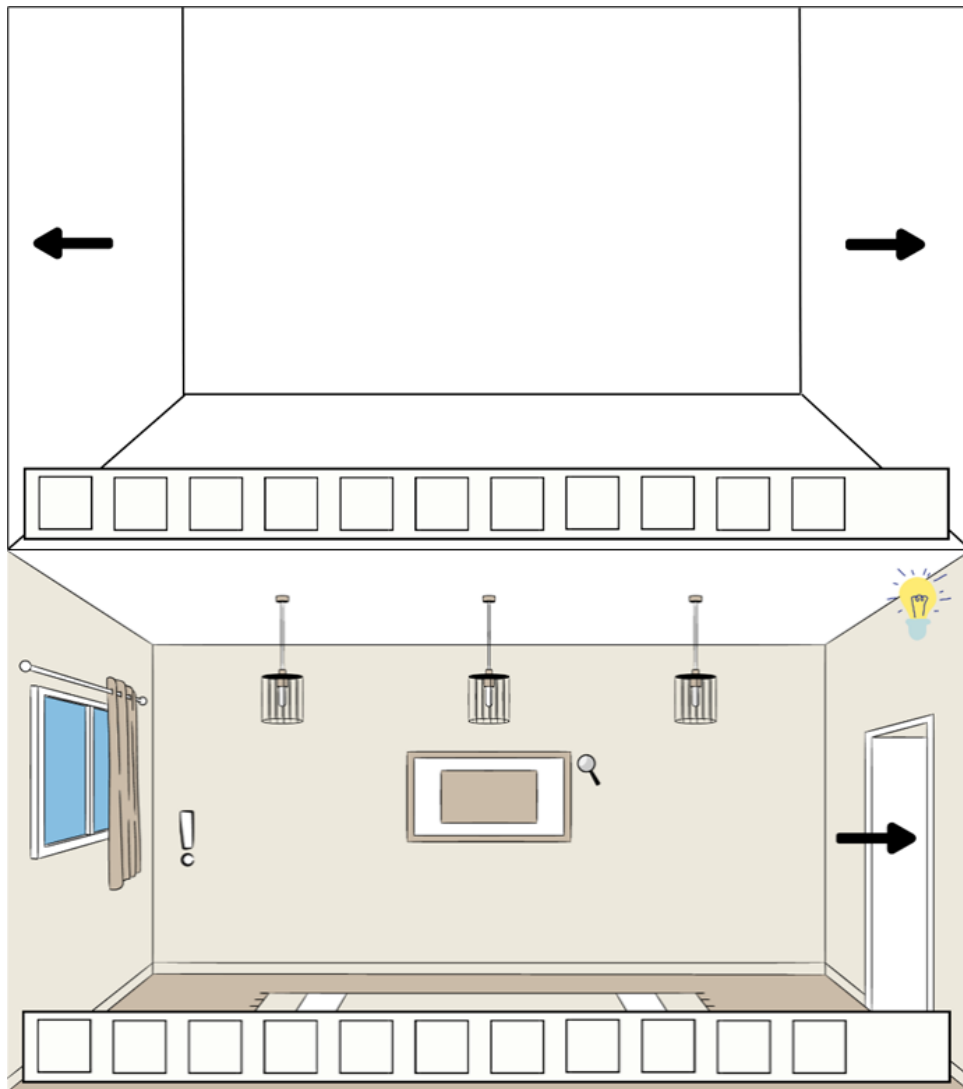
### **Prototipo de baja fidelidad**

Para desarrollar el primer prototipo de juego, se seleccionaron características clave para cumplir con los objetivos educativos y de entretenimiento del proyecto:

- Sala de escape con habitaciones: Diseñado para promover la comprensión básica de conceptos de programación.
- Integración de problemas de Python en escenas del juego: Permite brindar una experiencia de aprendizaje más inclusiva y contextualizada.
- Los desafíos diferentes en cada nivel: Proporciona variedad y profundidad en la resolución de problemas.
- Integración de historias y narrativa en cada nivel: Ayuda a mantener su interés y motivación a lo largo del juego.

Como se muestra en la **Figura 2**, el diseño conceptual de las habitaciones ha sido diseñado con el propósito de introducir y reforzar conceptos específicos de programación mediante puzzles y desafíos integrados. El enfoque del diseño es intuitivo, con el objetivo principal de fomentar la resolución de problemas de programación. Además, en segundo plano, se promueve la búsqueda de pistas y objetos interactivos para añadir un componente adicional de desafío y exploración.

**Figura 2** Diseño inicial de las habitaciones de la sala de escape

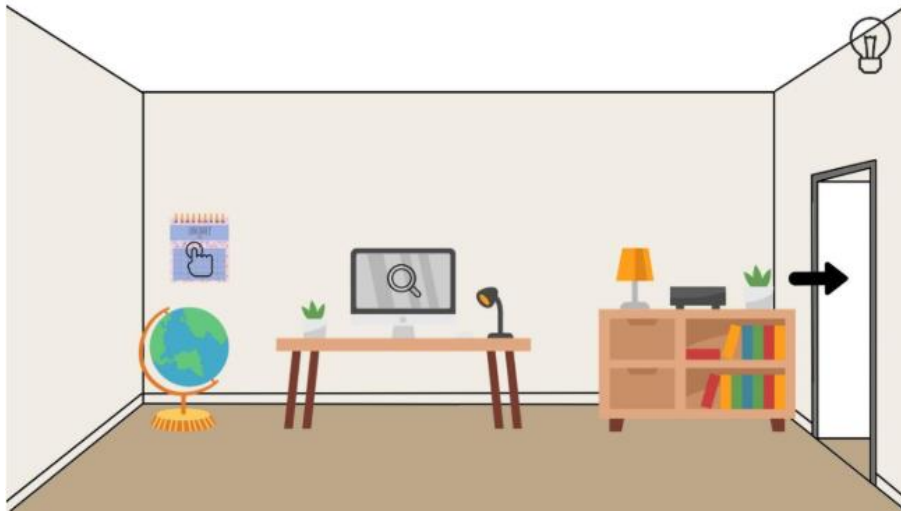


**FUENTE:** AUTOR

### Prototipo de media fidelidad

En el prototipo de media fidelidad se centró en definir las mecánicas del juego, los elementos de juego principales, y cómo los usuarios interactuarían con estos componentes.

**Figura 3** Diseño del prototipo nivel medio



**FUENTE:** AUTOR

Los desafíos se presentan dentro del contexto del juego, lo que hace que el aprendizaje de la programación sea más relevante. Esto permite un enfoque más práctico y atractivo para los estudiantes. Además, se ha diseñado una diversidad de desafíos que se presentan en cada nivel. En la **Figura 4**, se ilustra un ejemplo de cómo los problemas de programación se incorporan en la narrativa del juego.

**Figura 4** Diseño inicial de la integración de Problemas de Programación



**FUENTE:** AUTOR

## Prototipo de alta fidelidad

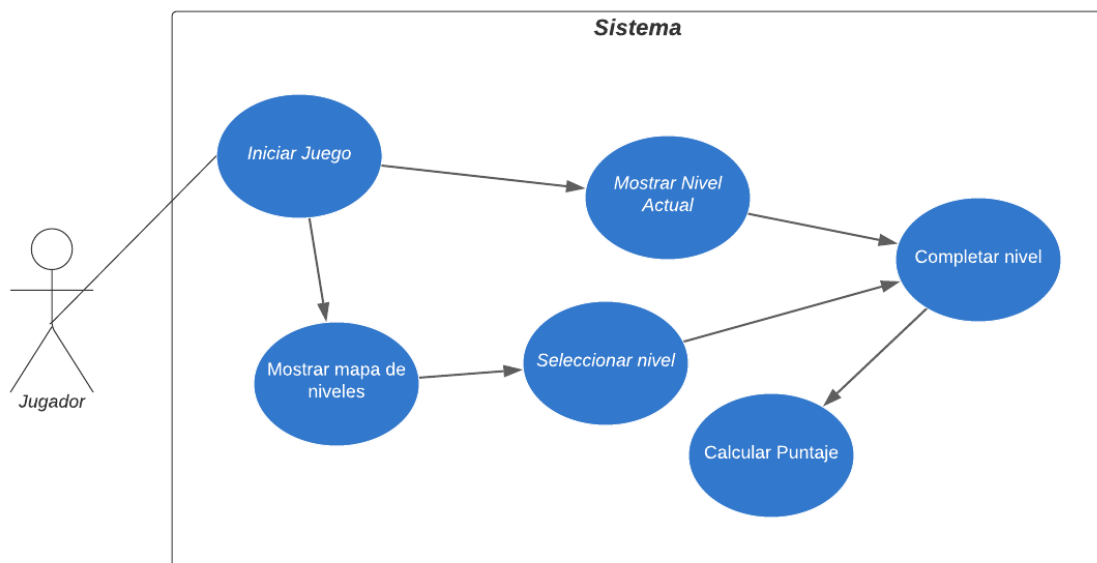
Una vez finalizado el diseño preliminar y aplicados los principios del diseño participativo, se dio inicio al desarrollo de un prototipo de alta fidelidad. Este prototipo fue desarrollado para plataforma web utilizando el framework Phaser, esto permitió implementar la lógica de programación necesaria para modelar escenarios del mundo real y alinearse con los objetivos educativos del proyecto. Este enfoque no sólo promueve una presentación visual más detallada, sino que también permite una evaluación continua del diseño a través de la interacción directa con los usuarios finales.

Se empleó localStorage, para gestionar la persistencia del estado del juego y las puntuaciones de los usuarios por nivel. LocalStorage permite almacenar datos localmente en el dispositivo del usuario. Esto significa que los datos se almacenan directamente en el navegador y están disponibles incluso después de cerrar y volver a abrir el navegador. Esta función es especialmente útil para juegos, ya que permite a los usuarios volver a jugar un nivel y comprobar sus puntuaciones sin depender del almacenamiento externo.

### 4.2.1. Diagrama de casos de uso

En la **Figura 5** se muestra el diagrama de casos de uso general, el cual ilustra las interacciones entre los diferentes usuarios y el sistema.

**Figura 5** Casos de uso del sistema

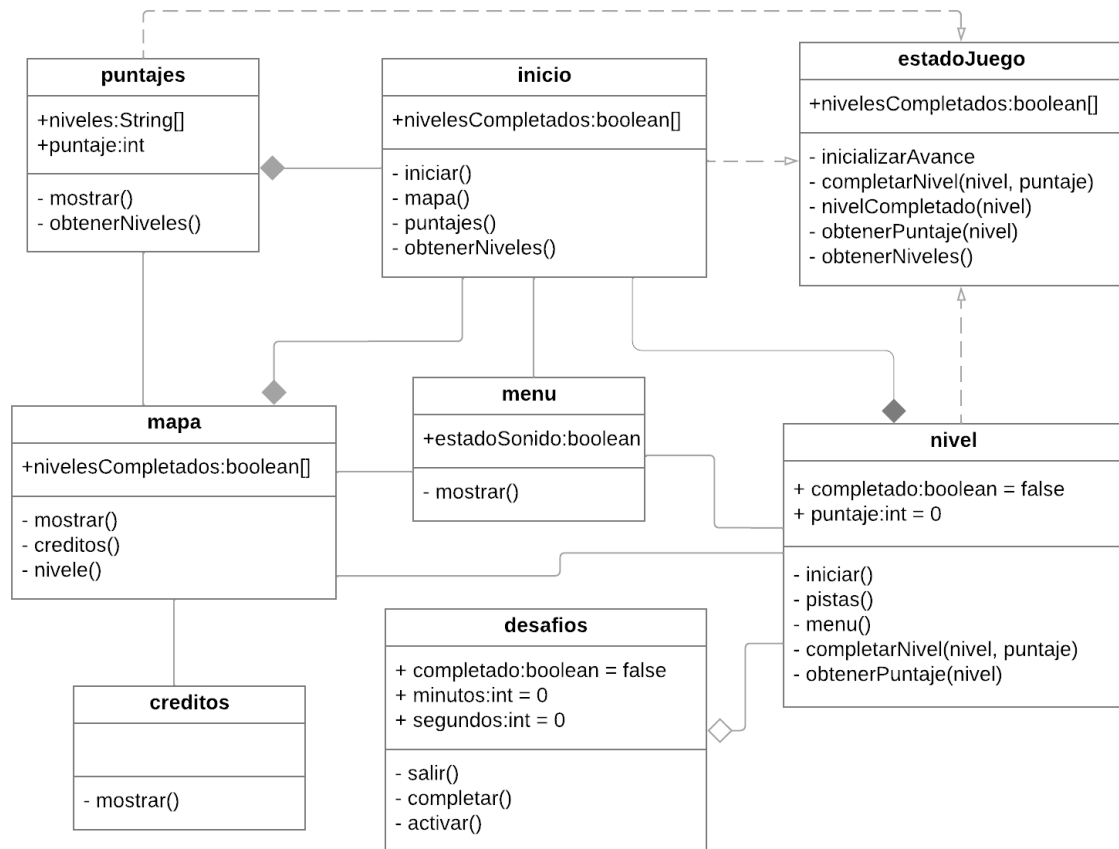


**FUENTE:** AUTOR

#### 4.2.2. Diagrama de clases

La **Figura 6** muestra el diagrama de clases del juego. En este diagrama, se detallan acciones como iniciar el juego, navegar por el mapa, gestionar los niveles y puntajes.

**Figura 6** Diagrama de clases

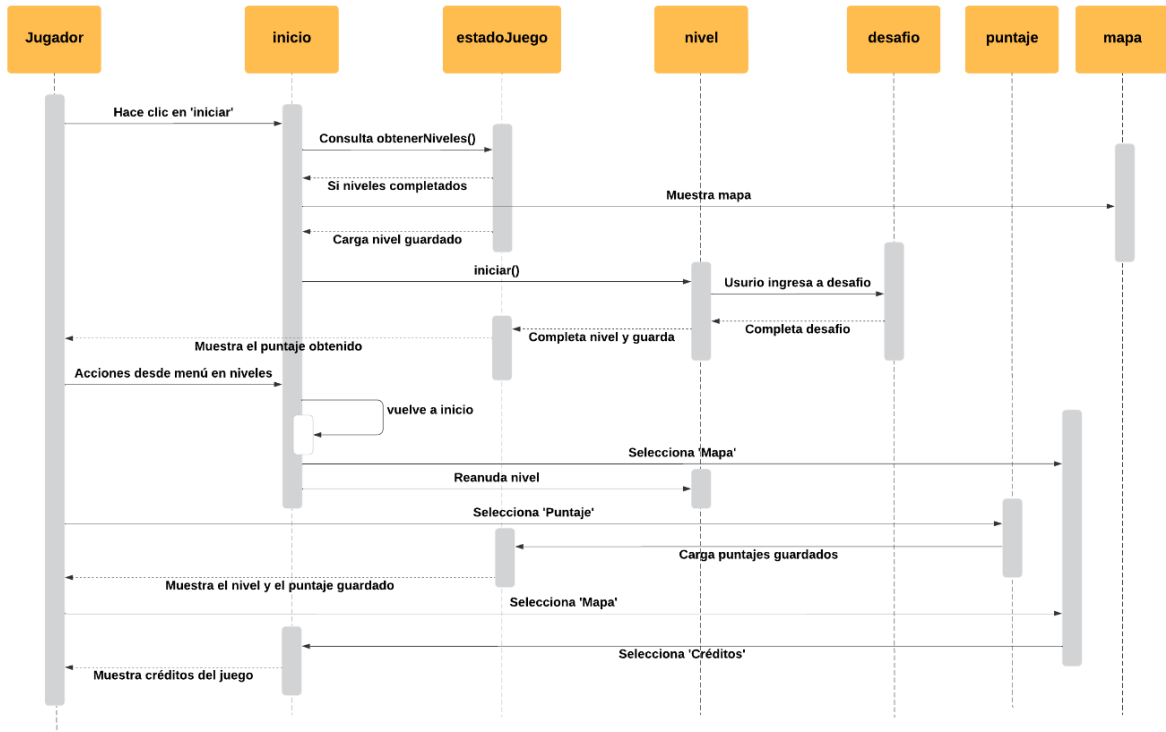


**FUENTE: AUTOR**

### 4.2.3. Diagrama de secuencia general

En la **Figura 7** se muestra el diagrama de secuencia general, que ofrece una visión global entre los componentes del sistema.

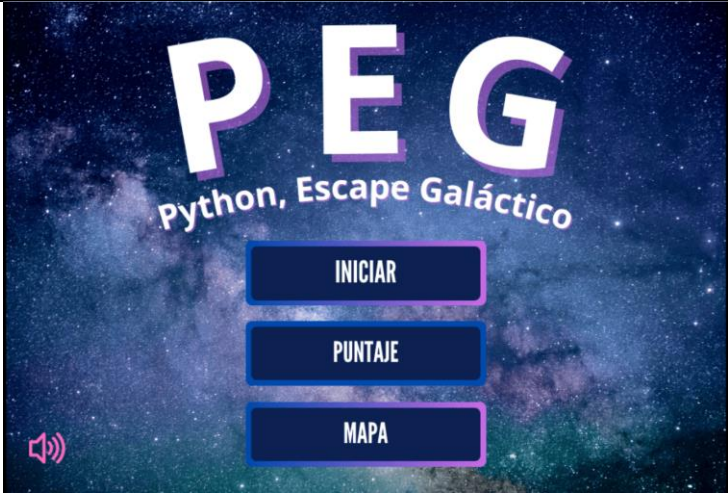
**Figura 7** Diagrama de secuencia



**FUENTE: AUTOR**

#### 4.2.4. Casos de uso extendidos

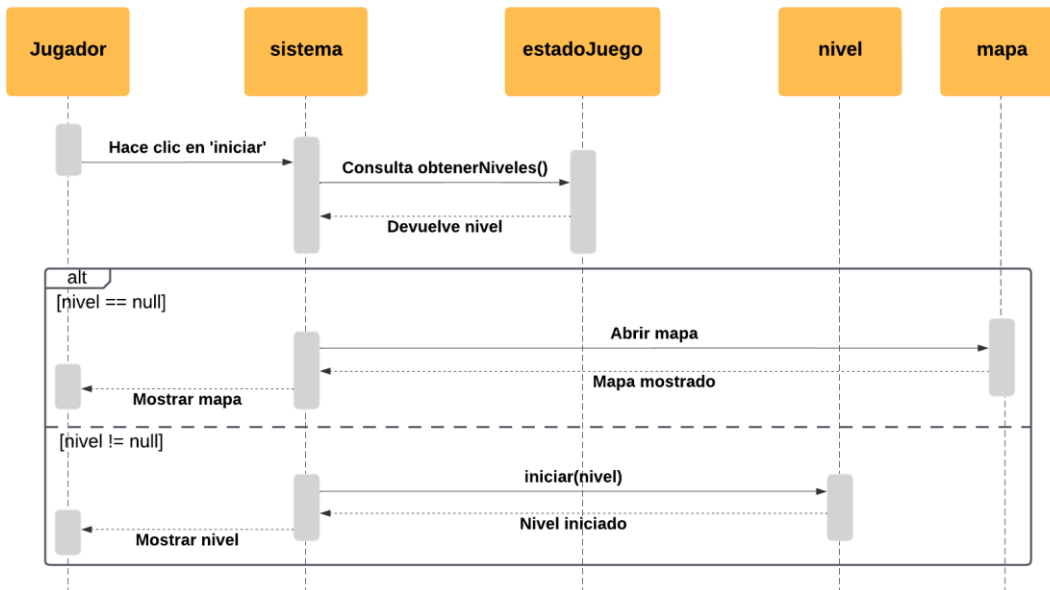
**Tabla 10** Caso de uso: Iniciar Juego

<b>Caso de uso:</b>	CU_01
<b>Nombre:</b>	Iniciar Juego
<b>Actores:</b>	Jugador
<b>Descripción:</b>	Permite al jugador comenzar una partida nueva o continuar desde el último nivel no completado. Si todos los niveles están completados, redirige al jugador al mapa de niveles.
<b>Precondiciones:</b>	El jugador debe estar en la pantalla de inicio.
<b>Postcondiciones:</b>	El jugador se encuentra en un nivel no completado o en el mapa de niveles.
<b>Flujo principal de eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Jugador selecciona la opción 'Iniciar'	2. El sistema verifica el progreso guardado.
	3. El sistema cargará el primer nivel incompleto.
4. El jugador comienza a jugar el nivel mostrado.	
<b>Flujo alterno</b>	
4.1. Si todos los niveles están completados, el sistema redirige al jugador al mapa de niveles.	
<b>Interfaz</b>	

**FUENTE:** AUTOR

El diagrama que se encuentra en la **Figura 8** muestra el flujo de eventos que se realiza cuando el usuario hace clic en 'Iniciar' en la página principal del juego.

**Figura 8** Diagrama de secuencia para iniciar juego



FUENTE: AUTOR

**Tabla 11** Caso de Uso: Mostrar Mapa de niveles

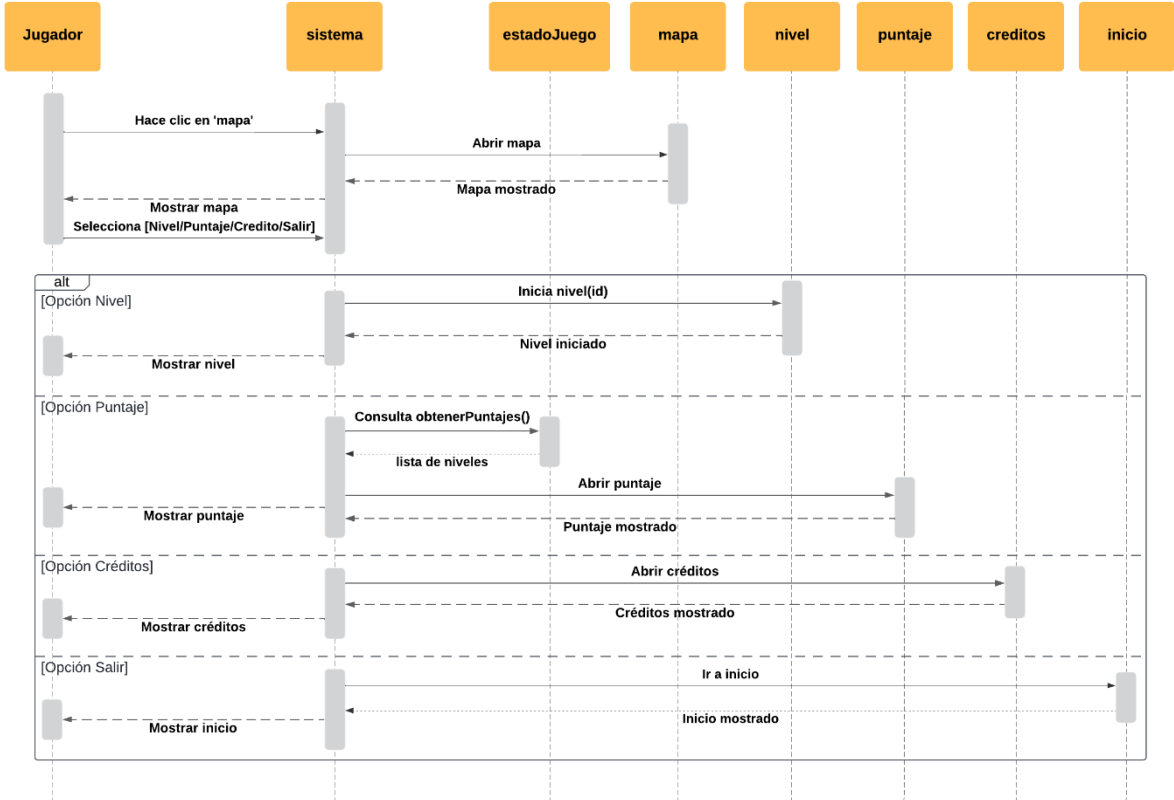
<b>Caso de uso:</b>	CU_02
<b>Nombre:</b>	Mostrar mapa de niveles
<b>Actores:</b>	Jugador
<b>Descripción:</b>	Permite al jugador visualizar y acceder a todos los niveles disponibles, además de acceder a los créditos y los puntajes de los niveles desde una interfaz central.
<b>Precondiciones:</b>	El jugador debe estar en la pantalla de inicio o en el menú de opciones.
<b>Postcondiciones:</b>	El mapa de niveles es visible para el jugador.
<b>Flujo principal de eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. El jugador selecciona la opción para ver el mapa de niveles.	2. El sistema muestra el mapa de niveles, incluyendo opciones para acceder a créditos y puntajes.
3. El jugador elige una opción en el mapa de niveles	



**FUENTE: AUTOR**

Continuando con el flujo anterior en la **Figura 9** se encuentra el diagrama de secuencia del mapa de niveles. Este diagrama detalla todas las opciones disponibles en el mapa y se activa cuando el jugador selecciona la opción para ir al mapa desde cualquier parte del juego.

**Figura 9** Diagrama de secuencia del mapa de niveles



**FUENTE: AUTOR**

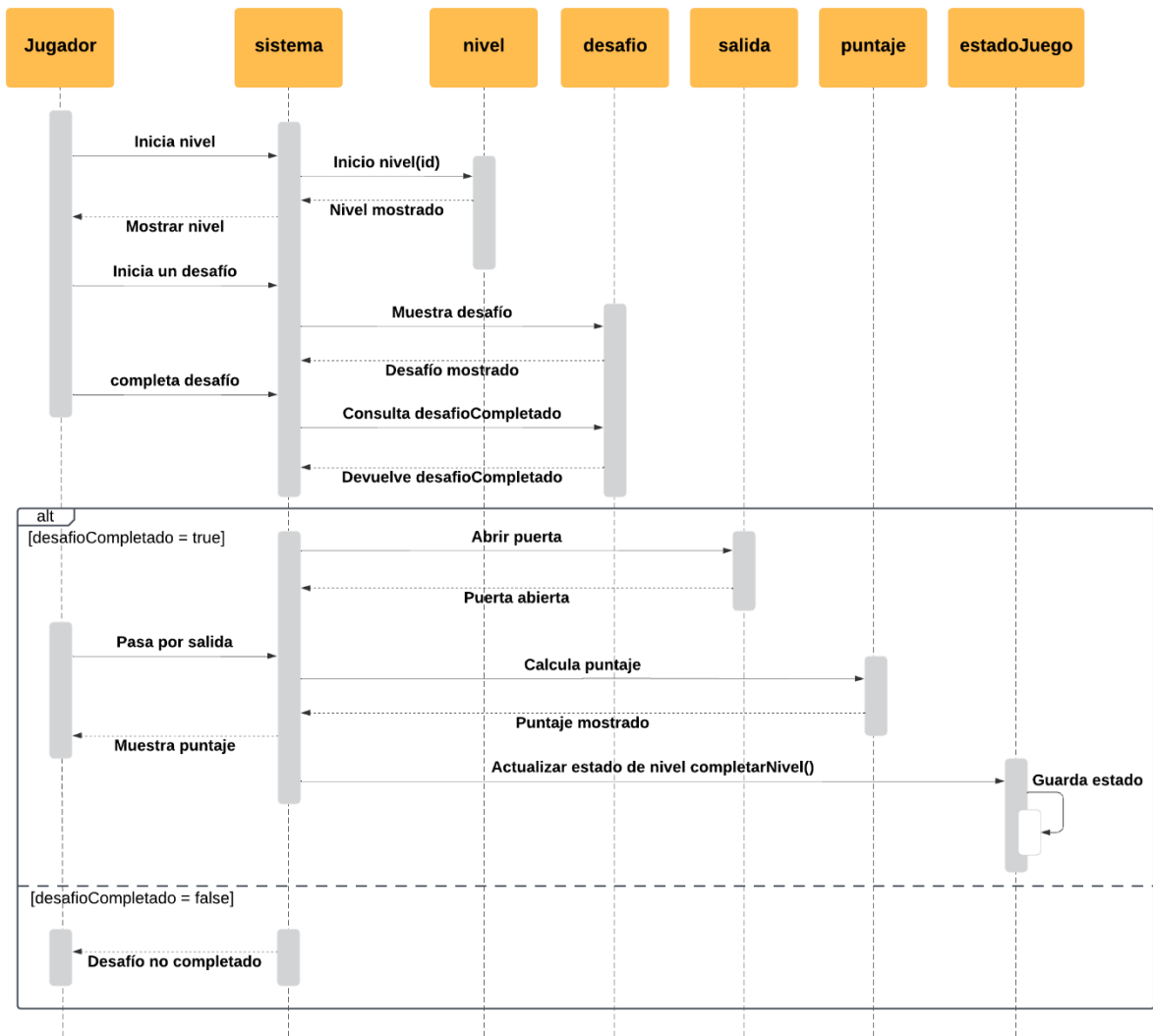
**Tabla 12** Caso de Uso: Seleccionar Nivel

<b>Caso de uso:</b>	CU_03	
<b>Nombre:</b>	Seleccionar niveles	
<b>Actores:</b>	Jugador	
<b>Descripción:</b>	Permite al jugador elegir un nivel específico desde el mapa de niveles para comenzar o reanudar el juego.	
<b>Precondiciones:</b>	El mapa de niveles está siendo mostrado y el nivel anterior al seleccionado debe haber sido completado	
<b>Postcondiciones:</b>	El nivel seleccionado se carga y el jugador comienza a jugar.	
<b>Flujo principal de eventos</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El jugador elige un nivel del mapa.	2. El sistema verifica que el nivel anterior este completo.	
	3. El sistema carga el nivel seleccionado.	
4. El jugador comienza a jugar el nivel o interactúa con los elementos del nivel cargado.		
<b>Flujo alterno</b>		
2.1. El nivel seleccionado es 0, puede cargar el nivel sin verificar.		
3.1. El nivel anterior no está completo, muestra un mensaje de advertencia y no carga el nivel.		

**FUENTE: AUTOR**

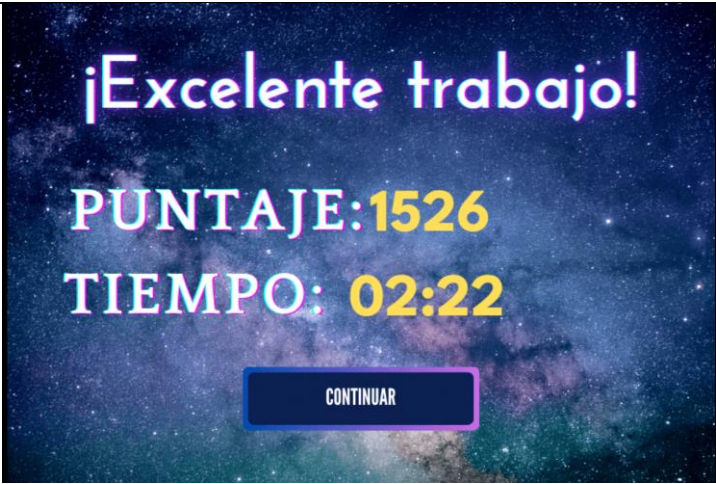
La **Figura 10** muestra el diagrama de secuencia de los niveles del juego. Este diagrama representa cómo interactúan los diferentes componentes del juego durante el progreso del jugador a través de los niveles.

**Figura 10** Diagrama de secuencia para jugar niveles



FUENTE: AUTOR

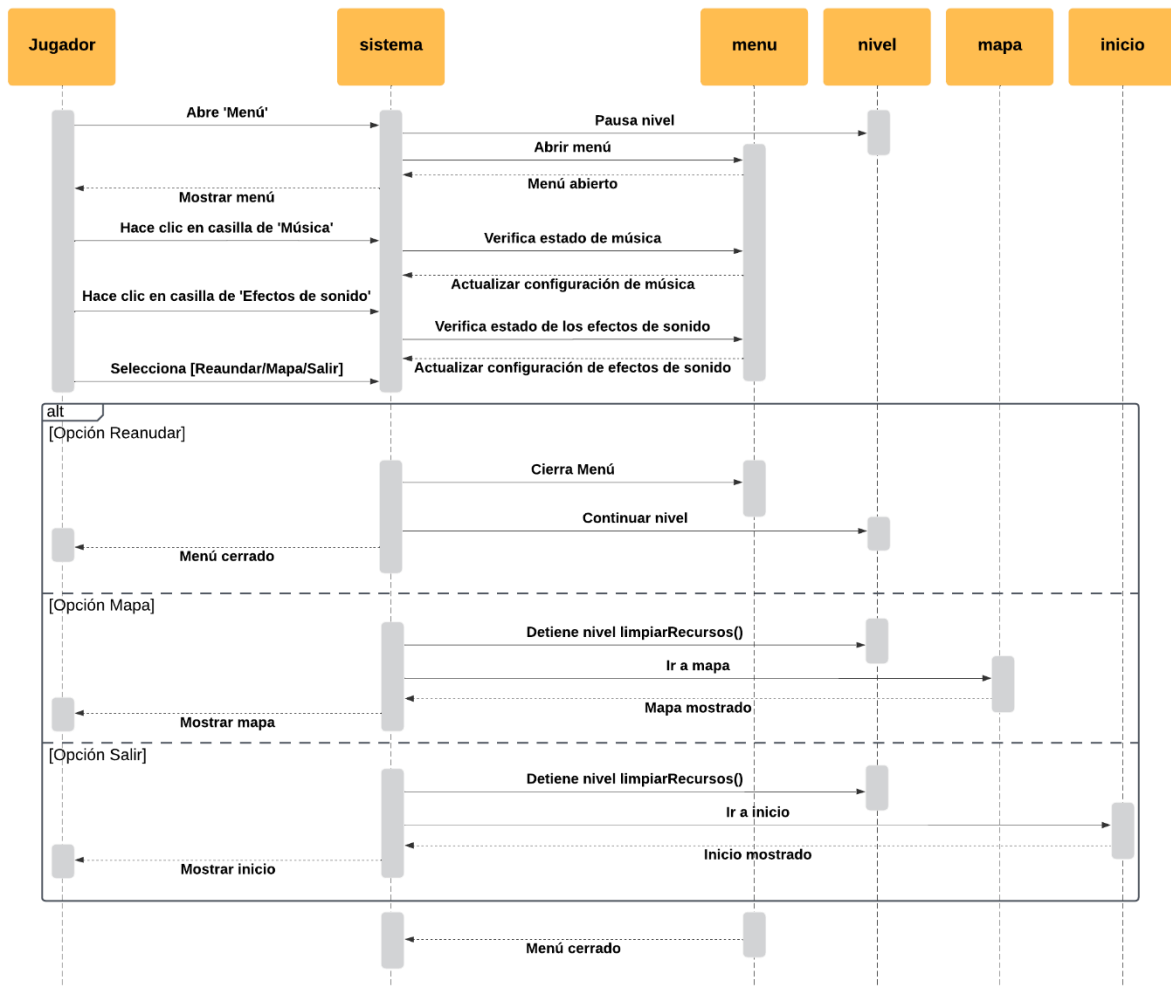
**Tabla 13** Caso de uso: Completar Nivel

<b>Caso de uso:</b>	CU_04
<b>Nombre:</b>	Completar Nivel
<b>Actores:</b>	Jugador
<b>Descripción:</b>	Permite al jugador completar un nivel después de finalizar todos los desafíos y seleccionar la opción de salida.
<b>Precondiciones:</b>	Todos los desafíos dentro del nivel han sido completados.
<b>Postcondiciones:</b>	El nivel se marca como completado en el sistema.
<b>Flujo principal de eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. El jugador selecciona la opción de salida después de completar todos los desafíos.	2. El sistema marca el nivel como completado y guarda el progreso.
	3. El sistema calcula el puntaje total del jugador basado en su rendimiento, muestra el puntaje calculado al jugador y guarda el puntaje en el sistema.
4. El jugador decide qué hacer a continuación	
<b>Flujo alterno</b>	
1.1. El jugador sale del nivel sin completarlo por medio del menú.	
<b>Interfaz</b>	

FUENTE: AUTOR

Finalmente, en la **Figura 11** se presenta el diagrama de secuencia del menú de opciones del juego, al cual solo se puede acceder si el jugador se encuentra dentro de un nivel.

**Figura 11** Diagrama de secuencia del menú de opciones

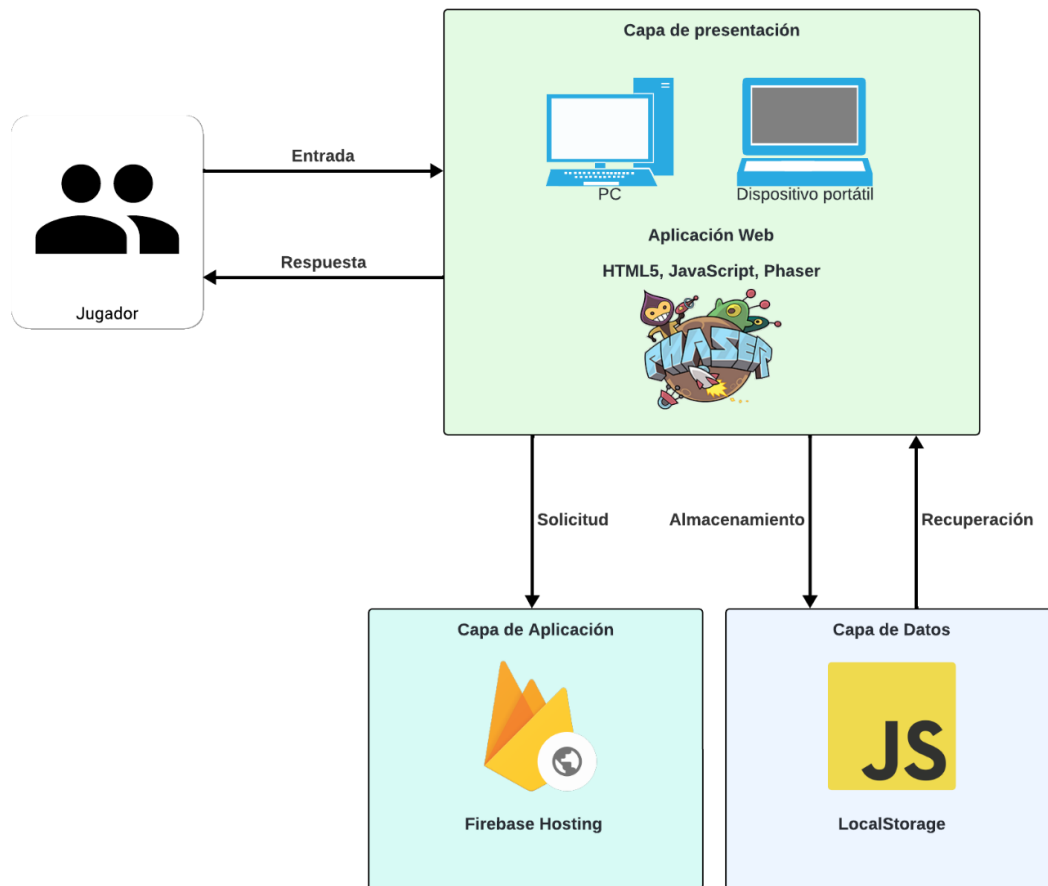


FUENTE: AUTOR

#### 4.2.5. Diagrama de arquitectura

En la **Figura 12** se muestra la arquitectura de la aplicación, ilustrando cómo el usuario interactúa directamente con la capa de presentación, desarrollada para PC/Laptop utilizando HTML5, JavaScript y el framework Phaser. Esta capa está conectada a la capa de servicio, que emplea Firebase Hosting como servidor. Además, la capa de presentación está vinculada a la capa de datos, que utiliza localStorage para la persistencia de datos del juego, permitiendo que la información como el progreso del usuario y configuraciones se mantengan almacenadas directamente en el dispositivo del usuario.

**Figura 12** Diagrama de arquitectura



**FUENTE:** AUTOR

#### 4.2.6. Puntaje del juego

Cada nivel del juego tiene un tiempo estándar de 5 minutos y está diseñado para estudiantes de 5to a 10mo grado según su velocidad de lectura. Los estudiantes de quinto y sexto grado leen de 115 a 134 palabras por minuto [68], [69], [70], lo que les permite dedicar entre 1,78 y 1,93 minutos a comprender las instrucciones y el contexto del juego, lo que les da suficiente tiempo para actuar y resolver el desafío. Se espera tener entre 150 - 200 palabras de explicación sobre programación y 30 – 80 palabras de la historia del juego por cada nivel, dando así un promedio de 230 palabras. Con esto daría un tiempo estimado entre 1.78 minutos (01:47) y 1.93 minutos (01:56) en promedio de lectura por nivel. Esto les deja un margen de más de 3 minutos adicionales para completar el nivel con el mejor puntaje, asegurando que tengan suficiente tiempo para actuar y resolver el desafío.

Por ello, se manejan puntajes de 1000 puntos por completar el nivel en cinco minutos, disminuyendo el puntaje si el jugador tarda en más en completar el nivel y aumentando si lo completa en menor tiempo. Si el tiempo es mayor o igual a 10 el puntaje será 0.

$$Puntos = 1000 + (1000 \times (\frac{5-T}{5}))$$

T = tiempo en minutos

$$T = minutos + \left(\frac{segundos}{60}\right)$$

Por ejemplo, si tarda 04:36 tendría un tiempo de 4.58 y un puntaje de 1083 (redondeando el puntaje para que sea un entero).

#### **4.2.7. Niveles**

El juego ha sido diseñado con niveles progresivos para enseñar y reforzar conceptos fundamentales de programación en Python. A continuación, se detalla cómo cada nivel está estructurado para desarrollar habilidades específicas en los estudiantes.

##### **Nivel 0: Introducción a python - print()**

Utilizar ejercicios prácticos para enseñar a los estudiantes cómo emplear la función print() en Python, permitiéndoles mostrar mensajes en la consola.

##### **Nivel 1: Variables**

Utilizar ejemplos prácticos para enseñar a los estudiantes a declarar y usar variables, facilitando el almacenamiento y manejo de datos en la programación.

##### **Nivel 2: Tipos de datos**

Aplicar actividades interactivas para explicar a los estudiantes los distintos tipos de datos, mejorando su capacidad para seleccionar el tipo adecuado para cada situación en la programación.

### 4.3. Evaluación de usabilidad

#### 4.3.1. Resultados Grupo A

A continuación, se presentan los resultados individuales de cada participante. La **Tabla 14** resume las respuestas a las 10 preguntas del cuestionario SUS y su puntuación final de usabilidad.

**Tabla 14** Resultados del cuestionario SUS - Grupo A

N°	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Puntaje
1	4	2	3	4	5	4	3	3	4	2	60
2	4	3	5	4	4	1	4	2	4	4	67,5
3	5	1	4	3	5	2	5	2	5	3	82,5
4	5	2	4	5	4	1	4	5	2	5	52,5
5	3	3	5	4	4	3	4	2	4	4	60
6	4	2	3	4	5	3	4	1	4	3	67,5
7	5	2	4	4	5	1	4	2	3	2	75
8	2	3	4	5	4	3	3	3	4	3	50
9	5	3	5	4	4	2	5	5	3	5	57,5
10	4	5	4	4	5	2	5	4	5	2	65

**FUENTE:** AUTOR

El análisis de los resultados del cuestionario SUS, aplicado a 10 estudiantes de 9 a 11 años refleja que el promedio de usabilidad es de 63.75. Teniendo en cuenta la interpretación de puntajes de Sauro [71] se encuentra dentro del promedio marginal que indica una usabilidad media. Adicionalmente, se tiene una desviación estándar de 9.44, esto puede señalar que existe un rango amplio de percepción en los estudiantes, ya que parece fácil de usar para unos, mientras que otros estudiantes muestran más dificultad al utilizar el juego.

Teniendo en cuenta las edades de los participantes, los estudiantes de 9 años tienen la puntuación más alta de usabilidad de 67.5 y una desviación estándar de 0. Esto sugiere que este grupo de edad encontró el sistema más fácil de usar a diferencia de los otros dos grupos, el grupo de 10 años y 11 años tienen un puntaje promedio de 62.50 y 63.00, mientras que se presenta una desviación estándar mucho mayor que la anterior de 11.46 y 12.17 respectivamente. Por lo cual, los estudiantes de 10 y 11 pueden haber enfrentado más inconvenientes en el sistema.

Las respuestas a las preguntas abiertas ayudaron a tener una mejor comprensión de la experiencia de los estudiantes del Grupo A en el juego. A continuación, se presenta una tabla resumiendo las respuestas de los estudiantes como puntos importantes del juego.

**Tabla 15** Resultados de las preguntas abiertas - Grupo A

<b>Pregunta</b>	<b>Puntos importantes</b>
¿Qué te gustó más del juego?	<ul style="list-style-type: none"> <li>• Todo el juego</li> <li>• Ejercicios</li> <li>• Gráficos</li> <li>• Niveles para escapar y códigos</li> <li>• Desafíos y pistas</li> </ul>
¿Hubo algo que te resultó confuso o difícil?	<ul style="list-style-type: none"> <li>• Último nivel (Tipos de datos)</li> <li>• Encontrar las pistas</li> <li>• Dificultad de algunos ejercicios</li> <li>• Ambigüedad en las explicaciones</li> </ul>
¿Cambiarías o mejorarías algo en el juego?	<ul style="list-style-type: none"> <li>• Incorporar la opción de retroceder en las animaciones</li> <li>• Simplificar las pistas</li> <li>• Cambiar la música</li> <li>• Añadir más niveles y desafíos</li> </ul>
¿Te gustaría seguir aprendiendo con este juego?	<ul style="list-style-type: none"> <li>• 8 estudiantes respondieron 'sí';</li> <li>• 3 estudiantes respondieron 'no sé'</li> </ul>

**FUENTE:** AUTOR

Por último, la prueba de conocimiento constó de cuatro preguntas, cada una sobre un tema que fue explicado dentro del juego, con cada pregunta se pudo determinar que temas resultaron más difíciles de aprender para los estudiantes.

**Tabla 16** Resultados de la prueba de conocimientos - Grupo A

<b>N°</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>Calificación</b>
<b>1</b>	1	0	1	0	5
<b>2</b>	1	0	1	0	5
<b>3</b>	1	1	0	1	7,5
<b>4</b>	1	0	0	1	5
<b>5</b>	1	1	1	0	7,5
<b>6</b>	1	1	0	0	5
<b>7</b>	1	1	1	1	10
<b>8</b>	1	0	0	1	5
<b>9</b>	1	1	0	1	7,5
<b>10</b>	1	1	1	0	7,5
<b>Total</b>	10	6	5	5	6,50

**FUENTE:** AUTOR

Como se visualizó en la tabla anterior, las preguntas presentaron distintos niveles de dificultad para los estudiantes. La pregunta 1 fue respondida correctamente por todos los estudiantes, la pregunta 2 presentó un poco más de conflicto marcando 6 correctas y 4 incorrectas. Las preguntas 3 y 4 tuvieron una tasa de éxito del 50% marcando 5 correctas y 5 incorrectas, revelando que fueron temas complicados para los estudiantes.

El análisis a las calificaciones obtenidas por los estudiantes revela que el promedio de calificación es 6,50 lo que sugiere un rendimiento moderado en aprendizaje. Además, la desviación estándar de estas calificaciones es de 1.66, este valor indica que las calificaciones no están muy dispersas respecto al promedio. En conjunto con las preguntas abiertas, se puede determinar que el tema de mayor dificultad fueron los tipos de datos, mientras que los primeros niveles no presentaron dificultad en resolver.

### 4.3.2. Resultados Grupo B

A continuación, se presentan los resultados individuales de cada participante del grupo B. La **Tabla 17** resume las respuestas a las 10 preguntas del cuestionario SUS y su puntuación final de usabilidad.

**Tabla 17** Resultados del cuestionario SUS - Grupo B

N°	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Puntaje
1	4	3	4	3	4	3	5	2	2	3	62,5
2	3	2	5	2	4	2	5	3	2	3	67,5
3	4	1	4	2	5	1	3	1	4	2	82,5
4	4	1	4	4	5	1	4	2	5	2	80
5	5	3	3	2	5	1	5	1	4	3	80
6	5	2	4	4	5	1	4	1	5	2	82,5
7	5	1	4	1	5	2	4	1	5	1	92,5
8	4	1	4	2	5	1	5	1	4	2	87,5
9	4	1	5	2	5	1	4	1	5	2	90
10	4	2	4	4	5	1	5	3	5	2	77,5

**FUENTE:** AUTOR

El análisis de los resultados del cuestionario SUS, aplicado a 10 estudiantes de 12 a 14 años refleja que el promedio de usabilidad es de 80,25. Muy por delante del promedio del Grupo A, teniendo en cuenta la interpretación de puntajes de Sauro [71] se encuentra dentro de la escala aceptable que indica un puntaje de usabilidad por encima del promedio. Adicionalmente, se tiene una desviación estándar de 8.91, esto puede señalar que podría haber aspectos específicos en el juego que no funcionaron de la misma manera para todos los usuarios.

Dentro de Grupo B, se pueden tomar diferentes enfoques en el análisis de datos del cuestionario SUS. Los estudiantes varones muestran un promedio más alto de usabilidad de 85,00 con una desviación estándar de 5,92. Indicando que este grupo encontró el juego considerablemente más fácil de usar en comparación con las mujeres que tuvieron un promedio de 73,13 y una desviación estándar de 9,66. A pesar de las variaciones en los puntajes, ambos se encuentran dentro del rango aceptable de usabilidad.

En términos de edad, al igual que en el grupo A los estudiantes de menor edad, en este caso en el rango de 12 a 14, tuvieron mayor puntaje en el promedio, siendo que los estudiantes de 12 años lograron un puntaje medio de 83,33 y una desviación estándar de 3,82. Por otro lado, los estudiantes de 13 años tuvieron un puntaje promedio de 75,00 y una desviación estándar de 6,61. Mientras que los estudiantes de 14 años tuvieron un puntaje promedio de 81,88 y una desviación de 13,60. Lo que indica que, aunque encontraron el juego fácil de usar, las experiencias individuales variaron más que en los otros grupos de edad.

La **Tabla 18** resume las respuestas a las preguntas abiertas, destacando los aspectos principales que deben considerarse.

**Tabla 18** Resultados de las preguntas abiertas - Grupo B

<b>Pregunta</b>	<b>Puntos importantes</b>
¿Qué te gustó más del juego?	<ul style="list-style-type: none"> <li>• Temática espacial y de programación</li> <li>• Mecánicas de búsqueda y escape</li> <li>• Las imágenes</li> <li>• Concepto general del juego</li> </ul>
¿Hubo algo que te resultó confuso o difícil?	<ul style="list-style-type: none"> <li>• Algunos puzzles</li> <li>• Temas de programación y tipos de datos</li> <li>• Localización de pistas</li> </ul>
¿Cambiarías o mejorarías algo en el juego?	<ul style="list-style-type: none"> <li>• Añadir más niveles y contenido</li> <li>• Incluir más ejercicios prácticos y ejemplos</li> <li>• Retroceder en las animaciones</li> <li>• Música de fondo</li> </ul>
¿Te gustaría seguir aprendiendo con este juego?	<ul style="list-style-type: none"> <li>• Todos los estudiantes mostraron interés en seguir aprendiendo con este juego</li> </ul>

**FUENTE: AUTOR**

Finalmente, la prueba de conocimientos en la cual se muestra una mejora significativa en comparación con el Grupo A.

**Tabla 19** Resultados de la prueba de conocimientos - Grupo B

N°	P1	P2	P3	P4	Calificación
1	1	0	1	0	5
2	1	1	1	0	7,5
3	1	1	0	1	7,5
4	1	1	1	1	10
5	0	1	1	1	7,5
6	1	1	1	1	10
7	1	1	1	1	10
8	1	1	1	1	10
9	1	1	1	1	10
10	1	1	1	0	7,5
<b>Total</b>	9	9	9	7	8,50

**FUENTE:** AUTOR

Como se refleja en los resultados obtenidos, las respuestas muestran un cambio significativo en comparación con el Grupo A. Las preguntas 1, 2 y 3 fueron respondidas correctamente por la mayoría de los estudiantes, con un total de 9 respuestas correctas y 1 incorrecta. Sin embargo, la pregunta 4 tiene un nivel más bajo de respuestas correctas con un total de 7, lo que destaca que fue una pregunta de mayor dificultad para los estudiantes.

El análisis de las calificaciones revela que el promedio fue de 8,5. Sugiriendo un rendimiento generalmente bueno, si se tiene en cuenta la desviación estándar de 1,66 se puede determinar que las calificaciones de los estudiantes no están muy dispersas entre sí, por lo cual el rendimiento general de los estudiantes fue consistente.

#### **4.4. Discusión**

En relación con la usabilidad del juego, los puntajes del cuestionario SUS sugieren que la usabilidad es aceptable. Sin embargo, al analizar los resultados de los dos grupos, se observa que los estudiantes de 9 a 11 años tuvieron más dificultades al utilizar el juego que los estudiantes de 12 a 14 años. Durante la evaluación de usabilidad, que se llevó a cabo de manera presencial, se identificaron los factores que influyeron en su rendimiento. Se observó que los estudiantes de 9 a 11 años tenían más dificultades en concentrarse en el juego.

Los niños de 9 años lograban completar los niveles con una mínima intervención de un adulto, pero su nivel de aprendizaje fue bastante bajo. Por otro lado, los niños de 11 a 12 años asimilaban las explicaciones, pero mostraban mayor distracción al enfrentar los desafíos del juego. En un caso particular, el participante no mostró ningún interés por el juego, lo que se reflejó en una puntuación de usabilidad muy baja. Este resultado resalta la importancia de considerar la motivación y la atracción del estudiante hacia una aplicación como parte de la experiencia de usabilidad.

El segundo grupo de estudiantes, compuesto por jóvenes de 12 a 14 años, experimentó una mejora significativa tanto en la puntuación de usabilidad como en la prueba de conocimiento. Durante la evaluación, solo se observó un mínimo de desinterés por el juego. Aunque al principio necesitaron una explicación sobre cómo funciona el juego, lograron completar los niveles sin problemas.

Al analizar las diferencias entre los dos grupos, teniendo en cuenta los rangos de edad, se evidencia la necesidad de realizar ajustes en el diseño del juego para adecuarlo a las necesidades de cada grupo. Este análisis permite identificar cómo las distintas características del juego impactan de manera diferente a los usuarios, dependiendo de su edad.

Los comentarios sobre los aspectos más atractivos del juego indican que los usuarios tienen una opinión positiva sobre la temática y las mecánicas del juego. Esto es importante para destacar las fortalezas del juego, lo que incentiva a los estudiantes a seguir aprendiendo programación. Por otro lado, las dificultades encontradas revelan que hubo mayor confusión en el último nivel de tipos de datos. Esto ofrece una oportunidad para analizar cómo estas dificultades afectaron la experiencia del usuario y proponer mejoras. Por ejemplo, mejorar las instrucciones y pistas podría hacer que el juego sea más accesible y agradable para todos los usuarios.

Los resultados de las pruebas de conocimiento muestran un rendimiento moderado en la comprensión de conceptos de programación, con variaciones en el éxito según la pregunta y el rango de edad. Este resultado proporciona una base para discutir cómo el juego combina educación con entretenimiento y cómo podría mejorarse para apoyar de manera más efectiva el aprendizaje de lenguajes de programación.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

## 5.1. Conclusiones

A partir de los resultados obtenidos en este proyecto de investigación, se pueden destacar las siguientes conclusiones:

La aplicación web basada en Salas de Escape ha logrado captar la atención de los estudiantes y facilitar la introducción a los fundamentos de programación. Este diseño inicial ha sido efectivo en fomentar el aprendizaje en un contexto lúdico y entretenido, adecuado para estudiantes desde la educación básica hasta la media superior.

Durante la fase inicial de entrevistas para recopilar requisitos, se observó una aceptación positiva de la aplicación entre los usuarios representativos. Los estudiantes expresaron entusiasmo por la oportunidad de aprender programación mediante un formato de videojuego, y consideraron que este enfoque podría mejorar significativamente su comprensión de los conceptos de programación. Por su parte, los docentes reconocieron la importancia de integrar tecnologías de juegos en los procesos educativos y destacaron que podría fomentar un aprendizaje más profundo.

Esta positiva receptividad inicial fue esencial para validar la dirección del proyecto y confirmar la necesidad de continuar desarrollando y refinando la aplicación para satisfacer las necesidades educativas de los estudiantes. Asimismo, destacó la importancia de seguir colaborando estrechamente con estos grupos de usuarios para asegurar que el diseño del proyecto esté alineado con los objetivos de aprendizaje y las preferencias del usuario final.

El diseño de un prototipo de aplicación web para facilitar el aprendizaje de los conceptos básicos de programación cumplió con las expectativas establecidas en las fases iniciales de planificación y diseño. Durante el proceso de desarrollo, se tomó en cuenta las sugerencias de los usuarios para optimizar la usabilidad y el impacto educativo del prototipo, lo que resultó en mejoras significativas en la interfaz y la funcionalidad general.

La usabilidad general de la aplicación ha sido calificada como aceptable por los usuarios representativos, cumpliendo así con los objetivos del proyecto de evaluar y mejorar continuamente el prototipo basado en la experiencia del usuario. Los resultados obtenidos del cuestionario SUS y las opiniones recogidas subrayan la necesidad de realizar revisiones constantes para optimizar tanto la interfaz como la experiencia general del usuario.

## 5.2. Recomendaciones

- Se recomienda considerar ajustes en el diseño del juego para proporcionar experiencias más personalizadas en función de la edad de los usuarios, ampliando así la variedad de conceptos de programación incluidos en la aplicación. Además, es crucial desarrollar rutas de aprendizaje que se adapten progresivamente a los distintos niveles de habilidad y conocimiento de los estudiantes.
- Es importante introducir funcionalidades que permitan a los estudiantes personalizar su experiencia de aprendizaje dentro de la aplicación. Esto podría incluir opciones para ajustar el nivel de dificultad de los desafíos y seleccionar distintas rutas de aprendizaje que se alineen con sus necesidades o edad.
- Se debe continuar con las investigaciones para evaluar la efectividad de la aplicación en diversos entornos educativos. Además, es esencial mantener un proceso continuo de recopilación y análisis de comentarios de los usuarios para iterar y mejorar la aplicación de manera constante.

**CAPÍTULO VI**  
**BIBLIOGRAFÍA**

- [1] R. S. N. Lindberg, T. H. Laine, and L. Haaranen, “Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games,” *British Journal of Educational Technology*, vol. 50, no. 4, pp. 1979–1995, Jul. 2019, doi: 10.1111/bjet.12685.
- [2] R. Barroso, A. Castro, and A. Rocha, “Computer programming as a tool to improve mathematic skills in basic education,” in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Jun. 2018, pp. 1–3. doi: 10.23919/CISTI.2018.8399192.
- [3] C. Dindler, R. Smith, and O. S. Iversen, “Computational empowerment: participatory design in education,” *CoDesign*, vol. 16, no. 1, pp. 66–80, Jan. 2020, doi: 10.1080/15710882.2020.1722173.
- [4] S. Bell, T. Frey, and E. Vasserman, “Spreading the word,” in *Proceedings of the 45th ACM technical symposium on Computer science education*, New York, NY, USA: ACM, Mar. 2014, pp. 187–192. doi: 10.1145/2538862.2538963.
- [5] A. Yadav, H. Hong, and C. Stephenson, “Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms,” *TechTrends*, vol. 60, no. 6, pp. 565–568, Nov. 2016, doi: 10.1007/s11528-016-0087-7.
- [6] V. F. dos Santos, L. R. Sabino, G. M. Morais, and C. A. Goncalves, “E-Commerce: A Short History Follow-up on Possible Trends,” *International Journal of Business Administration*, vol. 8, no. 7, p. 130, Oct. 2017, doi: 10.5430/ijba.v8n7p130.
- [7] C. J. Costa, M. Aparicio, and C. Cordeiro, “Web-Based Graphic Environment to Support Programming in the Beginning Learning Process,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7522 LNCS, Springer, Berlin, Heidelberg, 2012, pp. 413–416. doi: 10.1007/978-3-642-33542-6\_41.
- [8] J. A. Rodríguez-Martínez, J. A. González-Calero, and J. M. Sáez-López, “Computational thinking and mathematics using Scratch: an experiment with sixth-grade students,” *Interactive Learning Environments*, vol. 28, no. 3, pp. 316–327, Apr. 2020, doi: 10.1080/10494820.2019.1612448.

- [9] V. Garneli and K. Chorianopoulos, "Programming video games and simulations in science education: exploring computational thinking through code analysis," *Interactive Learning Environments*, vol. 26, no. 3, pp. 386–401, Apr. 2018, doi: 10.1080/10494820.2017.1337036.
- [10] M. L. Cabra Páez and S. A. Ramírez Gamboa, "Desarrollo del pensamiento computacional y las competencias matemáticas en análisis y solución de problemas: una experiencia de aprendizaje con Scratch en la plataforma Moodle," *Revista Educación*, vol. 6, no. 1, pp. 171–187, Oct. 2021, doi: 10.15517/revedu.v46i1.44970.
- [11] R. P. Medeiros, G. L. Ramalho, and T. P. Falcao, "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," *IEEE Transactions on Education*, vol. 62, no. 2, pp. 77–90, May 2019, doi: 10.1109/TE.2018.2864133.
- [12] C. Kazimoglu, M. Kiernan, L. Bacon, and L. MacKinnon, "Learning Programming at the Computational Thinking Level via Digital Game-Play," *Procedia Comput Sci*, vol. 9, pp. 522–531, Jan. 2012, doi: 10.1016/j.procs.2012.04.056.
- [13] H. Yildiz Durak, "The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving," *Technology, Knowledge and Learning*, vol. 25, no. 1, pp. 179–195, Mar. 2020, doi: 10.1007/s10758-018-9391-y.
- [14] B. Kaldarova et al., "Applying game-based learning to a primary school class in computer science terminology learning," *Front Educ (Lausanne)*, vol. 8, Feb. 2023, doi: 10.3389/educ.2023.1100275.
- [15] P. Fotaris and T. Mastoras, "Escape Rooms for Learning: A Systematic Review," in *Proceedings of the 12th European Conference on Game Based Learning, ACPI*, Oct. 2019, p. 30. doi: 10.34190/GBL.19.179.
- [16] P. J. Vergragt, "How technology could contribute to a sustainable world," *GTI Paper Series*, vol. 28, 2006.
- [17] "Sustainable Development Goal 4: Educación de calidad | Naciones Unidas en Ecuador." Accessed: Jul. 30, 2023. [Online]. Available: <https://ecuador.un.org/es/sdgs/4>

- [18] I. Kotini and S. Tzelepi, “A Gamification-Based Framework for Developing Learning Activities of Computational Thinking,” in *Gamification in Education and Business*, vol. 10, Cham: Springer International Publishing, 2015, pp. 219–252. doi: 10.1007/978-3-319-10208-5\_12.
- [19] A. Yadav, J. Good, J. Voogt, and P. Fisser, “Computational Thinking as an Emerging Competence Domain,” 2017, pp. 1051–1067. doi: 10.1007/978-3-319-41713-4\_49.
- [20] J. L. Plass, B. D. Homer, and C. K. Kinzer, “Foundations of Game-Based Learning,” *Educ Psychol*, vol. 50, no. 4, pp. 258–283, Oct. 2015, doi: 10.1080/00461520.2015.1122533.
- [21] ... M. R. conference. A. computer society and undefined 2014, “Give P’sa chance: Projects, peers, passion, play,” [media.mit.edu](http://media.mit.edu), Accessed: Jun. 24, 2023. [Online]. Available: <https://www.media.mit.edu/~mres/papers/constructionism-2014.pdf>
- [22] A. Manzano-León et al., “Escape Rooms as a Learning Strategy for Special Education Master’s Degree Students,” *Int J Environ Res Public Health*, vol. 18, no. 14, p. 7304, Jul. 2021, doi: 10.3390/ijerph18147304.
- [23] X. Chen, D. Song, and Y. Tian, “Latent Execution for Neural Program Synthesis,” *Adv Neural Inf Process Syst*, vol. 27, pp. 22196–22208, Jun. 2021, Accessed: Jul. 08, 2023. [Online]. Available: <http://arxiv.org/abs/2107.00101>
- [24] D. Pollak, V. Layka, and A. Sacco, “DSL and Parser Combinator,” in *Beginning Scala 3*, Berkeley, CA: Apress, 2022, pp. 237–245. doi: 10.1007/978-1-4842-7422-4\_10.
- [25] S. Höppner, T. Kehrer, and M. Tichy, “Contrasting dedicated model transformation languages versus general purpose languages: a historical perspective on ATL versus Java based on complexity and size,” *Softw Syst Model*, vol. 21, no. 2, pp. 805–837, Apr. 2022, doi: 10.1007/s10270-021-00937-3.
- [26] S. Grover and R. Pea, “Computational Thinking in K–12,” *Educational Researcher*, vol. 42, no. 1, pp. 38–43, Jan. 2013, doi: 10.3102/0013189X12463051.
- [27] U. Kale et al., “Computational What? Relating Computational Thinking to Teaching,” *TechTrends*, vol. 62, no. 6, pp. 574–584, Nov. 2018, doi: 10.1007/s11528-018-0290-9.

- [28] D. Bau, J. Gray, C. Kelleher, J. Sheldon, and F. Turbak, “Learnable programming,” *Commun ACM*, vol. 60, no. 6, pp. 72–80, May 2017, doi: 10.1145/3015455.
- [29] R. Pelánek and T. Effenberger, “Design and analysis of microworlds and puzzles for block-based programming,” *Computer Science Education*, vol. 32, no. 1, pp. 66–104, Jan. 2022, doi: 10.1080/08993408.2020.1832813.
- [30] “Learn | Code.org.” Accessed: Jul. 10, 2023. [Online]. Available: <https://code.org/learn>
- [31] M. Resnick et al., “Scratch,” *Commun ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009, doi: 10.1145/1592761.1592779.
- [32] Y. Park and Y. Shin, “Text Processing Education Using a Block-Based Programming Language,” *IEEE Access*, vol. 10, pp. 128484–128497, 2022, doi: 10.1109/ACCESS.2022.3227765.
- [33] M. Seraj, E.-S. Katterfeldt, K. Bub, S. Autexier, and R. Drechsler, “Scratch and Google Blockly,” in *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, New York, NY, USA: ACM, Nov. 2019, pp. 1–10. doi: 10.1145/3364510.3364515.
- [34] A. Yadav, C. Mayfield, N. Zhou, S. Hambruch, and J. T. Korb, “Computational Thinking in Elementary and Secondary Teacher Education,” *ACM Transactions on Computing Education*, vol. 14, no. 1, pp. 1–16, Mar. 2014, doi: 10.1145/2576872.
- [35] Z. Yu, M. Gao, and L. Wang, “The Effect of Educational Games on Learning Outcomes, Student Motivation, Engagement and Satisfaction,” *Journal of Educational Computing Research*, vol. 59, no. 3, pp. 522–546, Jun. 2021, doi: 10.1177/0735633120969214.
- [36] R. S. N. Lindberg, T. H. Laine, and L. Haaranen, “Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games,” *British Journal of Educational Technology*, vol. 50, no. 4, pp. 1979–1995, Jul. 2019, doi: 10.1111/bjet.12685.
- [37] Y. Zhonggen, “A Meta-Analysis of Use of Serious Games in Education over a Decade,” *International Journal of Computer Games Technology*, vol. 2019, pp. 1–8, Feb. 2019, doi: 10.1155/2019/4797032.

- [38] H. Lode, G. E. Franchi, and N. G. Frederiksen, "Machineers," in CHI '13 Extended Abstracts on Human Factors in Computing Systems, New York, NY, USA: ACM, Apr. 2013, pp. 2639–2642. doi: 10.1145/2468356.2479483.
- [39] M. D. Coover, J. Winner, W. Bennett, and D. J. Howard, "Serious Games are a Serious Tool for Team Research," *International Journal of Serious Games*, vol. 4, no. 1, Mar. 2017, doi: 10.17083/ijsg.v4i1.141.
- [40] M. Arias-Calderón, J. Castro, and S. Gayol, "Serious Games as a Method for Enhancing Learning Engagement: Student Perception on Online Higher Education During COVID-19," *Front Psychol*, vol. 13, p. 889975, Apr. 2022, doi: 10.3389/fpsyg.2022.889975.
- [41] T. Govender and J. Arnedo-Moreno, "An Analysis of Game Design Elements Used in Digital Game-Based Language Learning," *Sustainability*, vol. 13, no. 12, p. 6679, Jun. 2021, doi: 10.3390/su13126679.
- [42] P. M, N. M. N, R. Dakshina, S. S., and B. S. R, "Learning Analytics: Game-based Learning for Programming Course in Higher Education," *Procedia Comput Sci*, vol. 172, pp. 468–472, Jan. 2020, doi: 10.1016/j.procs.2020.05.143.
- [43] L. C. Shum, Y. Rosunally, S. Scarle, and K. Munir, "Personalised Learning through Context-Based Adaptation in the Serious Games with Gating Mechanism," *Educ Inf Technol (Dordr)*, vol. 28, no. 10, pp. 13077–13108, Oct. 2023, doi: 10.1007/s10639-023-11695-8.
- [44] H. Montes, R. Hijon-Neira, D. Perez-Marin, and S. Montes, "Using an Online Serious Game to Teach Basic Programming Concepts and Facilitate Gameful Experiences for High School Students," *IEEE Access*, vol. 9, pp. 12567–12578, 2021, doi: 10.1109/ACCESS.2021.3049690.
- [45] S. Mystakidis, E. Cachafeiro, and I. Hatzilygeroudis, "Enter the Serious E-scape Room: A Cost-Effective Serious Game Model for Deep and Meaningful E-learning," in 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), IEEE, Jul. 2019, pp. 1–6. doi: 10.1109/IISA.2019.8900673.
- [46] S. Nicholson, "Creating Engaging Escape Rooms for the Classroom," *Child Educ*, vol. 94, no. 1, pp. 44–49, Jan. 2018, doi: 10.1080/00094056.2018.1420363.

- [47] S. Mystakidis and E. Berki, “The Case of Literacy Motivation,” *International Journal of Web-Based Learning and Teaching Technologies*, vol. 13, no. 1, pp. 64–79, Jan. 2018, doi: 10.4018/IJWLTT.2018010105.
- [48] S. López-Pernas, M. Saqr, A. Gordillo, and E. Barra, “A learning analytics perspective on educational escape rooms,” *Interactive Learning Environments*, vol. 31, no. 10, pp. 6509–6525, Dec. 2023, doi: 10.1080/10494820.2022.2041045.
- [49] C. Lathwesen and N. Belova, “Escape Rooms in STEM Teaching and Learning—Prospective Field or Declining Trend? A Literature Review,” *Educ Sci (Basel)*, vol. 11, no. 6, p. 308, Jun. 2021, doi: 10.3390/educsci11060308.
- [50] A. Makri, D. Vlachopoulos, and R. A. Martina, “Digital Escape Rooms as Innovative Pedagogical Tools in Education: A Systematic Literature Review,” *Sustainability*, vol. 13, no. 8, p. 4587, Apr. 2021, doi: 10.3390/su13084587.
- [51] T. Bratteteig and I. Wagner, “Unpacking the Notion of Participation in Participatory Design,” *Computer Supported Cooperative Work (CSCW)*, vol. 25, no. 6, pp. 425–475, Dec. 2016, doi: 10.1007/s10606-016-9259-4.
- [52] M. Bano and D. Zowghi, “A systematic review on the relationship between user involvement and system success,” *Inf Softw Technol*, vol. 58, pp. 148–169, Feb. 2015, doi: 10.1016/j.infsof.2014.06.011.
- [53] M. M. J. Engelbertink, S. M. Kelders, K. M. Woudt-Mittendorff, and G. J. Westerhof, “Participatory design of persuasive technology in a blended learning course: A qualitative study,” *Educ Inf Technol (Dordr)*, vol. 25, no. 5, pp. 4115–4138, Sep. 2020, doi: 10.1007/s10639-020-10147-x.
- [54] A. Booker and S. Goldman, “Participatory Design Research as a Practice for Systemic Repair: Doing Hand-in-Hand Math Research with Families,” *Cogn Instr*, vol. 34, no. 3, pp. 222–235, Jul. 2016, doi: 10.1080/07370008.2016.1179535.
- [55] S. Merkel and A. Kucharski, “Participatory Design in Gerontechnology: A Systematic Literature Review,” *Gerontologist*, vol. 59, no. 1, pp. e16–e25, Jan. 2019, doi: 10.1093/geront/gny034.
- [56] A. Reddy, C. A. Lester, J. A. Stone, R. J. Holden, C. H. Phelan, and M. A. Chui, “Applying participatory design to a pharmacy system intervention,” *Research in*

- Social and Administrative Pharmacy, vol. 15, no. 11, pp. 1358–1367, Nov. 2019, doi: 10.1016/j.sapharm.2018.11.012.
- [57] D. Baumer, W. Bischofberger, H. Lichter, and H. Zullighoven, “User interface prototyping-concepts, tools, and experience,” in Proceedings of IEEE 18th International Conference on Software Engineering, IEEE Comput. Soc. Press, 1995, pp. 532–541. doi: 10.1109/ICSE.1996.493447.
- [58] A. C. Oran, G. Santos, B. Gadelha, and T. Conte, “A framework for evaluating and improving requirements specifications based on the developers and testers perspective,” *Requir Eng*, vol. 26, no. 4, pp. 481–508, Dec. 2021, doi: 10.1007/s00766-021-00352-6.
- [59] M. Walker, L. Takayama, and J. A. Landay, “High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes when Testing Web Prototypes,” Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 46, no. 5, pp. 661–665, Sep. 2002, doi: 10.1177/154193120204600513.
- [60] “What is ChatGPT? | OpenAI Help Center.” Accessed: Aug. 17, 2023. [Online]. Available: <https://help.openai.com/en/articles/6783457-what-is-chatgpt>
- [61] “Preguntas frecuentes sobre Bard.” Accessed: Aug. 17, 2023. [Online]. Available: <https://bard.google.com/faq>
- [62] “Tu copiloto con inteligencia artificial para la Web.” Accessed: Aug. 17, 2023. [Online]. Available: <https://www.microsoft.com/es-es/bing?form=MA13FV>
- [63] U. S. Senarath, “Waterfall methodology, prototyping and agile development,” *Tech. Rep.*, pp. 1–16, 2021.
- [64] “Genially, la herramienta online para crear contenido interactivo.” Accessed: Mar. 13, 2024. [Online]. Available: <https://genial.ly/es/>
- [65] “Phaser - A fast, fun and free open source HTML5 game framework - Phaser.” Accessed: Mar. 13, 2024. [Online]. Available: <https://phaser.io/>
- [66] S. C. Peres, T. Pham, and R. Phillips, “Validation of the System Usability Scale (SUS),” Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 57, no. 1, pp. 192–196, Sep. 2013, doi: 10.1177/1541931213571043.

- [67] C. Putnam, M. Puthenmadom, M. A. Cuervo, W. Wang, and N. Paul, "Adaptation of the System Usability Scale for User Testing with Children," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, Apr. 2020, pp. 1–7. doi: 10.1145/3334480.3382840.
- [68] J. Cruz Ripoll, "Referencias de velocidad lectora." Accessed: Apr. 26, 2024. [Online]. Available: <https://www.intralineas.com/blog/referencias-de-velocidad-lectora>
- [69] J. C. Ripoll Salceda, M. M. Tapia Montesinos, and G. Aguado Alonso, "Velocidad lectora en alumnado hispanohablante: un metaanálisis," *Revista de Psicodidáctica*, vol. 25, no. 2, pp. 158–165, Jul. 2020, doi: 10.1016/j.psicod.2020.01.002.
- [70] J. C. Ripoll Salceda, D. S. Zevallos Polo, and N. P. Arcos, "La concepción simple de la lectura en alumnos de 4º de primaria de una escuela fiscal de Quito," *Alteridad*, vol. 12, no. 1, p. 115, Apr. 2017, doi: 10.17163/alt.v12n1.2017.10.
- [71] "Measuring Usability with the System Usability Scale (SUS) – MeasuringU." Accessed: May 06, 2024. [Online]. Available: <https://measuringu.com/sus/>

## **CAPÍTULO VII**

### **ANEXOS**

## **7.1. Anexo 1: Invitación a participar en entrevistas iniciales**

Estimado/a Ing. [Nombre del participante],

Se dirige a usted Diana Avilés, estudiante de Ingeniería en Software de la Universidad Técnica Estatal de Quevedo, quien se encuentra llevando a cabo su proyecto de titulación denominado "Aplicación web para incentivar el aprendizaje de fundamentos de programación de estudiantes de educación básica media-superior.", bajo la dirección del Dr. Orlando Erazo. Teniendo en cuenta que usted tiene experiencia en la rama de sistemas y docencia, el motivo de mi mensaje es solicitar su amable colaboración brindando una opinión para el desarrollo de mi propuesta. Además, me gustaría contar con la participación de su hijo/a, quien, según tengo entendido, posee conocimientos previos en el uso de Scratch y se encuentra en edad de educación básica media-superior.

La propuesta en cuestión se centra en la creación de una aplicación web diseñada para fomentar el aprendizaje de los fundamentos de la programación a través de un juego serio inspirado en el concepto de las salas de escape. Por lo tanto, si usted pudiera colaborar conmigo, lo que se requiere es realizar una entrevista mediante videoconferencia para socializar y discutir ideas a incluir en el software a manera de requerimientos de usuario.

Deseo asegurarme de que esta entrevista se ajuste perfectamente a sus agenda y preferencias. Por ello, le invito a que seleccione la fecha y hora que les resulte más conveniente y me notifique por este medio.

Agradezco de antemano su tiempo y disposición para contribuir al éxito de este proyecto.

Atentamente,

Diana Avilés

Estudiante de Ingeniería en Software

Universidad Técnica Estatal de Quevedo

## **7.2. Anexo 2: Preguntas abiertas de la entrevista no estructurada a docentes.**

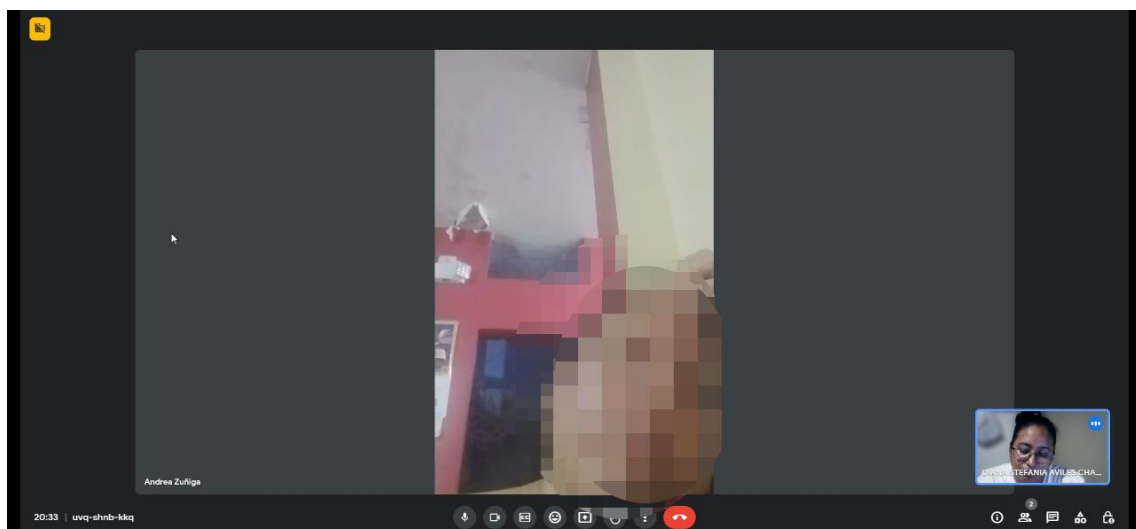
- ¿Tiene experiencia previa en la integración de juegos educativos en su enseñanza?  
¿Puede compartir alguna experiencia al respecto?
- ¿Qué opinión tiene sobre la idea de incorporar un juego de una sala de escape educativa para enseñar programación?
- ¿Cuáles cree que son los beneficios potenciales de utilizar un juego de sala de escape en la enseñanza de la programación en comparación con métodos más tradicionales?
- ¿Cómo cree que el juego de sala de escape podría complementar sus objetivos educativos y planes de estudio?
- ¿Qué nivel de dificultad considera apropiado para el juego en función del nivel de los estudiantes a los que se dirigirá?
- ¿Cuáles son los temas o conceptos específicos de programación que considera que se aborden en el juego?
- ¿Cómo visualiza la participación y el seguimiento de los estudiantes en relación con el juego? ¿Debería ser una tarea asignada o una actividad opcional?
- ¿Considera importante la colaboración entre los estudiantes en el juego? ¿Cómo podría fomentar esta colaboración?
- ¿Tiene alguna preocupación o desafío específico en mente que podría surgir al utilizar un juego de sala de escape educativo en su clase?

### 7.3. Anexo 3: Preguntas abiertas de la entrevista no estructurada a estudiantes.

- ¿Has jugado a algún juego de sala de escape antes? ¿Qué te gusta de ellos?
- ¿Te gustaría aprender programación a través de un juego de sala de escape? ¿Por qué sí o por qué no?
- ¿Cómo te gustaría que el juego te enseñara sobre programación? ¿Quieres resolver acertijos, escribir código, completar palabras o hacer algo más?
- ¿Qué cosas te gustaría encontrar en un juego de sala de escape sobre programación?
- ¿Cómo te gustaría que el juego te enseñara sobre programación? ¿Quieres resolver acertijos, escribir código o hacer algo más?
- ¿Te gustaría que el juego fuera individual o que pudieras jugar en equipo con amigos?
- ¿Qué tipo de desafíos te gustaría encontrar en el juego? ¿Tienes alguna idea específica?
- ¿Qué tipo de personajes o historias crees que podrían hacer que el juego sea más divertido?
- ¿Hay algo que no te gustaría que estuviera en el juego o que te preocupe?
- ¿Qué te motivaría a seguir jugando y aprender más sobre programación a medida que avanzas en el juego?

### 7.4. Anexo 4: Evidencia de entrevista

#### Anexo 1 Evidencia de entrevista inicial



FUENTE: AUTOR

## 7.5. Anexo 5: Preguntas realizadas a ChatGPT

### Anexo 2 Preguntas realizadas a ChatGPT

Aspectos	Características
<b>Temas</b>	<ul style="list-style-type: none"><li>• ¿Cuáles son algunos temas educativos relacionados con la programación que podrían abordarse en el juego?</li><li>• ¿Se podrían incorporar conceptos de algoritmos y resolución de problemas en el juego? ¿Cómo?</li><li>• ¿Cuáles son los temas de programación más importantes para los niños de educación básica media-superior que deberían cubrirse en el juego?</li></ul>
<b>Desafíos</b>	<ul style="list-style-type: none"><li>• ¿Cómo se podría aumentar gradualmente la dificultad de los desafíos para asegurar un aprendizaje progresivo?</li></ul>
<b>Mecánicas de juego</b>	<ul style="list-style-type: none"><li>• ¿Qué mecánicas de juego específicas podrían utilizarse para enseñar programación de manera efectiva?</li><li>• ¿Cómo podría el juego recompensar el progreso en la programación y el logro de objetivos?</li><li>• ¿Se podrían incorporar mecánicas de colaboración entre jugadores para fomentar el aprendizaje en equipo?</li></ul>
<b>Actividades relacionadas con la programación</b>	<ul style="list-style-type: none"><li>• ¿Qué ejercicios de codificación se pueden incluir en el juego para fortalecer las habilidades de programación de los jugadores?</li><li>• ¿Cómo podría el juego conectar el aprendizaje de la programación con situaciones del mundo real o escenarios interesantes para los niños?</li><li>• ¿Se podrían incluir tutoriales interactivos o pistas en el juego para ayudar a los jugadores cuando se enfrenten a desafíos difíciles?</li></ul>

FUENTE: AUTOR

## 7.6. Anexo 6: Consentimiento de participación y la toma de imágenes

### Anexo 3 Consentimiento de participación

#### CONSENTIMIENTO PARA PARTICIPACIÓN Y LA TOMA DE IMÁGENES

Nombre del estudiante:

Fecha:

De acuerdo con lo estipulado en el Registro Oficial Suplemento No. 459 de 26 de mayo de 2021 donde se publicó la Ley Orgánica de Protección de Datos Personales. Esta tiene por objeto garantizar el derecho a la protección de datos personales, que incluye el acceso y decisión sobre la información y datos de este carácter, así como su correspondiente protección. Se procede con el siguiente consentimiento.

El propósito de este estudio es investigar si una aplicación web de sala de escape puede promover eficazmente el aprendizaje de programación básica de los niños. Se busca evaluar cómo la experiencia interactiva de resolución de problemas en una sala de escape virtual influye positivamente en la comprensión y retención de conceptos clave de programación.

#### CONSENTIMIENTO PARA LA TOMA DE IMÁGENES

Por la presente, doy mi consentimiento para que me tomen fotografías. El término "imagen" incluye video o fotografía fija, en formato digital. Además, entiendo y acepto que se mantendrá anónima la identidad, ocultando el rostro y sin revelar nombres, en caso de que las imágenes sean utilizadas con fines didácticos o educativos.

#### PROPÓSITO

Por la presente, autorizo el uso de la(s) imágenes(s) para el propósito de evidencia del proceso de investigación en la Carrera de Software de la Facultad Ciencias de la Ingeniería, que pertenece a la UTEQ. Para la realización de un proyecto de titulación denominado " Aplicación web para incentivar el aprendizaje de fundamentos de programación de estudiantes de educación básica media-superior.", elaborado por Diana Aviles, bajo la dirección del Dr. Orlando Erazo.



Nombre y apellido del representante

*Orlando Erazo*

FUENTE: AUTOR

## 7.7. Anexo 7: Cuestionario SUS

### Anexo 4 Cuestionario SUS

#### Cuestionario para evaluar la experiencia del usuario en la Aplicación Web PEG

Responde las siguientes preguntas usando emojis. 😡 significa que estás muy en desacuerdo, mientras que 😊 significa que estás muy de acuerdo.

N.	Preguntas	😡	😞	😐	😊	😄
1	Me gustaría usar este juego muchas veces.					✓
2	Encontré el juego muy complicado.		✓			
3	Creo que el juego es fácil de usar.				✓	
4	Necesité la ayuda de un adulto para utilizar este juego.				✓	
5	Todas las partes del juego trabajan bien juntas.				✓	✓
6	Existen partes del juego que no funcionaban como esperaba.	✓				
7	Otros niños podrían aprender a jugar este juego rápidamente.				✓	
8	Me sentí muy confundido/a al usar el juego.	✓				
9	Me sentí muy confiado/a utilizando el juego.					✓
10	Creo que necesito aprender muchas cosas antes de poder jugar eficazmente este juego.		✓			

FUENTE: AUTOR

## 7.8. Anexo 8: Participante de evaluación de usabilidad

### Anexo 5 Participante de evaluación de usabilidad



FUENTE: AUTOR

## 7.9. Anexo 9: Planificación académica

### 7.9.1. Nivel 0 : Introducción a python - print()

#### Objetivo

Enseñar a los estudiantes la función print() y cómo utilizarla para mostrar mensajes en la consola de python.

#### Historia

Has aterrizado en un centro espacial abandonado. Tu nave se encuentra dañada y con los sistemas reiniciados. Tu misión es salir de la nave y recorrer este centro espacial, buscando la manera de reparar tu nave.

#### 7.9.1.1. Habitación 1

Dormitorio de la nave donde se encuentra una tarjeta de acceso para abrir la puerta de la habitación 2. La tarjeta de acceso se encuentra en el cajón del escritorio.

#### Pistas

- Indica el lugar donde se encuentra el cajón.
- Dentro del cajón indica donde se encuentra la tarjeta

### 7.9.1.2. Habitación 2

Habitación de almacenamiento donde hay una puerta cerrada

#### Pistas

- Indica donde debe dar clic

### 7.9.1.3. Habitación 3

Habitación principal donde se encuentra una computadora donde estará la información de programación para obtener la clave de la puerta de la habitación 4

#### Pistas

- Indica que debe dar clic en la computadora

#### Programación

Python es un lenguaje de programación que se utiliza para crear todo tipo de aplicaciones, desde videojuegos hasta sitios web.

Supongamos que queremos mostrar el siguiente mensaje: **¡Hola, mundo!**

En Python, esto se puede hacer con el siguiente código: **print("¡Hola, mundo!")**

En el caso de números: **print(15)**

#### Y si queremos hacer una oración con números

- **Primer ejercicio**

Ingresa tu nombre en el siguiente cuadro: **print("Hola, \_\_\_\_\_ !")**

```
>>> print("Hola, [tu_nombre]!")
Hola, [tu_nombre]!
```

- **Segundo ejercicio**

Ordena el código

Aleatoriamente sale uno de los siguientes:

1. **print("Tengo 5 manzanas")**  
[ (" ] [ 5 ] [ print ] [ tengo ] [ " ) ] [ manzanas ]
2. **print("Tengo 3 libros")**

```
[ 3 ] [ print ] [ tengo ] [ " ] [ libros ] [ " ]
3. print("Poseo 4 gatos")
[ " ] [ 4 ] [ " ] [ gatos ] [ print ] [ poseo ]
4. print("Hay 2 lápices")
[ print ] [ hay ] [ lápices ] [ " ] [ 2 ] [ " ]
5. print("Veo 7 estrellas")
[ veo ] [ " ] [ estrellas ] [ " ] [ 7 ] [ print ]
6. print("Comí 6 chocolates")
[ 6 ] [ print ] [ comí ] [ " ] [ " ] [ chocolates ]
```

Felicidades, ¡has aprendido a usar 'print'!

Como recompensa, aquí está el código de la puerta de salida: **código aleatorio**

#### 7.9.1.4. Habitación 4

Ultima habitación se encuentra la puerta de salida donde hay que ingresar el código obtenido en la habitación anterior.

#### Pistas

- Ubicación donde se encuentra la puerta

#### 7.9.2. Nivel 1: Variables

##### Objetivo

Utilizar ejemplos prácticos para enseñar a los estudiantes a declarar y usar variables, facilitando el almacenamiento y manejo de datos en la programación.

##### Historia

Has logrado salir de la nave, ahora buscas herramientas y piezas para reparar tu nave. Caminando encuentras una mina abandonada donde puedes obtener una gema que dará energía a tu nave.

##### 7.9.2.1. Habitación 1

Entrada a la mina donde hay una puerta bloqueada se necesitan dos pistas para poder desbloquear la puerta, la primera se encuentra en una caja donde estará una numeración cirílica que servirá para interpretar la siguiente pista.

Texto de referencia:

Utiliza los valores asignados a cada símbolo para determinar el valor numérico correspondiente a cada letra. Por ejemplo: Si,  $h=B+IO$ . Entonces, el resultado será:  $h=3$

$$a = IO + \Gamma \quad b = И \times Б \quad c = \Gamma \times Б$$

$$d = \Upsilon + Д \quad e = S + IO \quad f = \Phi \times \epsilon$$

$$IO = 1 \quad Б = 3 \quad \Gamma = 3$$

$$Д = 4 \quad \epsilon = 5 \quad S = 6$$

$$\Phi = 7 \quad И = 8 \quad \Upsilon = 9$$

Si el resultado es un número de dos dígitos o más, debes sumar los dígitos de ese número. Ejemplo: Si el resultado es 12, se sumará  $1+2=3$

En la puerta tendrá que colocar el valor según la variable que se encuentra reflejada.

### **Pistas**

- Ubicación de la pista 1
- Ubicación de la pista 2
- Ubicación de la puerta
- Explicación de la puerta

### **7.9.2.2. Habitación 2**

Parte de la mina donde hay tarjetas que se deben recoger del suelo

### **Pistas**

- Indica que debe recoger las tarjetas del suelo, después de recogerlas muestra que debe continuar a la siguiente puerta.

### **7.9.2.3. Habitación 3**

Parte de la mina donde hay tarjetas que se deben recoger del suelo, la puerta se encuentra bloqueada y para desbloquearla debe usar un código de colores según el número que se encuentra en cada tarjeta.

### **Pistas**

- Indica que debe recoger las tarjetas del suelo, después de recogerlas muestra que debe continuar a la siguiente puerta.

#### 7.9.2.4. Habitación 4

Ultima habitación se encuentra la puerta de salida donde debes obtener la gema para la nave y para ello debes ingresar en la computadora que se encuentra en el escritorio y realizar los ejercicios mostrados.

#### Pistas

- Indica que debe recoger las tarjetas del suelo, después de recogerlas muestra que debe continuar a la siguiente puerta.

#### Programación

Hoy vamos a explorar cómo funcionan las variables en el mundo de la programación.

Imagina una **variable** como una caja donde puedes guardar **cosas**.

En programación, estas 'cosas' pueden ser números, texto, o incluso listas de elementos.

Una **variable** es como un apodo donde guardamos **información** que puede cambiar.

Por ejemplo: **numero = 10**

**Producto = “manzana”**

Aquí, **numero** es el nombre de nuestra variable, y **10** es el valor que estamos almacenando en ella. También **producto** es el nombre de nuestra variable y **“manzana”** el valor que estamos almacenando.

Este valor podemos cambiarlo cuando queramos. Por ejemplo, para cambiar **numero** a **15**, hacemos: **numero = 15**

También podemos hacer operaciones **matemáticas** con ellas. Si tenemos dos **variables**:

```
>>> a = 5
>>> b = 2
```

Y las sumamos así:

```
>>> suma = a + b
```

Ahora, ¿cómo mostramos el resultado?

Opción 1: `print( suma )`

Opción 2: `print("suma")`

Resultado:

```
>>> a = 5
>>> b = 2
>>> suma = a + b
>>> print(suma)
7
```

## Ejercicios

- **Completar el código:**

En aleatorio pueden salir 3 de estos ejercicios:

### Resta de variables

```
1. >>> x = 10
2. >>> y = 4
3. >>> resta = x - y
4. >>> print(resta)
5.
```

### Multiplicación de variables

```
1. >>> m = 7
2. >>> n = 3
3. >>> producto = m * n
4. >>> print(producto)
5.
```

### División de variables

```
1. >>> c = 15
2. >>> d = 5
3. >>> division = c / d
4. >>> print(division)
5.
```

### Concatenación

```
1. >>> nombre = "Ana"
2. >>> apellido = "López"
3. >>> nombre_completo = nombre + " " + apellido
4. >>> print(nombre_completo)
5.
```

### Concatenación con variables y texto

```
1. >>> animal = "gato"
2. >>> accion = "corre"
3. >>> mensaje = "El " + animal + " " + accion + " rápidamente."
4. >>> print(mensaje)
5.
```

### **7.9.3. Nivel 2: Tipos de datos**

#### **Objetivo**

Aplicar actividades interactivas para explicar a los estudiantes los distintos tipos de datos, mejorando su capacidad para seleccionar el tipo adecuado para cada situación en la programación.

#### **Historia**

Has obtenido la gema para dar energía a tu nave, pero no sabes cómo utilizarla. En tu camino, encuentras una biblioteca donde puedes obtener información sobre cómo reparar tu nave.

#### **7.9.3.1. Habitación 1**

Recepción de la biblioteca donde hay una computadora, pero no parece tener energía, se debe buscar el generador de energía para activar la computadora de recepción.

Una vez activado se puede ingresar a la computadora donde habrá información sobre print() con los tipos de variables y diferentes ejercicios. Una vez completado se obtiene el manual de reparación de la nave y se puede salir del nivel.

#### **Pistas**

- Ubicación de la puerta de la siguiente habitación
- Ubicación de la computadora en la recepción

#### **7.9.3.2. Habitación 2**

Habitación cerrada donde hay un panel que activa la puerta, en el panel hay un ejercicio de ordenar código para poder activar la puerta

#### **Pistas**

- Ubicación del panel para activar la puerta

#### **7.9.3.3. Habitación 3**

Habitación de cartografía donde hay una computadora con información sobre los tipos de variables, al finalizar de resolver los ejercicios lo que desbloquea el generador.

#### **Pistas**

- Indica donde se debe clicar la computadora

#### 7.9.3.4. Habitación 4

Generador de energía donde se debe dar clic para activar la computadora en la recepción

##### **Pistas**

- Indica donde se debe clicar para activar la energía en la recepción

##### **Programación en habitación de cartografía:**

¿Cuáles son los tipos de datos?

**int:** No tienen decimales, solo números enteros. Los números enteros son como contar los bloques de construcción.

Ejemplo: 8, 3, 45

**float:** Tienen partes que no son enteros, es decir, tienen decimales. Los números decimales son como medir el agua en una jarra.

Ejemplo: 8.5

**str:** En Python, el texto se llama "cadena de caracteres" y siempre va entre comillas. Cuando queremos mostrar un mensaje en la pantalla.

Ejemplo: "Hola mundo!", "1234!", "Agua"

**bool:** En Python, usamos True (Verdadero) cuando el interruptor está encendido, y False (Falso) cuando está apagado. Este tipo de dato es como un interruptor de luz: solo puede estar encendido o apagado.

Ejemplo: ¿Es un gato? Verdadero, ¿Es un perro? Falso.

- **Primer ejercicio:**

**Unir con línea:** Unir los tipos de variable con su ejemplo

- **Segundo ejercicio**

**Verdadero o falso:** En aleatorio pueden salir 2 de estas preguntas.

1. Las variables de tipo **int** pueden almacenar números con decimales. (Falso)
2. Una variable de tipo **str** puede contener letras, números y otros caracteres. (Verdadero)
3. Una variable de tipo **float** se utiliza para almacenar valores numéricos enteros. (Falso)

4. Las variables de tipo **int** pueden almacenar números negativos. (Verdadero)
5. Una variable de tipo **bool** solo puede tener uno de dos valores: True o False. (Verdadero)
6. Sumar dos variables de tipo **int** siempre resultará en una variable de tipo **int**. (Verdadero)
7. Concatenar dos variables de tipo **str** dará como resultado una nueva **cadena** que une los dos **str** originales. (Verdadero).

### Programación en la recepción:

print() con tipos de datos

```
>>> nombre = "Juan" #es un str
>>> edad = 25 #es un int
```

Si queremos usar print() para mostrar el valor de nuestras variables usamos:

```
>>> print("Mi nombre es: ", nombre)
Mi nombre es: Juan
>>> print("Mi edad es: ", edad)
Mi edad es: 25
```

¿Pero qué pasa si queremos mostrar una oración como esta?

"Hola, Juan. Tienes 25 años."

```
>>> print("Hola, " + nombre + ". Tienes " + str(edad) + " años.")
Hola, Juan. Tienes 25 años.
```

Utilizamos str(edad) para convertir el valor de la variable edad (que es un número entero) en una cadena de texto antes de añadirlo a la oración.

- **Primer ejercicio**

**Completar el código:**

```
1. >>> # Definición de variables
2. >>> producto = "manzanas"
3. >>> cantidad = 5
4. >>> print("Hay" + ..... + producto +
"En el árbol")
```

Opción 1: str(cantidad)

Opción 2: cantidad

- **Segundo ejercicio**

## Ordenar el código:

Ordenar el orden de las líneas de código, puede salir en aleatorio dos de los siguientes ejercicios:

### Ejemplo con str y int:

```
1. >>> # Definición de variables
2. >>> fruta = "peras"
3. >>> numero = 10
4. >>> print("El mercado tiene " + str(numero) + " " + fruta +
" en venta.")
```

### Ejemplo con str, int, y float:

```
1. >>> # Definición de variables
2. >>> articulo = "libros"
3. >>> total_articulos = 3
5. >>> print("Compré " + str(total_articulos) + " " + articulo + ")")
6.
```

### Ejemplo con str y bool:

```
1. >>> # Definición de variables
2. >>> producto = "zapatos"
3. >>> disponibilidad = True
4. >>> print("Los " + producto + " que buscas están disponibles: " + str(disponibilidad) +
".")
5.
```

### Ejemplo con str y int:

```
1. # Definición de variables
2. vehiculo = "bicicletas"
3. cantidad_en_tienda = 8
4. print("Actualmente, tenemos " + str(cantidad_en_tienda) + " " + vehiculo + " en
stock.")
5.
```

## 7.10. Anexo 10: Preguntas prueba de conocimiento

---

**Tema:** `print()` en Python

---

**Pregunta** ¿Cuál de las siguientes opciones muestra correctamente cómo usar la función `print()` para mostrar el mensaje "Hola mundo" en Python?

- Opciones**
- a) `print(Hola mundo)`
  - b) `print("Hola mundo")`
  - c) `print(Hola_mundo)`
  - d) `print['Hola mundo']`

**Respuesta correcta** B

---

---

**Tema:** Variables

---

**Pregunta** ¿Cuál de las siguientes líneas de código asigna correctamente el número 10 a una variable llamada `edad`?

- Opciones**
- a) `edad == 10`
  - b) `10 = edad`
  - c) `edad: 10`
  - d) `edad = 10`

**Respuesta correcta** D

---

---

**Tema:** Variables

---

**Pregunta** ¿Cuál de las siguientes líneas de código asigna correctamente el número 10 a una variable llamada `edad`?

---

- 
- Opciones**
- a) edad == 10
  - b) 10 = edad
  - c) edad: 10
  - d) edad = 10

**Respuesta correcta** D

---

---

**Tema:** Tipos de Datos

---

**Pregunta** ¿Qué tipo de dato se usaría para almacenar el mensaje "¡Bienvenido al juego!" en Python?

- Opciones**
- a) int
  - b) boolean
  - c) str
  - d) float

**Respuesta correcta** C

---

---

**Tema:** Imprimir datos

---

**Pregunta** ¿Qué imprime el siguiente código en Python?

```
>>> edad = 12
>>> print(edad)
```

- Opciones**
- a) edad
  - b) 12
  - c) Error
  - d) Nada, porque falta comillas

**Respuesta correcta** B

---