

Universidad Católica de Santa María
Facultad de Ciencias e Ingenierías Físicas y Formales
Escuela Profesional de Ingeniería de Sistemas



**“SOFTWARE EDUCATIVO ENFOCADO EN LA
ENSEÑANZA DE CONCEPTOS DE PROGRAMACIÓN
MEDIANTE TÉCNICAS DE GAMIFICATION”**

Tesis presentada por el Bachiller:

Iqira Becerra, Diego Alonso

Para optar el Título Profesional de

**Ingeniero de Sistemas: Especialidad en Ingeniería de
Software**

Asesor:

Mg. Montesinos Murillo, Ángel Felipe

Arequipa- Perú

2019

FACULTAD DE CIENCIAS E INGENIERIAS FISICAS Y FORMALES
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

INFORME DICTAMEN DE BORRADOR TESIS

VISTO

El Borrador de TESIS titulado:

"Software Educativo Enfocado en la Enseñanza de
Conceptos de Programación Mediante Técnicas de
Gamificación"

Presentado por (el) (la) (los) Bachiller (es):

Iqura Becerra, Diego Alonso

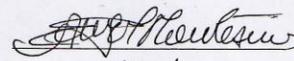
Nuestro dictamen es:

PROCEDENTE

OBSERVACIONES:

Arequipa, 19 de OCTUBRE de 2018


HECTOR VELARDE B.
1064


1631

PRESENTACIÓN

Señor Decano de la Facultad de Ciencias e Ingenierías Físicas y Formales.

Señor Director de la Escuela Profesional de Ingeniería de Sistemas.

Señores Miembros del Jurado Dictaminador de la Tesis

De conformidad con las disposiciones del Reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el presente trabajo de investigación titulado:

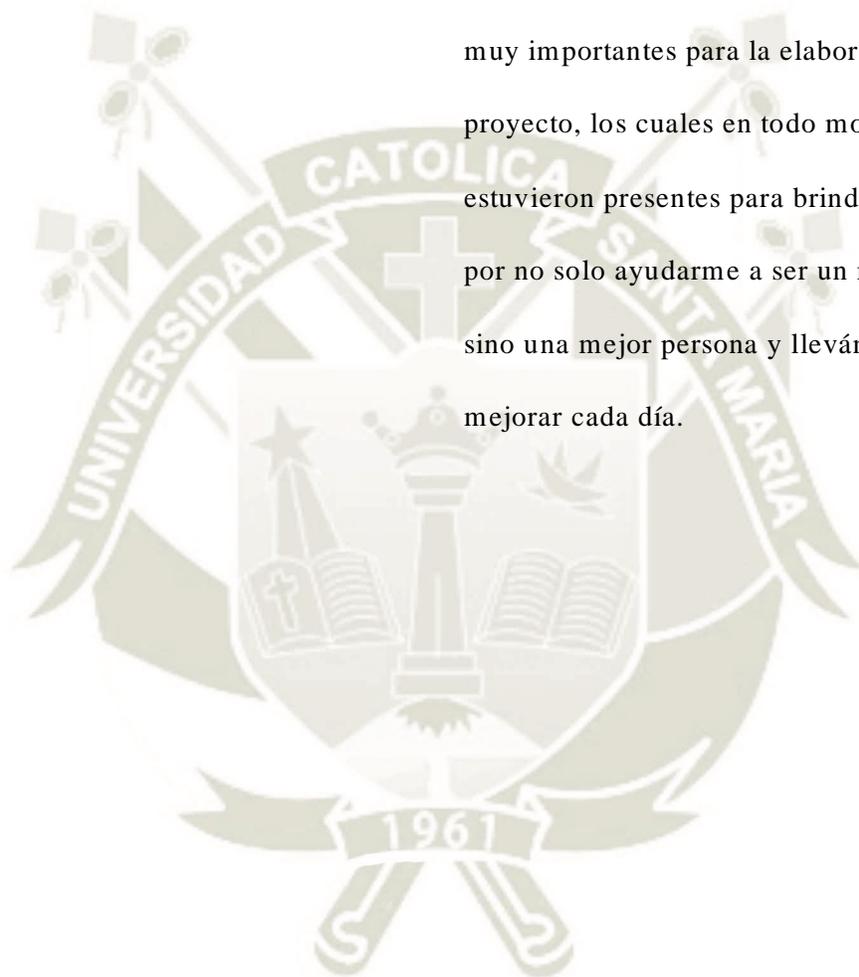
SOFTWARE EDUCATIVO ENFOCADO EN LA ENSEÑANZA DE CONCEPTOS DE PROGRAMACIÓN MEDIANTE TÉCNICAS DE GAMIFICATION

El trabajo de investigación fue realizado aplicando los conocimientos adquiridos durante mi formación universitaria, el mismo que al ser aprobado me permitirá optar por el Título Profesional de Ingeniero de Sistemas

Iquira Becerra Diego Alonso

DEDICATORIA

Este trabajo está dedicado a mi familia, mis amigos y mis profesores, que fueron personas muy importantes para la elaboración de este proyecto, los cuales en todo momento estuvieron presentes para brindarme su apoyo, por no solo ayudarme a ser un mejor profesional sino una mejor persona y llevándome siempre a mejorar cada día.



AGRADECIMIENTOS

Mi agradecimiento a la Universidad Católica de Santa María de Arequipa, institución en la cual estudié y me brindó la oportunidad de desarrollarme tanto profesional como personalmente en la carrera de ingeniería de sistemas.

A mi asesor por brindarme sus conocimientos e inculcarme valores a la investigación y desarrollo profesional.

Agradezco a mis profesores de la carrera de Ingeniería de Sistemas los cuales me ayudaron a esforzarme y a motivarme durante todo mi proceso de aprendizaje.

Quisiera también agradecer a mis amigos de promoción a quienes estimo y con quienes pude compartir estos 5 años llenos de diversas experiencias, ellos me inspiraron a crecer tanto como profesional y como persona.

Por último, quisiera agradecer a mi familia por haberme apoyado en todos estos años y sin ellos no hubiera podido haber cumplido mis metas.

INTRODUCCIÓN

La presente tesis se enfoca en el diseño, desarrollo y evaluación de un software educativo enfocado en la enseñanza de conceptos básicos de programación mediante el uso de técnicas de gamification, con lo cual se podrá dar las pautas para el desarrollo de aplicaciones similares que permitan mejorar la motivación de los estudiantes con respecto a los cursos que deben aprender.

Se ha realizado un trabajo de investigación sobre el diseño y desarrollo de los software educativos, con el objetivo de poder determinar la importancia de la tecnología en la educación, como complemento al alumno en su proceso educativo.

Se determinó la creación de un software educativo para dispositivos Android, debido a que los celulares se han vuelto un dispositivo común para las personas, por tal razón existe una mayor disponibilidad de dichos dispositivos por parte de los alumnos, adicionalmente el uso de la tecnología como apoyo se ha vuelto una necesidad para los educadores ya que la educación se debe adaptar a las nuevas tecnologías.

La presente tesis cuenta con cuatro capítulos:

En el capítulo 1 se hace una descripción general del proyecto, donde se ve: los objetivos de la tesis, alcances y limitaciones, los fundamentos teóricos de las tesis, estado del arte, técnicas, herramientas y aspectos relevantes del desarrollo.

El capítulo 2 se enfoca en la documentación técnica como: plan del proyecto, especificación del requisito de software, especificación de diseño, documentación técnica de programación, pruebas de ejecución y manuales de usuario.

En el capítulo 3 se realiza las conclusiones del software desarrollado.

En el capítulo 4 se ve las recomendaciones basándonos en los resultados de la tesis.

RESUMEN

La presente tesis se enfoca en investigar las diferentes herramientas para el desarrollo de un software educativo para plataformas móviles, el cual utiliza técnicas de gamification con el objetivo de motivar a los estudiantes a aprender sobre la programación.

Primero se realizó un plan del proyecto informático en el cual se determina el tiempo y la viabilidad del proyecto, después se realizaron las especificaciones del requisito del software con lo cual se pudo determinar el alcance del proyecto, las especificaciones funcionales de la aplicación y los diferentes requisitos.

Otro punto importante fue las especificaciones de diseño donde se realizaron tanto la vista lógica como física del sistema, se describe el comportamiento del sistema y la interfaz del usuario.

Por otro lado para realizar la implementación del software se ha descrito la documentación técnica de la programación donde se describe el hardware, la codificación utilizada y la implementación de la aplicación,

Finalmente se ha realizado las pruebas de ejecución las cuales fueron: pruebas funcionales, de usabilidad y aceptación, se realizó los manuales de usuarios que describen el funcionamiento del sistema y se realizó pruebas con alumnos universitarios.

Palabras Clave: Gamification, Ingeniería de Software, Aprendizaje Móvil

ABSTRACT

In this thesis an investigation was made on the different tools for the development of educational software for mobile platform, which uses gamification techniques with the objective of motivate students to learn about programming.

First a computer project plan was carried out in which the time and viability of the project was determined, then the specifications of the software requirement were made, with which it was possible to determine the scope of the project, the functional specifications of the application and the different requirements.

Another important point was the design specifications where both the logical and physical view of the system was made, the behavior of the system and the user interface are described.

On the other hand, in order to implement the software, the technical documentation of the programming describing the hardware, the coding used and the implementation of the application have been described.

Finally, the execution tests were carried out, which were: functional, usability and acceptance tests, the user manuals that describe the operation of the system were made and tests were carried out with college students.

Keywords: Gamification, Software Engineering, Mobile Learning

TABLA DE CONTENIDOS

1. DESCRIPCIÓN DEL PROYECTO	1
1.1. Objetivos	1
1.1.1. Objetivo General	1
1.1.2. Objetivos Específicos	1
1.2. Alcances y Limitaciones.....	1
1.2.1. Alcances.....	1
1.2.2. Limitaciones.....	2
1.3. Fundamentos Teóricos	2
1.3.1. Estado del arte	2
1.3.2. Bases Teóricas del proyecto.....	6
1.4 Técnicas y Herramientas	8
1.4.1. Técnicas	8
1.4.2. Herramientas	10
1.5 Aspectos Relevantes del Desarrollo	11
2. DOCUMENTACIÓN TÉCNICA	17
2.1. Plan del Proyecto Informático	17
2.1.1. Planificación temporal del proyecto	17
2.1.2. Estudio de viabilidad del proyecto	18
2.1.3. Historias de usuario y criterios de aceptación	21
2.2 Especificación de requisitos del software	25
2.2.1 Propósito:	25

2.2.2 Alcance del proyecto	25
2.2.3 Glosario	26
2.2.4 Entorno del sistema	26
2.2.5 Especificación de requisitos funcionales	26
2.2.6 Características del usuario	33
2.2.7 Requisitos de interfaz externa	33
2.2.8 Requerimientos funcionales	34
2.2.9 Requerimientos no funcionales	38
2.2.10 Estructura Lógica de los Datos	41
2.3 Especificación de diseño	43
2.3.1 Vista Lógica del Sistema	43
2.3.2 Vista Física del Sistema	45
2.3.3 Vista de Procesos del Sistema	45
2.3.4 Comportamiento del Sistema	47
2.3.5 Persistencia	49
2.3.6 Interfaz de Usuario	51
2.4 Documentación técnica de programación	53
2.4.1 Hardware	53
2.4.2 Motor de videojuegos	54
2.4.3 Codificación o scripts	56
2.4.3 Entorno de desarrollo	57
2.4.4 Implementación de la aplicación	58
2.5 Pruebas de Ejecución	69

2.5.1 Pruebas funcionales	69
2.5.2 Pruebas de Usabilidad	70
2.5.3 Pruebas de Aceptación	72
2.6 Manuales de Usuario	74
2.6.1 Objetivo	74
2.6.2 Requisitos e instalación	75
2.6.3 Descripción del funcionamiento del sistema	75
2.7 Validación de la aplicación	87
2.7.1 Caso de Estudio	87
3. CONCLUSIONES	90
4. RECOMENDACIONES Y TRABAJOS FUTUROS	92
4.1. Recomendaciones	92
4.2. Trabajos Futuros	92
5. REFERENCIAS BIBLIOGRÁFICAS	94
Anexo A: Formato de pre-test y post-test	98
Anexo B: Formato de encuesta de heurística de usabilidad	100
Anexo C: Guía para la encuesta de usabilidad	101
Anexo D: Plan de Tesis	102

LISTA DE TABLAS

Tabla 1. Tabla del plan de trabajo del desarrollo del software (Fuente: Elaboración Propia).	19
Tabla 2. Tabla de las historias de usuarios (Fuente: Elaboración Propia)	21
Tabla 3. Tabla del glosario del requisito de software (Fuente: Elaboración Propia)	26
Tabla 4. Tabla del requisito funcional agregar perfil (Fuente: Elaboración Propia)	34
Tabla 5. Tabla del requisito funcional seleccionar actividad (Fuente: Elaboración Propia)	34
Tabla 6. Tabla del requisito funcional ver perfil (Fuente: Elaboración Propia)	35
Tabla 7. Tabla del requisito funcional actividad pregunta (Fuente: Elaboración Propia)	35
Tabla 8. Tabla del requisito funcional actividad verdadero o falso (Fuente: Elaboración Propia)	36
Tabla 9. Tabla del requisito funcional actividad conectar conceptos (Fuente: Elaboración Propia)	37
Tabla 10. Tabla de la estructura lógica de datos del alumno (Fuente: Elaboración Propia)	41
Tabla 11. Tabla de la estructura lógica de la actividad pregunta (Fuente: Elaboración Propia)...	42
Tabla 12. Tabla de la estructura lógica de la actividad verdadero o falso (Fuente: Elaboración Propia)	42
Tabla 13. Tabla de la estructura lógica de la actividad conectar conceptos (Fuente: Elaboración Propia)	42
Tabla 14. Tabla de verificación de la caja negra (Fuente: Elaboración Propia)	69
Tabla 15. Tabla de pruebas de usabilidad (Fuente: Elaboración Propia)	71
Tabla 16. Tabla de pruebas de aceptación (Fuente: Elaboración Propia)	72

LISTA DE FIGURAS

Figura 1. Iconos de las actividades (Fuente: Elaboración Propia)	12
Figura 2. Icono del nivel de jugador (Fuente: Elaboración Propia)	12
Figura 3. Interfaz de Duolingo (Fuente: Duolingo)	16
Figura 4. Entorno del sistema (Fuente: Elaboración Propia).....	26
Figura 5. Casos de uso del estudiante (Fuente: Elaboración Propia)	27
Figura 6. Casos de uso de agregar perfil (Fuente: Elaboración Propia)	27
Figura 7. Casos de uso de seleccionar actividad (Fuente: Elaboración Propia)	28
Figura 8. Casos de uso de ver perfil (Fuente: Elaboración Propia)	29
Figura 9. Casos de uso de la actividad pregunta (Fuente: Elaboración Propia)	30
Figura 10. Casos de uso de la actividad verdadero o falso (Fuente: Elaboración Propia)	31
Figura 11. Casos de uso de la actividad de conectar conceptos (Fuente: Elaboración Propia)	32
Figura 12. Vista lógica de la aplicación (Fuente: Elaboración Propia)	44
Figura 13. Vista de despliegue de la aplicación (Fuente: Elaboración Propia)	45
Figura 14. Diagrama de actividad ingreso de perfil (Fuente: Elaboración Propia).....	46
Figura 15. Diagrama de ver perfil (Fuente: Elaboración Propia)	46
Figura 16. Diagrama de ver logros (Fuente: Elaboración Propia)	46
Figura 17. Diagrama de seleccionar actividad (Fuente: Elaboración Propia)	46
Figura 18. Diagrama de actividad pregunta (Fuente: Elaboración Propia)	46
Figura 19. Diagrama de actividad verdadero o falso (Fuente: Elaboración Propia)	47
Figura 20. Diagrama de actividad conectar conceptos (Fuente: Elaboración Propia)	47
Figura 21. Diagrama de Clases (Fuente: Elaboración Propia)	48

Figura 22. Diagrama de Secuencia (Fuente: Elaboración Propia)	49
Figura 23. Modelo de Tablas (Fuente: Elaboración Propia)	51
Figura 24. Diagrama de Navegabilidad (Fuente: Elaboración Propia)	52
Figura 25. Descargar Unity (Fuente: Elaboración Propia)	58
Figura 26. Instalación de Unity (Fuente: Elaboración Propia)	58
Figura 27. Descargar SDK (Fuente: Elaboración Propia)	59
Figura 28. Android SDK Manager (Fuente: Elaboración Propia)	59
Figura 29. Descargar JDK (Fuente: Elaboración Propia).....	60
Figura 30. Preferencias de Unity – Herramientas Externas (Fuente: Elaboración Propia)	60
Figura 31. Asset Store (Fuente: Elaboración Propia)	61
Figura 32. Creación de nuevo proyecto en Unity (Fuente: Elaboración Propia)	62
Figura 33. Interfaz de Unity (Fuente: Elaboración Propia)	63
Figura 34. Interfaz de Gimp (Fuente: Elaboración Propia)	64
Figura 35. Interfaz de Audacity (Fuente: Elaboración Propia).....	64
Figura 36. Creación de un script (Fuente: Elaboración Propia)	65
Figura 37. Ventana de Build Settings (Fuente: Elaboración Propia)	66
Figura 38. Interfaz de MonoDevelop (Fuente: Elaboración Propia).....	67
Figura 39. Ventana de Player Settings (Fuente: Elaboración Propia).....	68
Figura 40. Instalación de la aplicación (Fuente: Elaboración Propia)	76
Figura 41. Pantalla principal del ingreso de datos del perfil (Fuente: Elaboración Propia)	76
Figura 42. Ingreso de nombre y edad (Fuente: Elaboración Propia)	77
Figura 43. Selección de la orientación sexual (Fuente: Elaboración Propia)	77
Figura 44. Menú principal de la aplicación y Perfil del alumno (Fuente: Elaboración Propia) ...	78

Figura 45. Logros del alumno y Descripción de la actividad preguntas (Fuente: Elaboración Propia)	79
Figura 46. Pantalla de cargando para el cambio de escena e Inicio de la actividad de pregunta (Fuente: Elaboración Propia)	80
Figura 47. Pantalla para salir de la actividad y Actividad de pregunta (Fuente: Elaboración Propia)	81
Figura 48. Respuesta correcta e incorrecta de la actividad pregunta (Fuente: Elaboración Propia)	82
Figura 49. Mensaje de logro desbloqueado y Salir de la actividad (Fuente: Elaboración Propia)	82
Figura 50. Inicio de la actividad de verdadero o falso (Fuente: Elaboración Propia)	83
Figura 51. Respuesta correcta e incorrecta de la actividad verdadero o falso (Fuente: Elaboración Propia)	84
Figura 52. Inicio de la actividad de conectar conceptos (Fuente: Elaboración Propia)	85
Figura 53. Arrastrar datos en la actividad conectar conceptos (Fuente: Elaboración Propia)	85
Figura 54. Respuesta correcta e incorrecta de la actividad conectar conceptos (Fuente: Elaboración Propia)	86
Figura 55. Completar ejercicio de la actividad conectar conceptos (Fuente: Elaboración Propia)	87
Figura 56. Resultados del Pre-test (Fuente: Elaboración Propia)	88
Figura 57. Resultados del Pos-test (Fuente: Elaboración Propia)	89
Figura 58. Resultados de la evaluación heurística (Fuente: Elaboración Propia)	89

1. DESCRIPCIÓN DEL PROYECTO

1.1. Objetivos

1.1.1. Objetivo General

Desarrollar un software educativo enfocado en la enseñanza de conceptos de programación mediante técnicas de gamification.

1.1.2. Objetivos Específicos

- a. Escoger una metodología de programación y técnicas de gamification para el desarrollo de la aplicación.
- b. Realizar la especificación de los requisitos de la aplicación y establecer una especificación del diseño.
- c. Desarrollar la aplicación.
- d. Generar pruebas de la ejecución de la aplicación y escribir un manual del usuario.

1.2. Alcances y Limitaciones

1.2.1. Alcances

- a. La presente tesis explorará el desarrollo y uso de un software educativo que utilice técnicas de gamification para la enseñanza de conceptos de programación.
- b. La investigación abarca únicamente a los estudiantes de los primeros años de ingeniería de sistemas, los cuales serán evaluados mediante un caso de estudio.

1.2.2. Limitaciones

- a. Las pruebas solo se realizaron una vez por lo cual no se podría evaluar la retención de conocimiento a largo plazo.
- b. Los ejercicios y temas de la aplicación se basan en los libros que usan los estudiantes.

1.3. Fundamentos Teóricos

1.3.1. Estado del arte

La educación es un concepto que va cambiando y adaptándose, donde en un principio pasa de ser un modelo basado en la transmisión y observación, a un modelo activo y participativo, con lo cual se puede establecer nuevas estrategias en el aprendizaje.

Pero para lograr dicha participación por parte de los estudiantes es necesario el uso de una metodología que permita crear experiencias educativas concretas y vivenciales.

Por ejemplo las actividades lúdicas permiten que los alumnos generen un potencial heurístico, el cual permite mejorar las capacidades tanto de análisis como de síntesis, debido a que dan una mayor libertad al estudiante.

En el contexto educativo de las carreras científicas existe una forma de enseñanza mediante la realización de ejercicios y trabajos, ya que permiten mejorar el aprendizaje de los estudiantes, facilitando la relación de lo aprendido en forma teórica con lo aprendido de manera práctica.

Un problema que se ha identificado en la educación actual son las limitaciones de materiales y la falta de recursos humanos que hacen que sea difícil aprender solo mediante ejercicios prácticos en el aula (Ballu et al., 2016).

Debido a esto, los software educativos son las herramientas educativas de la nueva sociedad del aprendizaje (Fredes, Hernández, & Díaz, 2012), por lo cual se han adoptado estrategias a nivel institucional y de aula para que los alumnos usen adecuadamente dichos software educativos.

Lamentablemente las lecciones realizadas en los salones de clases no suelen ser apreciadas por los alumnos, lo cual genera que en la realización de clases, el aprendizaje y la motivación del estudiante y del profesor se vean afectados por estímulos externos o por una falta de atención e interés. (Ballu et al., 2016).

Esto explicaría las razones porque se usa la gamification en la educación, ya que permite motivar a los estudiantes a aprender mediante elementos de juegos en entornos educativos con la meta de maximizar el entretenimiento y el compromiso del estudiante e inspirarlos a continuar aprendiendo (Dicheva, Dichev, Agre, & Angelova, 2015).

El uso de la tecnología en la educación se ha vuelto una necesidad en la actualidad, ya que permite brindar un apoyo al estudiante en su proceso educativo, por ejemplo el uso de la gamification permite mejorar el proceso de diferentes tareas, al convertirlas en acciones más lúdicas, motivando a los usuarios al realizar su trabajo.

La presente tesis busca el desarrollo de un software educativo con la intención de enseñar conceptos básicos de programación mediante técnicas como la *gamification*, con el objetivo de mejorar el proceso educativo de cursos enfocados en la informática.

La tesis implementación de un laboratorio virtual inmersivo aplicado a la enseñanza de física usando técnicas de gamification (Iquirá Becerra, 2018) que presente para obtener el título de maestro sirvió de inspiración para el presente trabajo, ya que sirvió como base para el desarrollo de una aplicación enfocada a la enseñanza de programación que brinde un modelo que dé mayor libertad a los profesores para agregar información.

La gamification busca usar conceptos de videojuegos en ambientes no lúdicos, permitiendo mejorar la motivación de los usuarios, es por esta razón que puede ser usado en diferentes campos como en la educación donde sirve como un complemento a los métodos tradicionales.

Se revisó investigaciones relacionadas con el de la gamification en la educación, donde se buscaba caso de estudios sobre su uso e impacto en los estudiantes.

Jill Laster (Laster, 2010) decidió cambiar el formato del curso volviéndolo un juego, donde el tiempo de clases se usaba para completar misiones, pelear contra enemigos (tomar exámenes o cuestionarios) y crear equipo (escribir documentos), dividió la clase en diferentes zonas, cambiando el formato de notas por puntos, lo cual permite cambiar la percepción de los estudiantes con respecto a las notas, dicho cambio premia a los estudiantes al realizar las cosas bien en lugar de solamente castigarlo en caso de hacer las cosas de manera incorrecta.

Otra investigación es Re-Mission, la cual tiene como objetivo controlar un nano robot, el cual se mueve en el cuerpo de un paciente destruyendo las células cancerígenas mediante quimioterapia y radiografía, con el fin de mostrar el beneficio de dichas

técnicas, los resultados obtenidos permitieron generar un mayor conocimiento y entendimiento del tratamiento del cáncer en los pacientes en comparación de los que no habían jugado (Orji, Mandryk, Vassileva, & Gerling, 2013).

Otro proyecto es Help que es una plataforma para diferentes estudiantes, donde los alumnos pueden buscar respuestas a preguntas, mediante la ayuda de otros estudiantes, donde si un alumno conoce la respuesta gana una recompensa (Díaz Cruzado & Troyano Rodríguez, 2013).

Los jugadores van comenzando con un cierto puntaje el cual se va incrementando o reduciendo dependiendo de su colaboración en el proyecto, ese puntaje puede ser usado para canjear objetos.

Otras empresas como Google y Yahoo crearon aplicaciones similares: Google Answer y Yahoo Answers (Orji et al., 2013).

Otro programa usado para la enseñanza de programación es CodeAcademy (Sims & Bubinski, 2011), donde se enseñan cursos de programación, los cuales son organizados e interconectados, consisten de una secuencia de ejercicios con un mensaje que explica el tema, instrucciones indicando la tarea a realizar y un ejercicio interactivo real que se debe completar, finalmente cada estudiante gana puntos por completar cada ejercicio y se guarda cada lección aprendida como logros.

Mientras que la mayoría de plataformas educativas enfocadas en la gamification usan logros, puntos o medallas, Codingames usa juegos para ayudar el proceso educativo, tema incluye más de 20 diferentes tipos de lenguaje de programación y cada alumno puede ver lo que está codificando en la pantalla (Butt, 2016).

1.3.2. Bases Teóricas del proyecto

Software Educativo

El software educativo es la herramienta tecnológica para la nueva sociedad del aprendizaje (Fredes et al., 2012; Minnaard & Minnaard, 2014) por lo que se han desarrollado estrategias en diferentes niveles, desde el nivel institucional hasta el salón de clases, con el objetivo de buscar el uso apropiado del software educativo.

La educación especialmente ha recibido mucha atención por parte de investigadores y compañías (Buisman & van Eekelen, 2014).

En la educación se ha utilizado la gamification para motivar a los estudiantes a aprender mediante el uso de elementos del diseño de juegos, donde se busca aumentar la diversión, participación y motivación del estudiante logrando que continúe aprendiendo (Azmi, Iahad, & Ahmad, 2015).

Gamification

El término gamification es la aplicación de elementos de juego y principios de diseños de juegos en otros contextos no relacionados con juegos (Deterding et al., 2011).

En resumen, la gamification utiliza elementos de diseño de juegos para mejorar la motivación, la participación y la productividad del usuario (Deterding, Dixon, Khaled, & Nacke, 2011).

La gamification se relaciona con los juegos y no con jugar, donde el jugar es una forma más libre y expresiva y el juego es un conjunto de normas para conseguir una meta (Caillois, 2001).

Los principales elementos de diseño de juegos usados en la gamification son (Akilli, 2011):

- Mecánicas de Progresión.
- Narrativa.
- Control del Jugador.
- Retroalimentación.
- Resolución de Problemas en Forma Colaborativa.
- Oportunidades de Dominio.
- Progresión por Niveles.
- Relaciones Sociales.
- Retos.
- Música.

Gamification en la Educación

En la educación se ha utilizado la gamification para motivar a los estudiantes a aprender mediante el uso de elementos del diseño de juegos, donde se busca aumentar la diversión, participación y motivación del estudiante logrando que continúe aprendiendo (Azmi et al., 2015).

Una diferencia en el uso de gamification con el aprendizaje basado en juegos, es que no involucra al estudiante en el diseño y creación del juego, ya que en la gamification se busca que el aprendizaje se consiga en un contexto no de juegos, sino en el aula por tal razón se busca realizar la actividad de aprendizaje en un salón de clases (Akilli, 2011).

Las ventajas del uso de gamification en la educación son (Akilli, 2011):

- Libertad para prueba y error sin repercusiones negativas.
- Posibilidad de aumentar la diversión en el salón.
- Educación diferenciada según las habilidades del estudiante.
- Visualización del aprendizaje.
- Brindar un conjunto de tareas y subtareas.
- Motivar a los estudiantes a aprender.
- Libertad al estudiante para realizar su propio aprendizaje.

De las cuales no todas son necesarias usarlas, por tal razón se deben escoger basándonos en la actividad que se desea realizar.

1.4 Técnicas y Herramientas

La técnica escogida para realizar la programación de la aplicación es la metodología ágil programación Extreme Programming (XP), las técnicas de Gamification escogida son mecánicas de progresión, retroalimentación, oportunidades de dominio, progresión por niveles, logros y la herramienta escogida para realizar la aplicación es el entorno de programación Unity

1.4.1. Técnicas

Se ha determinado el uso de una metodología ágil para el desarrollo del software educativo, ya que permite acelerar el proceso de desarrollo del aplicativo mediante iteraciones que se desarrollan a lo largo todo el ciclo de vida del software, existen distintas metodologías ágiles, para el desarrollo del software educativo se ha optado utilizar la metodología XP.

Una de las principales razones que se optó por el uso de una metodología ágil es la reducción de riesgos en el desarrollo de software debido a las iteraciones que se realizan en cortos lapsos de tiempo, una iteración puede ser comprendida como una nueva versión del programa hechas en un determinado lapso de tiempo, donde cada iteración dura entre una a cuatro semanas.

La metodología propuesta en XP está diseñada para entregar el software que los usuarios necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo (J. Joskowicz, 2008).

Se determinó que la metodología XP resultaría ser la metodología ideal, debido a que como el proyecto se va a desarrollar en un grupo pequeño, el cual necesita diferentes iteraciones para su desarrollo, debido a que se realizarán diferentes reuniones con profesores para determinar las tareas a realizar y poder hacer una comprobación del nivel educativo y finalmente en la última fase se hará una revisión final del software y la evaluación que se realizará a los alumnos en el ámbito de software.

Las técnicas de gamification escogidas para el desarrollo de la aplicación son las siguientes (Akilli, 2011):

- **Mecánicas de Progresión:** Mediante un sistema de niveles donde el alumno ganará experiencia y va subiendo de nivel.
- **Retroalimentación:** Se mostrará mensajes en la pantalla cuando el alumno se equivoque, se mostrará mensajes cuando seleccione una respuesta

correcta, dichos mensajes serán complementados con sonidos los cuales permitirán identificar si se ha equivocado o ha acertado.

- Oportunidades de Dominio: El alumno tendrá la oportunidad de volver a realizar las actividades hasta completarlas en su totalidad.
- Progresión por Niveles: Se mostrará una serie de niveles los cuales irán desde básico, intermedio y avanzado; se mostrará el avance de la progresión de cada nivel para saber cuál es el porcentaje faltante para completarlo.
- Logros: Se diseñará una serie de logros los cuales se irán desbloqueando cuando se completen, luego se mostrará los diferentes logros y sus avances.

1.4.2. Herramientas

Unity es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux, Unity tiene soporte para las siguientes plataformas: WebGL, Windows, Windows Store Apps, SteamOS, OS X, Linux, iOS, Android, Windows Phone, Tizen, tvOS, Samsung Smart TV, Android TV, PlayStation Vita, PlayStation 4, Xbox 360, Xbox One, Wii U, Nintendo 3DS, Nintendo Switch, Oculus Rift, Google Cardboard, HTC Vive, PlayStation VR, Samsung Gear VR y Microsoft Hololens (Unity Technologies, n.d.).

La versión actual de Unity es 2018.2.9 la cual presenta funciones nuevas que ayudan a equipos de artistas y desarrolladores a crear experiencias en conjunto.

Potentes herramientas nuevas, como Timeline y Cinemachine, permite a los artistas a crear contenido cinemático y secuencias de desarrollo de juegos sin necesidad de un ingeniero (Unity Technologies, n.d.).

El editor de Unity permite el desarrollo de proyectos 2D y 3D, cuenta con herramientas como pathfinding la cual permite crear un sistema de navegación que permite crear rutas predefinidas, donde se crea mallas de navegación basándose en la escena que determinan los obstáculos de manera dinámica y cambia el recorrido en tiempo de ejecución, adicionalmente existe un motor de física que usa de Box2D y NVIDIA PhysX que permiten crear simulaciones realista y de alto rendimiento, finalmente Unity permite personalizar su editor donde se puede agregar nuevas extensiones e identificadores (Unity Technologies, n.d.).

Unity cuenta con una tienda de contenido la cual permite comprar contenido personalizado optimizado para los proyectos de Unity, cuenta con un gran catálogo de recursos gratuito y de pago, existe una gran variedad de contenido como: arte, modelos, código, herramientas de productividad, y más (Unity Technologies, n.d.).

1.5 Aspectos Relevantes del Desarrollo

En una primera etapa se realizó un análisis de las actividades que se podrían incluir en el software educativo, luego de haber consultado con especialistas educativos se escogieron las siguientes actividades: verdadero o falso, preguntas y conectar conceptos.

Luego de haber establecido estas actividades, se comenzó definir el diseño de la interfaz donde se buscó un diseño minimalista en el cual se use poco texto y se base en

iconos, por lo cual se comenzó a diseñar una lista de iconos para representar las diferentes actividades como se observa en la Figura 1.



Figura 2. Iconos de las actividades (Fuente: Elaboración Propia)

Se definió como ícono del nivel del jugador mostrar un número que represente el nivel actual y que se muestre un círculo que se va llenando en base al porcentaje faltante para llegar al siguiente nivel como se muestra en la Figura 2



Figura 2. Icono del nivel de jugador (Fuente: Elaboración Propia)

La fuente que se estableció en la aplicación es droid serif la cual es la fuente por defecto de Google Android, se escogió esta letra debido a que es gratuita y la mayoría de

aplicaciones en Android usan este tipo de letra porque está diseñada para ser vista con facilidad en equipos móviles.

Se definieron las siguientes escenas para la aplicación:

- Una escena de comienzo la cual contendrá el nombre del programa educativo, un botón para comenzar la aplicación, un conjunto de opciones que permita cambiar las configuraciones de la aplicación.
- Una escena que sirva de menú principal, la cual muestre el progreso del jugador, las actividades a realizar, el perfil del jugador, y los logros.
- Crear una escena para la actividad de pregunta, la cual según una serie de opciones se debe escoger la respuesta y debe mostrar mensajes en caso que se equivoque para conocer su error.
- Crear una escena de verdadero o falso en la cual se muestre una pregunta y se presentan dos posibles opciones.
- Crear una escena que permita conectar conceptos, donde el estudiante pueda arrastrar los objetos en la pantalla, en caso de ser correcto se debe bloquear el objeto para que no se pueda mover.

Luego de definir el diseño se determina los componentes de la aplicación, los componentes que se definieron son:

- Primero se definió un sistema de control de sonido, en el cual se pondrá los diferentes efectos de sonidos y sonidos de ambiente, este componente será accedido por todos los demás componentes, se usará un patrón

Singleton en la codificación de este sistema, ya que permite ser accedido de una manera rápida.

- Segundo se creará un sistema de control del aplicativo, en el cual se podrá controlar el cambio entre escenas, la información del perfil y los logros desbloqueados por el jugador, este sistema estará a cargo de determinar la experiencia ganada por el jugador según la actividad, para desbloquear los logros será necesario el uso del patrón *Singleton*, el cual tendrá que ser llamado cada vez que un jugador realice una acción que le permita completar un logro ya que los logros son acumulativos y por esta razón se tiene que ir registrando cuántos puntos ha ganado, también es necesario saber cuándo se desbloquea el logro para mostrar una imagen que indique que logro ha sido obtenido.
- Tercero se debe crear un sistema de guardado de información, este sistema necesita ser llamado al iniciar la aplicación para permitir cargar la información, Unity permite el uso de serialización para el grabado de información pero cómo se va a trabajar con Android es necesario hacer modificaciones en el Manifest para lograr que se grabe dicha información dándole permiso para el almacenamiento externo, el sistema de guardado trabajara dentro de la aplicación como un objeto que se va a mover entre escenas el cual va a poder ser accedido en cualquier momento por cualquier otro componente.

- Finalmente, se debe realizar un sistema para todas las actividades, dicho sistema debe tener diferentes opciones según la actividad a realizar, en caso de ser una pregunta debe tener una lista de opciones, título de la pregunta, lista de mensajes de respuestas; para la actividad de verdadero o falso es necesario guardar el título de lo que se pregunta, un valor booleano que determine si es verdadero o falso y finalmente para la actividad de conectar conceptos es necesario crear un sistema de drag and drop que permitirá poder arrastrar objetos de la pantalla y ponerlos en un lugar predeterminado, finalmente se debe volver a colocar el objeto a su lugar en caso que se arrastre a una posición errónea.

Para el diseño de la interfaz se ha optado por basarse en la interfaz de Duolingo, la cual es una aplicación para la enseñanza de otro idioma mediante la gamification como se muestra en la Figura 3.

Para la programación en Unity primero fue necesario comenzar con bajar el Kit de desarrollo de software de Android (SDK) y el Java Development Kit (JDK) los cuales son necesarios para el desarrollo de aplicaciones en Android, los cambios que se deben hacer en la configuración de construcción es cambiar la plataforma a Android y cambiar el nombre del paquete.

Se escogió por usar el lenguaje de programación C# porque es el lenguaje que usa por defecto Unity.

Finalmente, para hacer las pruebas primero es necesario hacer la compilación desde Unity y luego realizar la compilación al dispositivo móvil, en las pruebas se revisa

si el sistema guarda la información, realiza de manera correcta las actividades, reconoce los logros y los botones se adaptan según la pantalla.

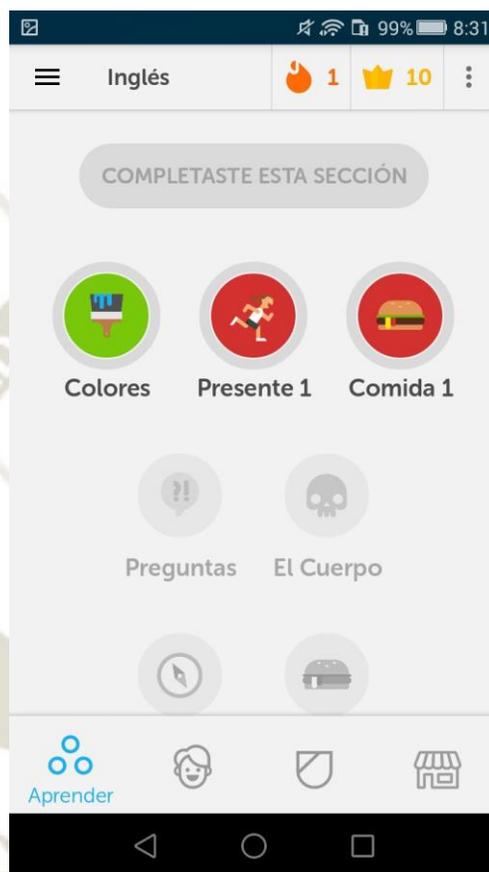


Figura 3. Interfaz de Duolingo (Fuente: Duolingo)

2. DOCUMENTACIÓN TÉCNICA

2.1. Plan del Proyecto Informático

2.1.1. Planificación temporal del proyecto

Se desarrolló un plan de trabajo del software educativo como se mira en la Tabla 1 en la cual se creó un diagrama de Gantt especificando las tareas a realizar, teniendo una totalidad de diez meses, el cual comenzó en el mes de marzo el primer mes se realizó una investigación de las diferentes plataformas educativas que usan gamification, luego de haber realizado esta investigación se hizo una revisión de las diferentes técnicas de gamification, seguidamente se comenzó con el desarrollo de los análisis de requisitos lo cual se realizó en conjunto con un profesor con el cual se determinó las actividades a realizar además gracias a la investigación de las plataformas se usaron su interfaz como referencia para el diseño de la aplicación, luego se preparó el diseño del programa donde se realizó un mockup del aplicativo en el cual se determinó el diseño de la interfaz y de las actividades, se usó como referencia para el diseño del menú la aplicación de Duolingo como se observa en la Figura 3, después se buscaron los recursos necesarios, como música de fondo se usó sonidos instrumentales para evitar distraer al alumno en las actividades, se buscaron efectos de sonidos como: sonido de clic, sonido de felicitación, sonido de fallo, entre otros, finalmente se buscaron imágenes para el fondo de la aplicación, para los botones, mensajes e iconos, luego se realizó la codificación de los cuatro módulos de la aplicación, se realizó la documentación del software, después se comenzaron a realizar las pruebas en busca de errores para corregirlos.

Finalmente se hizo una estimación del costo y el tiempo del desarrollo del proyecto, la cual fue realizada mediante el juicio de expertos donde se determinó en un inicio cuál sería el costo por hora según la actividad y se dio un costo de 4 soles a la hora de trabajo de campo, 5 soles a la hora de diseño e investigación y 8 soles a la hora de programación, según se observa en la Tabla 1 la duración del proyecto es de 925 horas y el costo del proyecto es 5,945.00 soles.

2.1.2. Estudio de viabilidad del proyecto

La presente tesis se encuentra orientada a profesores que deseen innovar el proceso educativo de temas referentes a la programación mediante el uso de una aplicación educativa que usa la gamification para motivar a los estudiantes.

La aplicación está orientada a profesores del área de programación, existe una serie de herramientas planteadas para la enseñanza de programación donde se da una mayor importancia al desarrollo de las habilidades de resolver problemas (Al-imamy, Alizadeh, & Nour, 2006; Robins, Rountree, & Rountree, 2003), también se han creado un conjunto de lecciones en las cuales se crea un ambiente de programación para la enseñanza de programación orientada a objetos (Satratzemi, Xinogalos, & Dagdilelis, 2003).

Otras aplicaciones han buscado crear una propuesta para la enseñanza del curso de programación usando gamification (Gallego-Durán, Villagrà-Arnedo, Llorens-Largo, & Molina-Carmona, 2017), donde se observó que el dar una retroalimentación y un sentido de progresión al curso de programación ayuda a motivar a los estudiantes, pero un problema que surge es el convertir la enseñanza de estos conceptos en juegos y actividades educativas.

La presente tesis demuestra que es posible realizar un software educativo para la enseñanza de programación usando técnicas de gamification, pero requiere un trabajo en conjunto tanto del desarrollador como del profesor para determinar las actividades a realizar y el diseño de la aplicación.

La solución se encuentra enfocada en el uso de dispositivos móviles específicamente celulares Android para su utilización, pero puede ser adaptado a otros entornos como celulares Iphone o computadoras, lo cual es una ventaja del entorno de programación escogido.

Pero se encontraron limitaciones ya que al querer crear una aplicación de mayor accesibilidad aparece una limitación en los dispositivos móviles para las pruebas, por lo cual se ha optado por el uso exclusivo de dispositivos Android de gama media.

Por otro lado el usar un motor de videojuegos como Unity facilita el desarrollo de una aplicación, ya que al existir una gran cantidad de tutoriales y guías se facilita el aprendizaje de dicho motor de videojuegos.

Otro factor que influyó el escoger Unity fue la tienda de recursos, la cual brinda una gran cantidad de recursos libres que facilitan la creación de un proyecto.

Problemas referentes a la complejidad técnica como: crear un renderizado 2D, motor de física, motor de audio, motor de colisiones, animaciones, etc., fueron solucionados con el uso de Unity ya que permite abstraer las capas más bajas de la arquitectura de una aplicación educativa.

Los componentes del aplicativo son creados mediante scripts que es el término que se le da en Unity a los archivos de código.

Se ha estimado el desarrollo de la aplicación en un periodo de 10 meses, los cuales incluyen un tiempo inicial para la investigación de plataformas similares, seguido del análisis de requerimientos, el diseño del aplicativo, realizar la codificación, unir los componentes, realizar la documentación y pruebas.

Las herramientas escogidas son libres por tal motivo se ha reducido los gastos económicos de licencias, ya que el motor de juegos Unity permite el desarrollo de aplicaciones en un plan personal en el cual no es necesario realizar ningún pago.

2.1.3. Historias de usuario y criterios de aceptación

Se hicieron entrevistas a diferentes profesores donde se recopilaban las historias de usuarios y los criterios de aceptación, con lo cual se pudo determinar los requerimientos funcionales y no funcionales como se observa en la Tabla 2.

Tabla 2. Tabla de las historias de usuarios (Fuente: Elaboración Propia)

Historias de Usuarios	
Número: 1	Nombre: Ingresar a la aplicación
Usuario: Estudiante	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Descripción: Como usuario de la aplicación, deseo ingresar a la aplicación, para poder hacer uso de las funcionalidades de la aplicación.	
Validación: Debe permitir el ingreso de información personal del usuario. Cuando pulse el botón de “Ok” debe de registrarse la solicitud en el sistema y guardar la información. No debo poder ingresar caracteres en el campo edad.	

Número: 2	Nombre: Ver logros
Usuario: Estudiante	
Prioridad en Negocio: Bajo	Riesgo en Desarrollo: Media
Descripción: Como usuario de la aplicación, quiero poder consultar los logros desbloqueados, para saber el estado de mis logros.	
Validación: Quiero poder ver el avance de mis logros. Quiero poder ver información sobre cómo se bloquea un logro. No debo poder desbloquear un mismo logro varias veces. Debo poder observar cuando un logro se desbloquea.	
Número: 3	Nombre: Ingresar a las actividades
Usuario: Estudiante	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Media
Descripción: Como usuario de la aplicación, quiero ingresar a las actividades, para poder cambiar a la escena de la actividad seleccionada.	
Validación: Se debe ver el avance de cada actividad. Debe mostrar una descripción de cada actividad. Cuando pulse el botón de “Ok” debo poder ingresar a la escena de cada actividad. Debe mostrar cuando una actividad ha sido completada. Se deben crear tres tipos de actividades: Preguntas, Verdadero o Falso y Conectar Conceptos.	
Número: 4	Nombre: Responder Actividad Pregunta
Usuario: Estudiante	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Descripción: Como usuario de la aplicación, quiero poder señalar una respuesta en la actividad	

pregunta, para poder completar la actividad pregunta.	
Validación: Se debe mostrar la pregunta y una lista de opciones. Debe permitir regresar al menú principal. Cuando se conteste una actividad de manera incorrecta se debe mostrar un mensaje que explique el error. Cuando se conteste una actividad de manera correcta se debe mostrar un mensaje que explique porqué es correcto.	
Número: 5	Nombre: Responder Actividad Verdadero o Falso
Usuario: Estudiante	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Descripción: Como usuario de la aplicación, quiero poder señalar una respuesta en la actividad verdadero o falso, para poder completar la actividad verdadero o falso.	
Validación: Se debe mostrar la pregunta y dos opciones verdadero o falso. Debe permitir regresar al menú principal. Cuando se conteste una actividad de manera incorrecta se debe mostrar un mensaje que explique el error. Cuando se conteste una actividad de manera correcta se debe mostrar un mensaje que explique porqué es correcto.	
Número: 6	Nombre: Arrastrar Alternativa
Usuario: Estudiante	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Media
Descripción: Como usuario de la aplicación, quiero poder arrastrar una alternativa, para poder completar la actividad conectar conceptos.	
Validación: Se debe mostrar una lista de conceptos y una lista de alternativas.	

<p>Debe permitir regresar al menú principal.</p> <p>Cuando se arrastre una alternativa a un concepto incorrecto se debe mostrar un mensaje que explique el error.</p> <p>Cuando se arrastre una alternativa a un concepto correcto se debe mostrar un mensaje que explique porqué es correcto.</p>	
Número: 7	Nombre: Desbloquear Logro
Usuario: Estudiante	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Media
Descripción: Como usuario de la aplicación, quiero poder desbloquear logros, para poder completar todos los logros.	
Validación: Se debe mostrar un mensaje cuando desbloqueo un logro. Quiero ver el nombre del logro y la descripción del logro. Cuando entre en la actividad ver logros quiero ver mi logro desbloqueado.	
Número: 8	Nombre: Guardar Información
Usuario: Estudiante	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Alto
Descripción: Como usuario de la aplicación, quiero que la aplicación grabe mi avance, para poder ver mi porcentaje de avance en cada actividad.	
Validación: Se debe guardar la información de cada actividad. Quiero ver el porcentaje de avance de cada actividad. Quiero ver el porcentaje de avance del curso.	
Número: 9	Nombre: Ir a la siguiente actividad
Usuario: Estudiante	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Media
Descripción:	

Como usuario de la aplicación, quiero al completar una actividad pueda ir a la siguiente actividad, para no tener que regresar al menú principal.

Validación:

Se debe mostrar una lista de botones al finalizar una actividad.

Quiero poder apretar un botón y que me llegue a la siguiente actividad que debo realizar.

Quiero que exista un botón que me permita regresar al menú principal.

2.2 Especificación de requisitos del software

2.2.1 Propósito:

El propósito de esta sección es presentar una descripción detallada del software educativo, explicando el propósito, características del sistema, restricciones bajo las cuales deben operar y como el sistema reaccionara ante los estímulos externos.

Esta sección tiene como objetivo tanto a los interesados como a los desarrolladores y se ha hecho basándonos en los estándares de IEEE (Standard, 1993).

2.2.2 Alcance del proyecto

Este programa está diseñado para maximizar la motivación de los estudiantes con respecto al curso de programación para ayudar a la enseñanza de conceptos básicos de programación, sirviendo como una herramienta complementaria al proceso educativo que de otra manera solamente se realizaría dentro del salón de clases y no se usaría dispositivos móviles para su enseñanza.

Más específicamente, este software educativo está diseñado para que un profesor del curso de programación pueda compartir con sus alumnos un programa para medir su avance sobre el curso de programación.

El software facilitará la enseñanza de conceptos básicos de programación, el uso de gamification permite mejorar la motivación del alumno con respecto al curso, esto es debido a que existirá una retroalimentación y permite a los alumnos percibir sus avances con respecto al curso, el sistema adicionalmente permite ver los avances de los alumnos.

2.2.3 Glosario

Tabla 3. Tabla del glosario del requisito de software (Fuente: Elaboración Propia)

Término	Definición
Base de Datos	Recolección de toda la información creada por este sistema.
Especificación de Requerimientos de Software	Un documento que describe completamente todas las funciones de un sistema propuesto y las restricciones bajo las cuales debe operar.
Usuario	Estudiante del curso de programación

2.2.4 Entorno del sistema

El software educativo para la enseñanza de programación cuenta con un usuario el estudiante, quien accede al sistema de la aplicación y desde ahí puede acceder a las actividades.

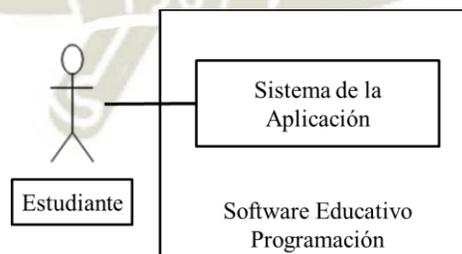


Figura 4. Entorno del sistema (Fuente: Elaboración Propia)

2.2.5 Especificación de requisitos funcionales

En esta sesión se describe los diferentes casos de estudios para el estudiante.

2.2.5.1 Estudiante

El estudiante tiene los siguientes casos de uso

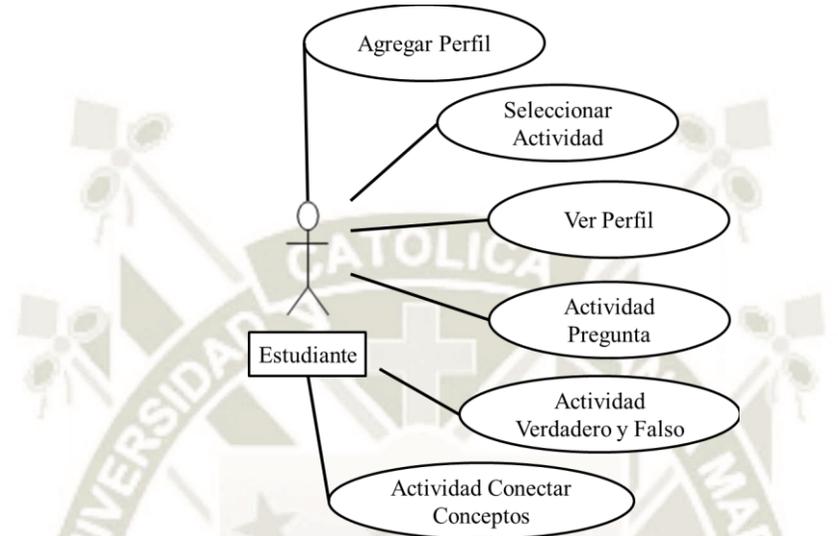


Figura 5. Casos de uso del estudiante (Fuente: Elaboración Propia)

Información de los casos de uso

2.2.5.1.1 Caso de uso: Agregar Perfil

Diagrama:

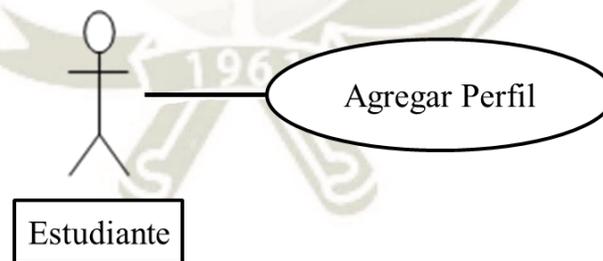


Figura 6. Casos de uso de agregar perfil (Fuente: Elaboración Propia)

Breve descripción:

El estudiante ingresa sus datos personales a la aplicación.

Descripción paso a paso:

Primero es necesaria la instalación de la aplicación.

1. El estudiante abre la aplicación.
2. El sistema brinda una caja de texto para agregar la información.
3. El estudiante agrega su información personal.
4. El sistema guarda la información y carga la escena del menú principal.

Requisito funcional:

Sección 2.2.9.1. Agregar Perfil

2.2.5.1.2 Caso de uso: Seleccionar Actividad

Diagrama:



Figura 7. Casos de uso de seleccionar actividad (Fuente: Elaboración Propia)

Breve descripción:

El estudiante ingresa al menú y selecciona qué actividad realizar.

Descripción pasó a paso:

Antes de que este caso de uso pueda ser inicializado es necesario haber ingresado los datos personales del usuario.

1. El estudiante entra al menú principal.
2. El sistema carga la información guardada del sistema.
3. El sistema muestra las diferentes actividades y reproduce el sonido de fondo correspondiente.

4. El estudiante escoge la actividad a realizar.
5. El sistema le muestra una descripción de la actividad a realizar.
6. El estudiante acepta ingresar a la actividad.
7. El sistema cambia la escena y carga la información de la actividad.

Requisito funcional:

Sección 2.2.9.2. Seleccionar Actividad

2.2.5.1.3 Caso de uso: Ver Perfil

Diagrama:

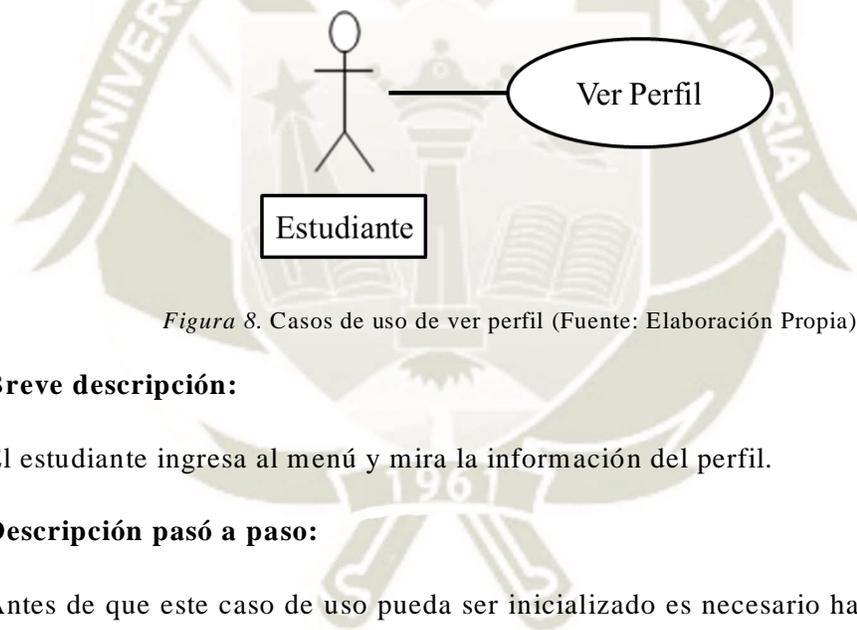


Figura 8. Casos de uso de ver perfil (Fuente: Elaboración Propia)

Breve descripción:

El estudiante ingresa al menú y mira la información del perfil.

Descripción pasó a paso:

Antes de que este caso de uso pueda ser inicializado es necesario haber ingresado los datos personales del usuario.

1. El estudiante entra al menú principal.
2. El sistema carga la información guardada del sistema.
3. El estudiante selecciona la opción de perfil
4. El sistema muestra la información del perfil
5. El estudiante navega y observa la información

Requisito funcional:

Sección 2.2.9.3. Ver Perfil

2.2.5.1.4 Caso de uso: Actividad Pregunta

Diagrama:

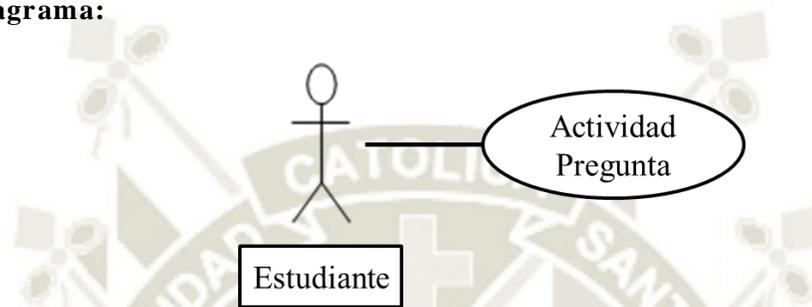


Figura 9. Casos de uso de la actividad pregunta (Fuente: Elaboración Propia)

Breve descripción:

El estudiante ingresa a la actividad de pregunta y resuelve la actividad.

Descripción pasó a pasó:

Antes de que este caso de uso pueda ser inicializado es necesario que el alumno haya seleccionado la actividad pregunta.

1. El estudiante entra a la escena actividad pregunta.
2. El sistema carga la información guardada del sistema.
3. El sistema selecciona una pregunta de manera aleatoria de una lista, carga el título, una lista de opciones que se carga como botones, y una lista de respuesta.
4. El estudiante selecciona una respuesta.
5. El sistema revisa la respuesta, en caso haya sido la correcta se muestra un mensaje de felicitación, en caso de ser incorrecto se muestra un mensaje de error y se guarda la información del resultado, finalmente si se ha cumplido todas las preguntas, se va al siguiente paso, caso contrario vuelve al paso 3.

6. El sistema carga los resultados obtenidos de la actividad y el estudiante luego puede ir a la siguiente actividad o regresar al menú principal.

Requisito funcional:

Sección 2.2.9.4. Actividad Pregunta

2.2.5.1.5 Caso de uso: Actividad Verdadero o Falso

Diagrama:

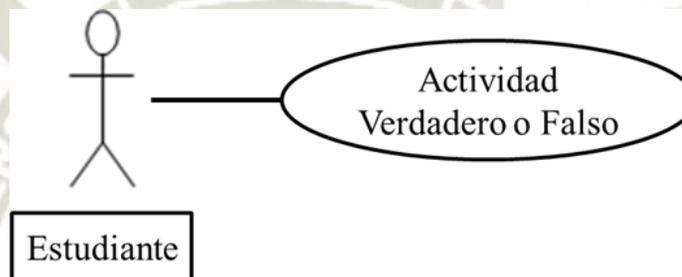


Figura 10. Casos de uso de la actividad verdadero o falso (Fuente: Elaboración Propia)

Breve descripción:

El estudiante ingresa a la actividad de verdadero o falso, luego resuelve la actividad.

Descripción pasó a paso:

Antes de que este caso de uso pueda ser inicializado es necesario que el alumno haya seleccionado la actividad pregunta.

1. El estudiante entra a la escena actividad pregunta.
2. El sistema carga la información guardada del sistema.
3. El sistema selecciona una pregunta de manera aleatoria de una lista, carga el título, la respuesta correcta y una lista de respuesta.
4. El estudiante selecciona una respuesta.

5. El sistema revisa la respuesta, en caso haya sido la correcta se muestra un mensaje de felicitación, en caso de ser incorrecto se muestra un mensaje de error, se guarda la información del resultado y finalmente si se ha cumplido todas las preguntas, se va al siguiente paso caso contrario vuelve al paso 3.

6. El sistema carga los resultados obtenidos de la actividad y el estudiante puede ir a la siguiente actividad o regresar al menú principal.

Requisito funcional:

Sección 2.2.9.5. Actividad Verdadero o Falso

2.2.5.1.6 Caso de uso: Actividad Conectar Conceptos

Diagrama:

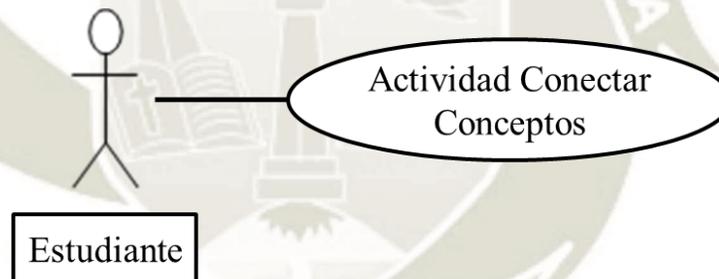


Figura 11. Casos de uso de la actividad de conectar conceptos (Fuente: Elaboración Propia)

Breve descripción:

El estudiante ingresa a la actividad de conectar conceptos, luego resuelve la actividad.

Descripción pasó a pasó:

Antes de que este caso de uso pueda ser inicializado es necesario que el alumno haya seleccionado la actividad pregunta.

1. El estudiante entra a la escena actividad pregunta.

2. El sistema carga la información guardada del sistema.
3. El sistema selecciona una pregunta de manera aleatoria de una lista, carga una lista de opciones, una lista de mensajes y un título.
4. El estudiante arrastra una opción y la coloca en un espacio predefinido.
5. El sistema revisa la respuesta, en caso haya sido la correcta la opción se queda en la nueva posición, en caso de ser incorrecto la opción vuelve a su posición original, se guarda la información del resultado y finalmente si se ha cumplido todas las preguntas, se va al siguiente paso caso contrario vuelve al paso 4.
6. El sistema carga los resultados obtenidos de la actividad y le da opción al estudiante para ir a la siguiente actividad o regresar al menú principal.

Requisito funcional:

Sección 2.2.9.5. Actividad Conectar Conceptos

2.2.6 Características del usuario

El estudiante debe tener conocimiento de uso de dispositivos móviles, debe conocer cómo usar aplicaciones de funcionamiento táctil.

2.2.7 Requisitos de interfaz externa

El vínculo externo que contiene esta aplicación es con un archivo serializado con lo cual se puede guardar la información del jugador, este archivo se encontrará en el celular y podrá ser accedido con un explorador de archivos.

2.2.8 Requerimientos funcionales

2.2.8.1 Agregar Perfil

Tabla 4. Tabla del requisito funcional agregar perfil (Fuente: Elaboración Propia)

Nombre	Agregar Perfil.
Especificación	Sección 2.2.5.1.1, Ver Perfil.
Trigger	El usuario inicial la aplicación por primera vez.
Condición previa	El usuario ha instalado la aplicación.
Ruta Básica	<ol style="list-style-type: none"> 1. El estudiante ingresa su nombre. 2. El estudiante ingresa su edad. 3. El estudiante ingresa su orientación sexual. 4. El sistema guarda la información ingresada.
Ruta Alternativa	Si en el paso 1 no se ha agregado información el sistema lo reconoce como un error.
Poscondición	La información es guardada en el sistema.
Rutas de excepción	El usuario cierra la aplicación.
Otros	Ninguno.

2.2.8.2 Seleccionar Actividad

Tabla 5. Tabla del requisito funcional seleccionar actividad (Fuente: Elaboración Propia)

Nombre	Seleccionar Actividad.
Especificación	Sección 2.2.5.1.2, Seleccionar Actividad.
Trigger	El usuario abre la aplicación y presiona el botón de comenzar.
Condición previa	El usuario ha agregado sus datos.
Ruta Básica	<ol style="list-style-type: none"> 1. El estudiante abre la aplicación. 2. El sistema carga la información referente al usuario como: su experiencia, nivel y porcentaje de avance. 3. El sistema carga y reproduce la música del fondo. 4. El sistema carga los nombres y datos de las actividades. 5. El estudiante navega en la aplicación y selecciona la actividad a realizar. 6. El sistema le muestra una descripción de cada actividad. 7. El estudiante acepta ingresar a la actividad. 8. El sistema realiza el cambio de la escena y carga información sobre la nueva actividad.
Ruta Alternativa	Si en el paso 8 el alumno opta por no aceptar la aplicación el sistema cierra la ventana y vuelve al menú principal.
Poscondición	Se realiza un cambio de escena según la actividad seleccionada
Rutas de excepción	El usuario cierra la aplicación
Otros	Ninguno

2.2.8.3 Ver Perfil

Tabla 6. Tabla del requisito funcional ver perfil (Fuente: Elaboración Propia)

Nombre	Ver Perfil.
Especificación	Sección 2.2.5.1.3, Ver Perfil.
Trigger	El usuario abre la aplicación y presiona el botón de comenzar.
Condición previa	El usuario ha agregado sus datos.
Ruta Básica	<ol style="list-style-type: none"> 1. El estudiante abre la aplicación. 2. El sistema carga la información referente al usuario. 3. El sistema carga la música del fondo de la aplicación y la reproduce. 4. El sistema carga los botones como los botones de perfil y logros. 5. El estudiante navega en la aplicación y presiona el botón de perfil. 6. El sistema carga la escena de perfil. 7. Se carga la información del estudiante como su nombre, edad y orientación sexual.
Ruta Alternativa	Si en el paso 5 el alumno opta por no presionar el botón el alumno se mantiene en el menú principal.
Poscondición	Se realiza un cambio de escena a la escena de perfil
Rutas de excepción	El usuario cierra la aplicación
Otros	Ninguno

2.2.8.4 Actividad Pregunta

Tabla 7. Tabla del requisito funcional actividad pregunta (Fuente: Elaboración Propia)

Nombre	Actividad Pregunta.
Especificación	Sección 2.2.5.1.4, Actividad Pregunta.
Trigger	El usuario ha hecho clic en la actividad pregunta.
Condición previa	El usuario ha seleccionado una actividad del tipo pregunta.
Ruta Básica	<ol style="list-style-type: none"> 1. El estudiante ha entrado a la escena actividad pregunta. 2. El sistema carga la información referente a la actividad pregunta. 3. El sistema carga la música del fondo de la aplicación y la reproduce. 4. El sistema carga una lista de preguntas referente a la actividad y se selecciona una de manera aleatoria. 5. El sistema carga el título de la pregunta, una lista de

	<p>opciones para responder, una lista de mensajes acorde a la respuesta.</p> <ol style="list-style-type: none"> 6. El estudiante navega en la aplicación y presiona el botón de respuesta correcto. 7. El sistema revisa la respuesta, en caso haya sido correcta se muestra un mensaje de felicitación. 8. El sistema guarda la información referente a la actividad. 9. El sistema revisa si se ha cumplido toda la lista de preguntas se pasa al siguiente paso en caso que aún quede preguntas se vuelve a seleccionar una pregunta de manera aleatoria y se regresa al paso 5. 10. El sistema graba los datos de la actividad. 11. El sistema carga un menú para cambiar de escena. 12. El estudiante selecciona ir a la siguiente actividad y se carga la siguiente escena.
Ruta Alternativa	<p>Si en el paso 7 el alumno opta por una respuesta incorrecta se muestra un mensaje de error.</p> <p>Si en el paso 12 se selecciona regresar al menú principal se carga la escena de menú principal.</p>
Poscondición	Se guarda la información de la actividad resuelta
Rutas de excepción	El usuario cierra la aplicación
Otros	Ninguno

2.2.8.5 Actividad Verdadero o Falso

Tabla 8. Tabla del requisito funcional actividad verdadero o falso (Fuente: Elaboración Propia)

Nombre	Actividad Pregunta.
Especificación	Sección 2.2.5.1.5, Actividad Verdadero o Falso.
Trigger	El usuario ha hecho clic en la actividad verdadero o falso.
Condición previa	El usuario ha seleccionado una actividad del tipo verdadero o falso.
Ruta Básica	<ol style="list-style-type: none"> 1. El estudiante ha entrado a la escena de actividad verdadero o falso 2. El sistema carga la información referente a la actividad verdadero o falso. 3. El sistema carga la música del fondo de la aplicación y la reproduce. 4. El sistema carga una lista de verdadero o falso referente a la actividad y se selecciona una de manera aleatoria. 5. El sistema carga el título de la pregunta, la respuesta

	<p>correcta, una lista de mensajes acorde a la respuesta.</p> <ol style="list-style-type: none"> 6. El estudiante navega en la aplicación y presiona el botón de verdadero o falso. 7. El sistema revisa la respuesta, en caso haya sido correcta se muestra un mensaje de felicitación. 8. El sistema guarda la información referente a la actividad. 9. El sistema revisa si se ha cumplido toda la lista de verdadero o falso se pasa al siguiente paso en caso que aún quede preguntas se vuelve a seleccionar un nuevo elemento de la lista de manera aleatoria y se regresa al paso 5. 10. El sistema graba los datos de la actividad. 11. El sistema carga un menú para cambiar de escena. 12. El estudiante selecciona ir a la siguiente actividad y se carga la siguiente escena.
Ruta Alternativa	<p>Si en el paso 7 el alumno opta por una respuesta incorrecta se muestra un mensaje de error.</p> <p>Si en el paso 12 se selecciona regresar al menú principal se carga la escena de menú principal.</p>
Poscondición	Se guarda la información de la actividad resuelta
Rutas de excepción	El usuario cierra la aplicación
Otros	Ninguno

2.2.8.6 Actividad Conectar Conceptos

Tabla 9. Tabla del requisito funcional actividad conectar conceptos (Fuente: Elaboración Propia)

Nombre	Conectar Conceptos.
Especificación	Sección 2.2.5.1.6, Actividad Conectar Conceptos.
Trigger	El usuario ha hecho clic en la actividad conectar conceptos.
Condición previa	El usuario ha seleccionado una actividad del tipo conectar conceptos.
Ruta Básica	<ol style="list-style-type: none"> 1. El estudiante ha entrado a la escena de actividad conectar conceptos 2. El sistema carga la información referente a la actividad conectar conceptos. 3. El sistema carga la música del fondo de la aplicación y la reproduce. 4. El sistema carga el título de la pregunta, una lista botones con las preguntas para seleccionar, se coloca descripciones de los conceptos para ser asociados. 5. El estudiante navega en la aplicación y arrastra un concepto

	<p>a una posición predeterminada que lo conecta con una descripción.</p> <ol style="list-style-type: none"> 6. El sistema revisa la respuesta, en caso haya sido correcta se genera un sonido de respuesta. 7. El sistema guarda la información referente a la actividad. 8. El sistema revisa si se ha cumplido en asociar todos los conceptos en caso que aún falte conceptos para asociar se regresa al paso 5. 9. El sistema graba los datos de la actividad. 10. El sistema carga un menú para cambiar de escena. 11. El estudiante selecciona ir a la siguiente actividad y se carga la siguiente escena.
Ruta Alternativa	<p>Si en el paso 7 el alumno arrastra el concepto a un sitio incorrecto se genera un mensaje de respuesta incorrecta. Si en el paso 11 se selecciona regresar al menú principal se carga la escena de menú principal.</p>
Poscondición	Se guarda la información de la actividad resuelta
Rutas de excepción	El usuario cierra la aplicación
Otros	Ninguno

2.2.9 Requerimientos no funcionales

2.2.9.1 Requerimientos de apariencia o interfaz externa

1. Requerimientos de apariencia o interfaz externa.
2. La aplicación debe tener un diseño minimalista, es importante el uso de iconos.
3. Se debe poder diferenciar los botones de las imágenes.
4. La plataforma debe estar en el idioma español.

2.2.9.2 Requerimientos de Usabilidad

1. El tiempo de aprendizaje de los usuarios con el sistema debería ser menor a 5 horas.
2. Deben existir manuales y guías para los estudiantes correctamente documentados.

3. Los mensajes de error y de éxito deben ser reportados por la aplicación mostrando información referente a la actividad de los usuarios.
4. La aplicación debe tener un diseño intuitivo que permita al estudiante distinguir los objetos para la interacción.
5. La información no debe ser accedida directamente sino mediante una interfaz que permita la interacción de los usuarios.

2.2.9.3 Requerimientos de Rendimientos

1. El sistema debe responder en el mínimo tiempo posible ante las solicitudes por parte del usuario.
2. La aplicación debe acceder a la base de datos de manera rápida cada vez que el usuario genere información.
3. El sistema debe mostrar mensajes de retroalimentación en el menor tiempo posible.

2.2.9.4 Requerimientos de Soporte

1. La aplicación debe contar con manual de ayuda.
2. El manual de ayuda debe contener imágenes que muestran las diferentes escenas.
3. Se debe describir cada escena, describir los diferentes elementos de las escenas y como el usuario se puede desplazar entre escenas.
4. Se debe describir cómo instalar la aplicación.

2.2.9.5 Requerimientos de Seguridad

1. La información debe estar protegida y no deber ser podida accedida por otras personas.
2. Cada usuario solo debe poder ver su información y no debe acceder a la de otro usuario.
3. En caso de contraseñas, la información se debe guardar con algoritmos de encriptación en la base de datos.

2.2.9.6 Requerimientos Políticos

1. La aplicación no debe mostrar lenguaje inapropiado para los estudiantes.
2. La aplicación no debe mostrar imágenes inapropiadas para los estudiantes.

2.2.9.7 Requerimientos Confiabilidad

1. El sistema debe ser tolerante ante fallos.
2. En caso que se produzca errores con el acceso a la información el sistema debe evitar fallos.

2.2.9.8 Ayuda y documentación en línea

1. El sistema debe permitir describir y brindar ayuda los usuarios.
2. Los usuarios deben poder tener acceso a un manual de ayuda.

2.2.9.9 Requerimientos de Software

1. La aplicación debe poder ejecutarse en la plataforma de Android teniendo como API mínima la 19.
2. El sistema debe ser capaz de usar diferentes motores de base de datos para guardar la información.

2.2.9.10 Requerimientos de Hardware

1. La aplicación debe permitir el uso de la base de datos local de los dispositivos móviles.
2. El programa deberá contar con los permisos para el guardado de información.

2.2.10 Estructura Lógica de los Datos

La estructura lógica de los datos que se almacenarán en la aplicación es la siguiente:

Datos del Alumno

Tabla 10. Tabla de la estructura lógica de datos del alumno (Fuente: Elaboración Propia)

Nombre	Tipo	Descripción
Nombre	Texto	Nombre del estudiante
Edad	Numérico	Edad del estudiante
Sexo	Booleano	Orientación sexual
Experiencia	Numérico	Puntos de experiencia
Nivel	Numérico	Nivel actual del estudiante
Preguntas Correctas	Numérico	Número de preguntas respondidas correctamente
Preguntas Incorrectas	Numérico	Número de preguntas respondidas incorrectamente
Porcentaje de Pregunta Verdadero o Falso Correctas	Numérico	Porcentaje para completar la actividad de pregunta
Verdadero o Falso Incorrectas	Numérico	Número de preguntas respondidas correctamente
Porcentaje de Verdadero o Falso	Numérico	Número de preguntas respondidas incorrectamente
Conectar Conceptos Correctas	Numérico	Porcentaje para completar la actividad de pregunta
Conectar Conceptos Incorrectas	Numérico	Número de preguntas respondidas correctamente
Porcentaje de Conectar Conceptos	Numérico	Número de preguntas respondidas incorrectamente
	Numérico	Porcentaje para completar la actividad de pregunta

Datos de la Actividad Pregunta

Tabla 11. Tabla de la estructura lógica de la actividad pregunta (Fuente: Elaboración Propia)

Nombre	Tipo	Descripción	Comentario
Nombre	Texto	Nombre de la pregunta	
Respuesta 1	Texto	Nombre de la primera respuesta a la pregunta	
Respuesta 2	Texto	Nombre de la segunda respuesta a la pregunta	
Respuesta 3	Texto	Nombre de la tercera respuesta a la pregunta	
Respuesta 4	Texto	Nombre de la cuarta respuesta a la pregunta	
Respuesta Correcta	Numérico	Número que indica la respuesta correcta	Es un número entre los valores 1 a 4

Datos de Actividad Verdadero o Falso

Tabla 12. Tabla de la estructura lógica de la actividad verdadero o falso (Fuente: Elaboración Propia)

Nombre	Tipo	Descripción	Comentario
Nombre	Texto	Nombre de la pregunta	
Respuesta Correcta	Booleano	Valor booleano que indica si es verdadero o falso	True para verdadero False para falso

Tabla 13. Tabla de la estructura lógica de la actividad conectar conceptos (Fuente: Elaboración Propia)

Nombre	Tipo	Descripción	Comentario
Nombre	Texto	Nombre de la pregunta	
Concepto 1	Texto	Nombre del primer concepto	
Concepto 2	Texto	Nombre del segundo concepto	
Concepto 3	Texto	Nombre del tercer concepto	
Concepto 4	Texto	Nombre del cuarto concepto	
Descripción 1	Texto	Nombre de la primera descripción	Se conecta con el concepto 1

Descripción 2	Texto	Nombre de la segunda descripción	Se conecta con el concepto 2
Descripción 3	Texto	Nombre de la tercera descripción	Se conecta con el concepto 3
Descripción 4	Texto	Nombre de la cuarta descripción	Se conecta con el concepto 4

2.3 Especificación de diseño

2.3.1 Vista Lógica del Sistema

En la Figura 12 se muestra la vista lógica del sistema, la cual muestra el conjunto de paquetes que contienen la mecánica del software educativo, se muestra el sistema general de la aplicación, el sistema de cada actividad y el sistema de guardado; cada capa contiene los siguientes componentes:

Sistema General: Dispone del sistema de Menú que permite estandarizar las ventanas emergentes y mensajes mostrados en la aplicación, HUD el cual permite adaptar los iconos y botones al tamaño de la pantalla y el sistema de sonido que permite la reproducción de música de fondo y de reproducir efectos de sonido como el hacer clic en los botones y en caso que se resuelva una actividad un sonido de acierto o equivocación.

Sistema de Actividad: Esta capa contiene un control que permite acceder a los diferentes sistemas de cada actividad como: el sistema de preguntas que tiene la función de controlar la actividad pregunta permitiendo mostrar el título de la actividad, una lista de preguntas a ser respondidas, controlar la respuesta del jugador, guardar la información, el siguiente sistema es el de verdadero o falso que dispone de un comportamiento similar al de preguntas con la diferencia que solamente existe dos opciones de respuestas, por último se encuentra el sistema que controla la actividad de conectar conceptos, el cual

permite realizar un controlar el desplazamiento de objetos en la pantalla y verificar que el objeto se encuentre en una posición predeterminada.

Sistema de Guardado: Esta capa controla la información y la conexión con la base de datos, entre sus funciones se encuentra el guardar la información y cargar la información.

Unity: Esta capa final contiene todas las funciones del motor de juegos Unity como: Sistema de colisión y física, control de audio, sistema de renderizado de imágenes, un sistema que controla eventos, un sistema de scripting para el manejo de código, un controlador de tiempo que permite controlar el tiempo que ha transcurrido y un sistema Canvas con el que se controla los menús.

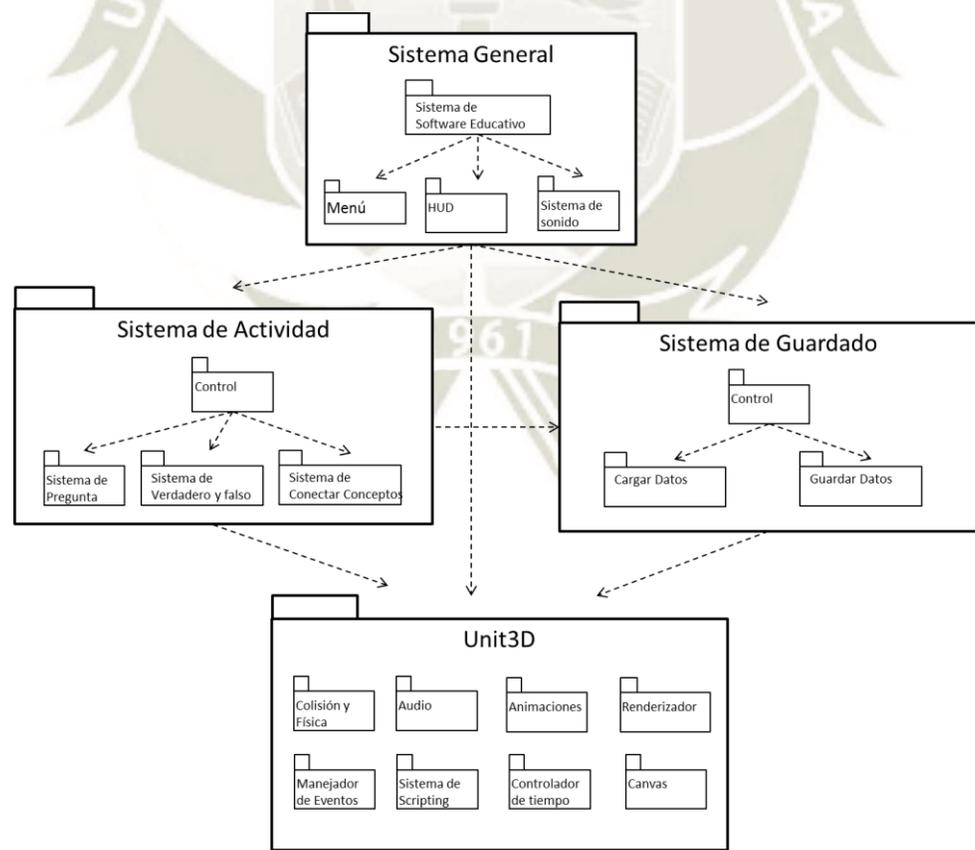


Figura 12. Vista lógica de la aplicación (Fuente: Elaboración Propia)

2.3.2 Vista Física del Sistema

La vista de despliegue que se observa en la Figura 13 muestra los componentes físicos del aplicativo, en este caso la aplicación está enfocada a la plataforma Android la cual depende del sistema operativo Linux, está codificada en Java donde las herramientas SDK compilan y empaquetan el código junto con los datos y los archivos de recursos necesarios en el archivo de archivo de la aplicación Android que tiene el sufijo .apk.

El archivo .apk representa una aplicación de Android que se implementará en los dispositivos móviles habilitados para Android.

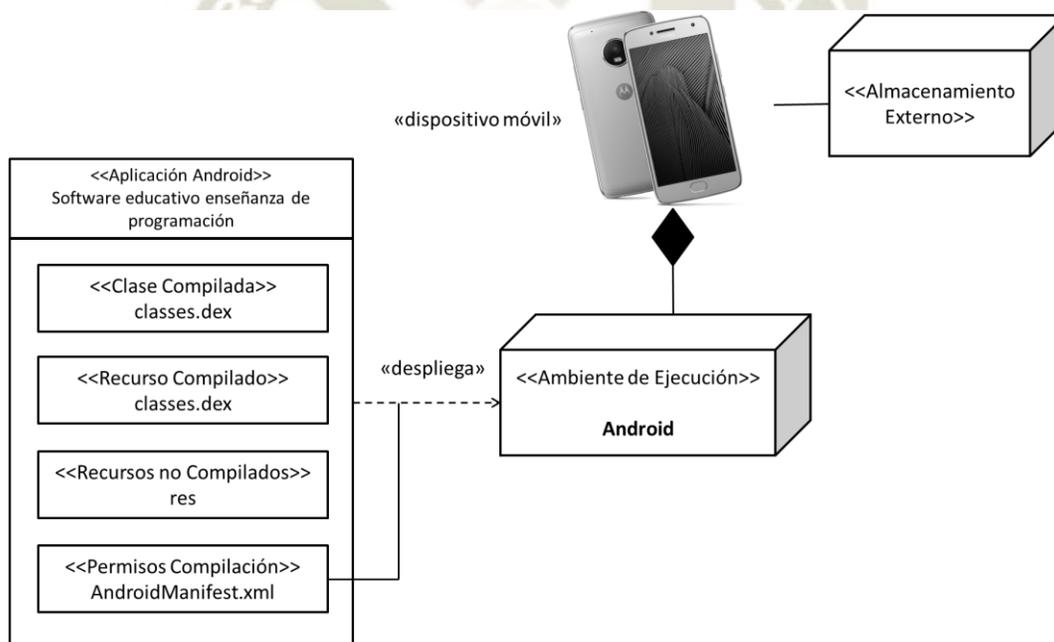


Figura 13. Vista de despliegue de la aplicación (Fuente: Elaboración Propia)

2.3.3 Vista de Procesos del Sistema

En esta sección se detalla por medio de figuras, los procesos para el ingreso de información del perfil del estudiante, la actividad de ver perfil, la actividad de pregunta, la actividad de verdadero o falso y la actividad de conectar conceptos.

1. Realizar el ingreso de perfil del estudiante

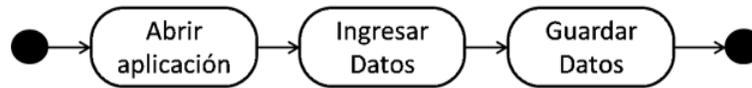


Figura 14. Diagrama de actividad ingreso de perfil (Fuente: Elaboración Propia)

2. Ver el perfil del estudiante



Figura 15. Diagrama de ver perfil (Fuente: Elaboración Propia)

3. Ver los logros del estudiante

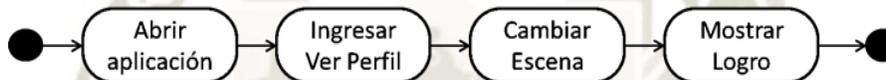


Figura 16. Diagrama de ver logros (Fuente: Elaboración Propia)

4. Seleccionar Actividad

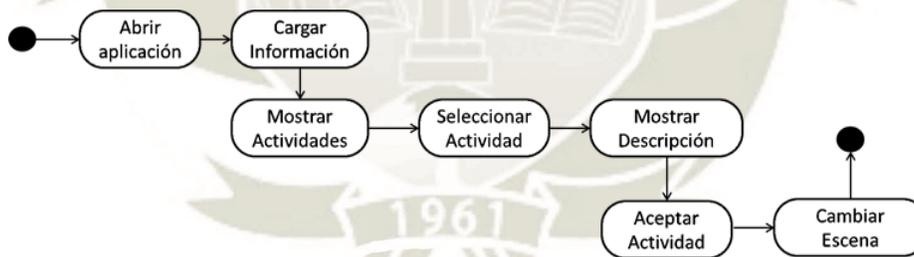


Figura 17. Diagrama de seleccionar actividad (Fuente: Elaboración Propia)

5. Realizar la actividad pregunta

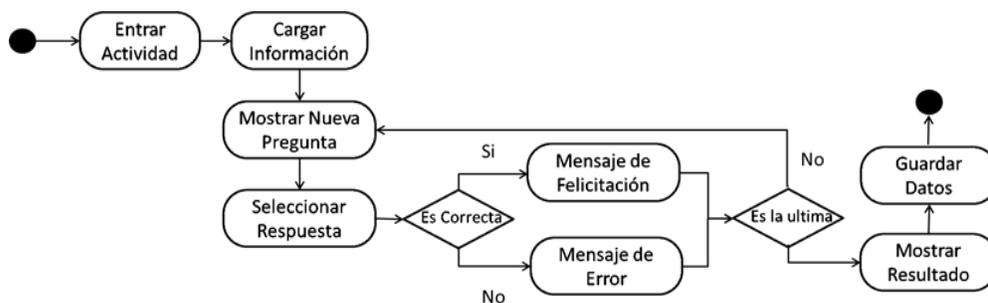


Figura 18. Diagrama de actividad pregunta (Fuente: Elaboración Propia)

6. Realizar la actividad verdadero o falso

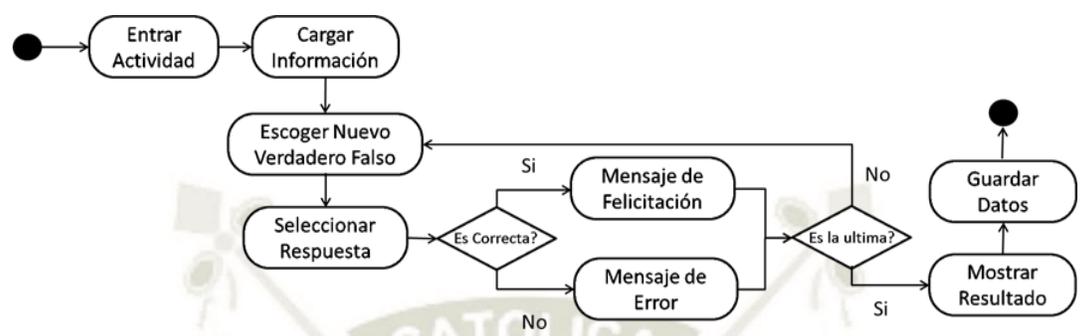


Figura 19. Diagrama de actividad verdadero o falso (Fuente: Elaboración Propia)

7. Realizar la actividad conectar conceptos

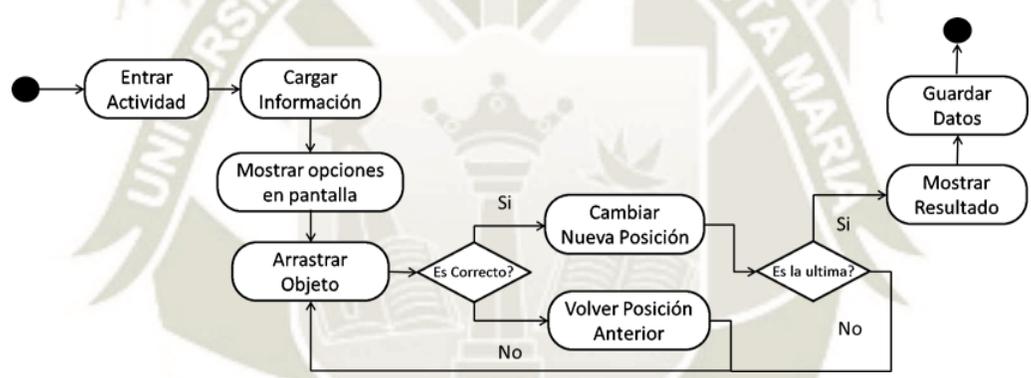


Figura 20. Diagrama de actividad conectar conceptos (Fuente: Elaboración Propia)

2.3.4 Comportamiento del Sistema

En esta sección se describe el comportamiento del sistema. Se muestra la interacción entre los objetos al usar el software educativo.

En la Figura 21 se muestra el diagrama de clases el cual contiene las clases Menú principal, Perfil, Logro, Actividad, Actividad Pregunta, Actividad Verdadero o Falso y Actividad Conectar Conceptos.

En la Figura 22 se puede ver el diagrama secuencia, cuando un estudiante entra por primera vez al aplicativo necesita registrar sus datos, estos serán guardados por el sistema, luego deberá acceder al menú principal desde ahí tendrá acceso a ver su perfil,

logros y las actividades a realizar; Una vez seleccione qué actividad realizar se cambia de escena y se comenzará a realizar la actividad al tener la actividad se guardará los datos y se dará la opción de ir a la siguiente actividad o volver al menú principal.

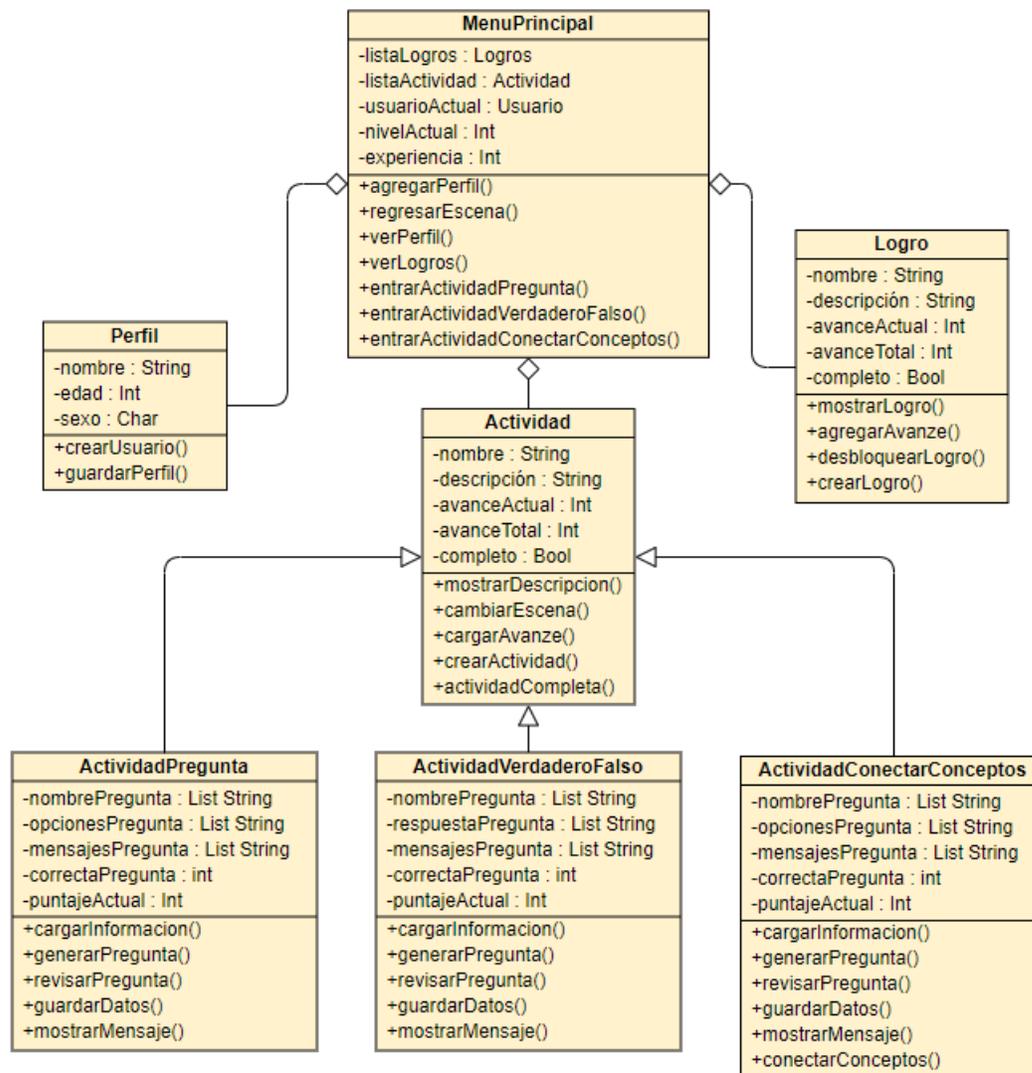


Figura 21. Diagrama de Clases (Fuente: Elaboración Propia)

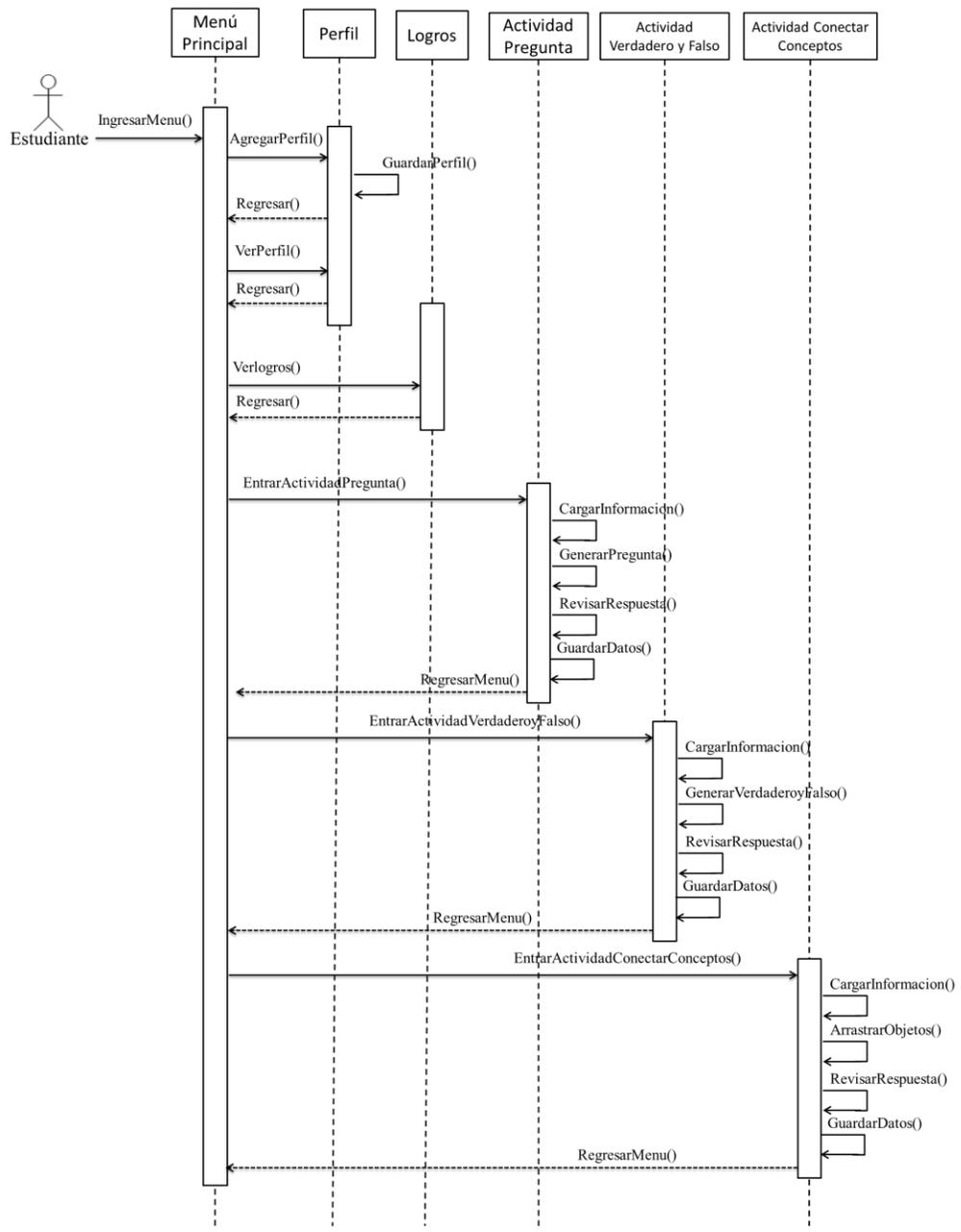


Figura 22. Diagrama de Secuencia (Fuente: Elaboración Propia)

2.3.5 Persistencia

Se muestra un modelo de tablas en la Figura 23 la cual es un nivel de abstracción más bajo donde se muestran las tablas donde se guardará la información del software educativo. Por ejemplo, los datos de perfil como el nombre, edad y orientación sexual,

experiencia, nivel, los cuales se guardarán en la tabla perfil, en este diagrama se buscan mostrar la estructura de los componentes de la aplicación.

La tabla de logro que contiene el nombre del logro, su descripción, el total para completar el logro y el nivel actual del logro.

Las tablas de actividad pregunta, actividad verdadero o falso y conectar conceptos contienen los campos de Correctos, Incorrectos y porcentaje para completar.

La tabla de dato pregunta contiene la información referente a las diferentes actividades como el nombre de la pregunta, actividad pregunta, los nombres de las opciones y la respuesta correcta que se mostrarán según la respuesta escogida.

La tabla de actividad verdadero o falso contiene la información referente al nombre de la pregunta, actividad verdadero o falso, la respuesta correcta y mensajes según la respuesta escogida.

Finalmente, la tabla de conectar conceptos contendrá información referente al título de la pregunta, actividad conectar conceptos, nombre de los diferentes conceptos y la descripción de los diferentes conceptos.

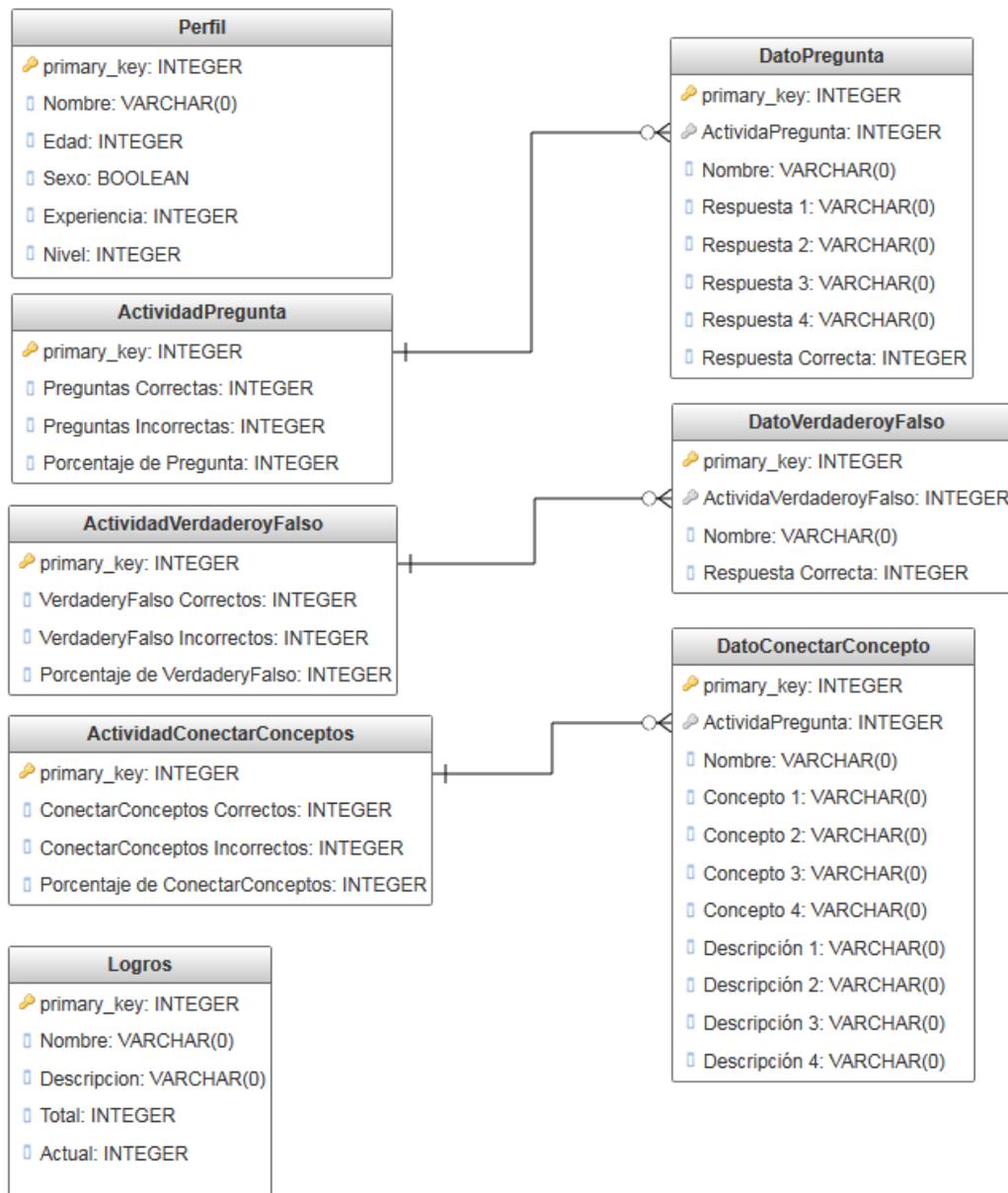


Figura 23. Modelo de Tablas (Fuente: Elaboración Propia)

2.3.6 Interfaz de Usuario

Se hace un diagrama de navegabilidad en el cual se muestra todos los detalles del software educativo, se muestra el orden que tiene el software educativo mediante una estructura jerárquica como se muestra en la Figura 24, solamente existe un usuario el cual

es el estudiante, el cual primero debe hacer su registro en la aplicación agregando su perfil, luego al entrar al menú puede ver su perfil, logros, realizar la actividad de pregunta, la actividad de verdadero o falso y la actividad de conectar conceptos.

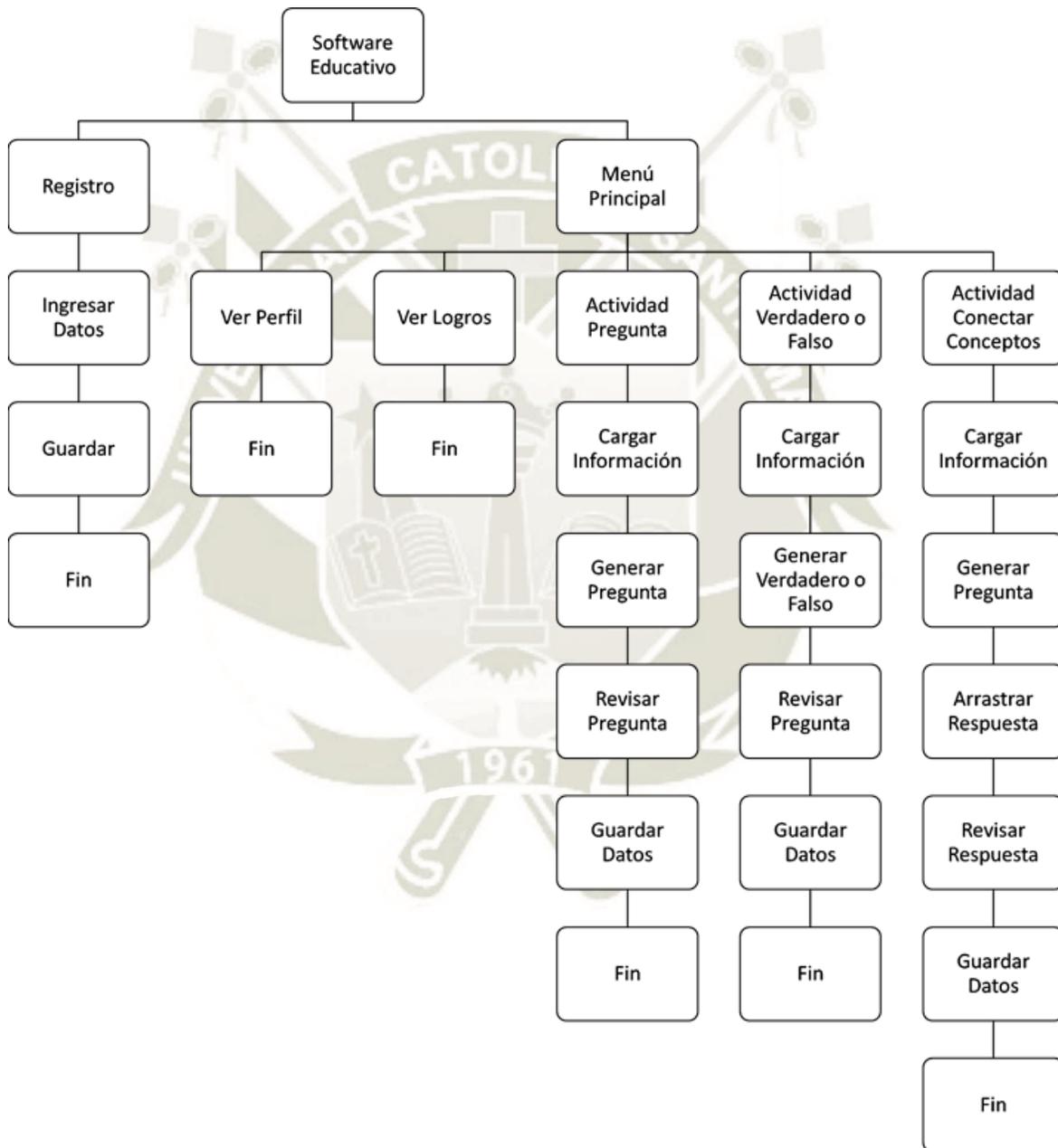


Figura 24. Diagrama de Navegabilidad (Fuente: Elaboración Propia)

2.4 Documentación técnica de programación

2.4.1 Hardware

Antes de entrar en la parte de programación, es necesario hacer mención a los principales componentes de computación involucrados en la elaboración del software educativo.

Laptop: La configuración de la computadora usada para realizar la programación es la siguiente:

- Marca: ASUS
- Modelo: G750JX
- CPU: Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz (8 CPUs)
- Sistema Operativo: Windows 8.1 64-bit
- Memoria: 16384 MB RAM
- Tarjeta de Video: NVIDIA GeForce GTX 770M

Este modelo de laptop ha sido escogido por su gran poder de procesamiento, con lo cual se logra un óptimo nivel de desempeño mediante la reducción de tiempos de carga, una simulación más fluida y una rápida respuesta para compilar el código y generar el aplicativo.

Dispositivos Móviles: El software educativo fue desarrollado para dispositivos móviles que sean del sistema operativo Android, las descripciones del dispositivo usado para las pruebas son:

- Sistema Operativo: Android
- Marca: Motorola

- Modelo: Moto G5
- Software: Android 7.0 Nougat
- Procesador: Snapdragon 430 octa-core 1,4 Ghz
- Ram: 2 GB

2.4.2 Motor de videojuegos

El motor es el responsable de la generación del software educativo, donde se ha buscado un motor que proporcione una correcta compilación a dispositivos móviles, el motor seleccionado para el desarrollo del aplicativo es Unity.

El motor de juegos Unity está desarrollado por Unity Technologies en Dinamarca, Unity integra un motor de renderizado personalizado con el motor de física nVidia PhysX y Mono con el cual se hace la implementación de código abierto de las bibliotecas .NET de Microsoft (Craighead, Burke, & Murphy, 2008).

Los beneficios de usar Unity son muchos cuando se los compara con otros motores de videojuegos, debido a que Unity brinda una mayor cantidad de soporte para el desarrollo de aplicaciones, viene incorporado con una tienda de recursos y cuenta con diferentes tutoriales para el aprendizaje del desarrollo de aplicaciones, las principales características que brinda Unity para el desarrollo del software educativo son las siguientes:

Documentación: El motor de Unity viene con una documentación completa con ejemplos, los cuales contienen videotutoriales y recursos completos para su desarrollo. Este es uno de los mayores beneficios de Unity y provee una mayor facilidad en su uso

en comparación con otros motores como Unreal o Source, que solamente proporcionan documentación parcial para usuarios que no pagan (Craighead et al., 2008).

Comunidad: Existe una comunidad activa de desarrolladores en línea que brindan soporte y ayuda a los nuevos usuarios de Unity, Los desarrolladores de Unity Technologies también agregan nuevas funciones al motor de videojuegos a petición de los usuarios, con lo cual ha permitido que Unity sea una herramienta que se adapta a diferentes desarrolladores, gracias esto Unity que en un inicio solamente daba soporte para el desarrollo de aplicaciones 3D, gracias a sus actualizaciones se ha vuelto en un motor de videojuegos que permite el desarrollo en 2D (Craighead et al., 2008).

Física y Renderización: Unity contiene un motor de física el cual le permite crear simulaciones realista dándoles propiedades a los objetos como resistencia a la masa, arrastre, elasticidad, rebote, velocidad y colisiones, estas propiedades se unen para una correcta simulación, pero para realizar estas simulaciones usan el motor PhysX de nVidia, el cual es usado por diferentes videojuegos. El renderizado es un proceso que permite generar una imagen o video mediante el cálculo de iluminación partiendo de un modelo 3D, entre las propiedades que se consiguen con el renderizado son el sombreado y asignación de textura la cual afecta a la apariencia de los objetos, para el renderizado Unity utiliza un lenguaje de sombreado simplificado que se compila en sombreadores DirectX 9 u OpenGL 2.0 dependiendo de la plataforma de destino (Craighead et al., 2008).

Multiplataforma: El editor del motor de Unity se ejecuta en OSX; sin embargo, las aplicaciones creadas con Unity se pueden compilar para OSX, Windows o páginas

web (que se ejecuta en un navegador web mediante un plugin, similar a Adobe Flash); no hay restricciones en la distribución de las aplicaciones creadas con Unity y debido a que las aplicaciones creadas con Unity son de propiedad del desarrollador, el programador tiene una mayor facilidad de compartir sus aplicaciones desarrolladas (Craighead et al., 2008).

Bajo Costo: Otra ventaja de Unity es su bajo costo, existe dos versiones de pago en Unity, plus en la cual se paga 35 dólares al mes y pro en la cual se realiza un pago de 125 dólares al mes. Pero para el desarrollo de la aplicación se está utilizando la licencia gratuita la cual contiene todas las funciones básicas de Unity las cuales permiten un completo desarrollo de la aplicación.

2.4.3 Codificación o scripts

Para el desarrollo de aplicaciones en Unity se usa un sistema de scripting el cual permite modificar los comportamientos de los objetos, crear interfaces, reconocimiento de eventos, entre otras acciones.

Unity permite el uso de tres tipos de lenguajes C#, JavaScript y Boo; cada uno tiene sus ventajas, se ha optado por el uso de lenguaje de C# debido a que es orientado a objetos, tiene una seguridad alta, existen diferentes ejemplos en la documentación oficial de Unity y el tesista ya ha tenido experiencia en el uso de este lenguaje

Los scripts usados en el proyecto permiten añadir comportamientos a cada uno de los objetos en la aplicación, estos scripts permiten cambiar tanto la parte lógica como la del diseño

Unity permite el uso de diferentes lenguajes en un mismo proyecto, pero debido a que eso dificulta al llamado entre scripts ya que un script que utiliza C# y otro que utilice JavaScript no pueden compartir funciones, se ha optado por usar solamente la codificación en C#.

En cuanto a los estándares de usados para la codificación, se ha optado por crear una arquitectura donde cada sistema tengo su script, con lo cual se busca facilitar el llamado entre script, para la codificación en los scripts de cada sistema se ha usado un patrón de diseño singleton el cual facilita el uso de scripts en sistemas.

En cuanto a los datos guardados se ha usado archivos serializados donde se crea un archivo el cual contiene toda la información necesaria.

2.4.3 Entorno de desarrollo

Para el desarrollo de aplicaciones en Unity es posible usar dos entorno de desarrollos como: MonoDevelop y Visual Studio.

MonoDevelop: permite un alto nivel de integración con Unity, permite la depuración, tiene una capacidad media de identificar errores de sintaxis y es de licencia libre.

Visual Studio: tiene un nivel bajo de integración con Unity, permite el autocompletado, tiene capacidad de identificar errores de sintaxis, permite la refactorización, permite el uso de plugins y necesita una licencia comercial.

Debido a las siguientes características se ha optado por el uso de MonoDevelop.

2.4.4 Implementación de la aplicación

Para el desarrollo de una aplicación con Unity es necesario primero la instalación del software, la cual se realiza desde su página oficial como se ve en la Figura 25.

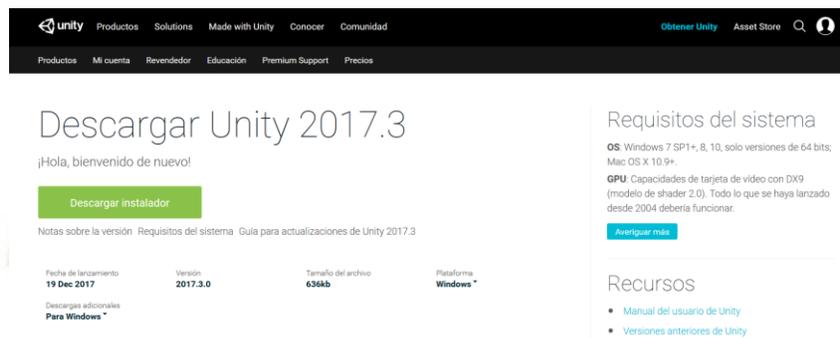


Figura 25. Descargar Unity (Fuente: Elaboración Propia)

Unity tiene diferentes versiones, se ha optado por usar la última versión estable de Unity la cual es 2017.3, luego de descargar Unity se escoge qué componentes instalar, los que se han escogido son: Unity 2017.3, MonoDevelop, documentación, recursos estándares, proyecto de ejemplo y soporte de Android como se ve en la Figura 26.

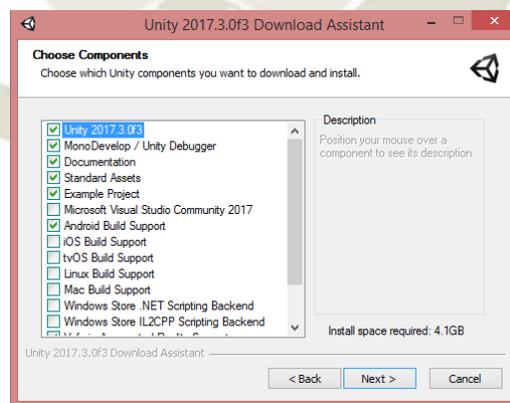


Figura 26. Instalación de Unity (Fuente: Elaboración Propia)

Otro punto importante para el desarrollo de un programa en Android es tener instalado el SDK (Software Development Kit), el cual es gratuito y se encuentra en la página oficial de Android como se ve en la Figura 27.

Get just the command line tools

If you do not need Android Studio, you can download the basic Android command line tools below. You can use the included [sdkmanager](#) to download other SDK packages.

These tools are included in Android Studio.

Platform	SDK tools package	Size	SHA-256 checksum
Windows	sdk-tools-windows-3859397.zip	132 MB (138,449,982 bytes)	7f6037d3a7d6789b4fdc06ee7af041e071e9860c51f66f7a4eb5913df9871fd2
Mac	sdk-tools-darwin-3859397.zip	82 MB (86,182,133 bytes)	4a81754a760fce88cba74d69c364b05b31c53d57b26f9f82355c61d5fe4b9df9
Linux	sdk-tools-linux-3859397.zip	130 MB (136,964,098 bytes)	444e22ce8ca0f67353bda4b85175ed3731cae3ffa695ca18119cbacef1c1bea0

Figura 27. Descargar SDK (Fuente: Elaboración Propia)

Luego de haber descargado el SDK es necesario instalar los paquetes correspondientes a la API que se desea usar, esto se refiere a cuál será la versión mínima soportada en Android, también es necesario descargar los controladores para USB como se ve en la Figura 28.

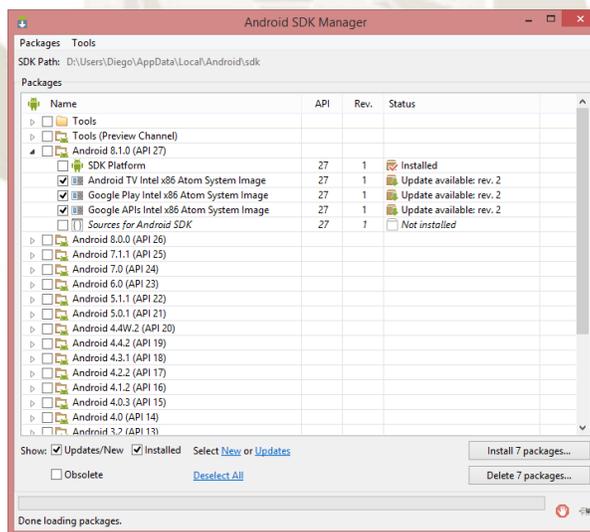


Figura 28. Android SDK Manager (Fuente: Elaboración Propia)

Otro componente es el JDK (Java Development Kit) que es un software que provee herramientas de desarrollo para la creación de programas en Java el cual se descarga en la página oficial de Oracle como se ve en la Figura 29

Java SE Development Kit 8u161		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.92 MB	jdk-8u161-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.88 MB	jdk-8u161-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.96 MB	jdk-8u161-linux-i586.rpm
Linux x86	183.76 MB	jdk-8u161-linux-i586.tar.gz
Linux x64	166.09 MB	jdk-8u161-linux-x64.rpm
Linux x64	180.97 MB	jdk-8u161-linux-x64.tar.gz
macOS	247.12 MB	jdk-8u161-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.99 MB	jdk-8u161-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.29 MB	jdk-8u161-solaris-sparcv9.tar.gz
Solaris x64	140.57 MB	jdk-8u161-solaris-x64.tar.Z
Solaris x64	97.02 MB	jdk-8u161-solaris-x64.tar.gz
Windows x86	198.54 MB	jdk-8u161-windows-i586.exe
Windows x64	206.51 MB	jdk-8u161-windows-x64.exe

Figura 29. Descargar JDK (Fuente: Elaboración Propia)

Luego de haber descargado los diferentes componentes, es necesario agregar las ubicaciones de dichos componentes en Unity, se realiza de forma automática si se han instalado antes descargar Unity, caso contrario se debe entrar a Edit luego a Preferences, de ahí se selecciona la pestaña External Tools y se coloca la ubicación de SDK y JDK como se observa en la Figura 30.

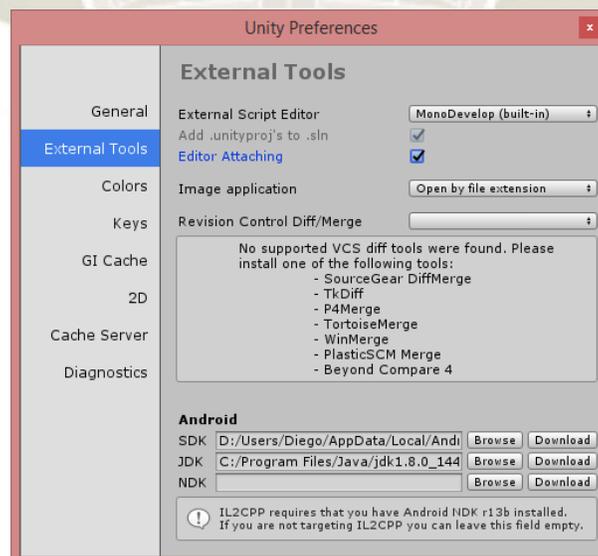


Figura 30. Preferencias de Unity – Herramientas Externas (Fuente: Elaboración Propia)

Para la creación de la interfaz se ha utilizado como referencia la interfaz de Duolingo como se observa en la Figura 3, para la búsqueda de recursos se ha utilizado la Asset Store como se observa en la Figura 31, la cual contiene diferentes recursos gratuitos como: Caracteres, ambientes, letras, iconos, texturas, materiales, proyectos completos, entre otros.

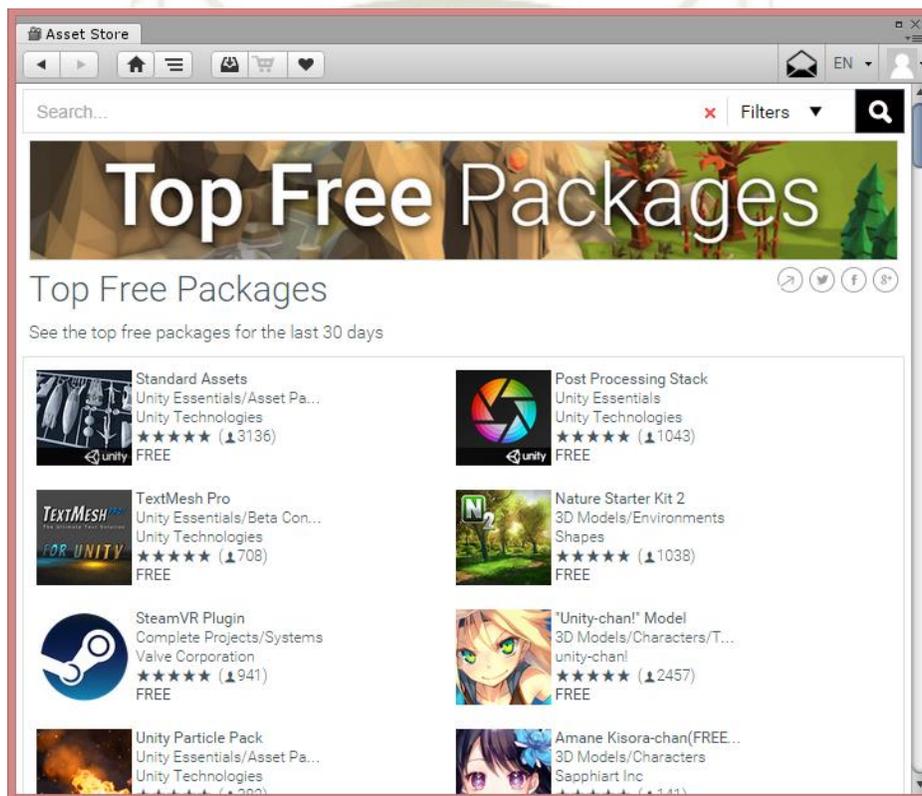


Figura 31. Asset Store (Fuente: Elaboración Propia)

Luego de haber instalado los programas, se procede al desarrollo de proyecto, La interfaz de Unity es intuitiva y existen diferentes tutoriales sobre su uso, para comenzar a realizar la aplicación primero se debe optar por un nombre el cual hemos escogido como Software Educativo, luego se debe seleccionar el tipo de proyecto 2D o 3D, esto determina el tipo de cámara por defecto que se usará para el proyecto y cuando se

importe imágenes determinar si serán sprites o materiales, todos estos cambios no afectan a la funciones de Unity solamente en las características por defecto, se ha optado por seleccionar 2D, luego se pueden agregar los paquetes se desean agregar y finalmente se hace clic en crear proyecto, como se observa en la Figura 32.

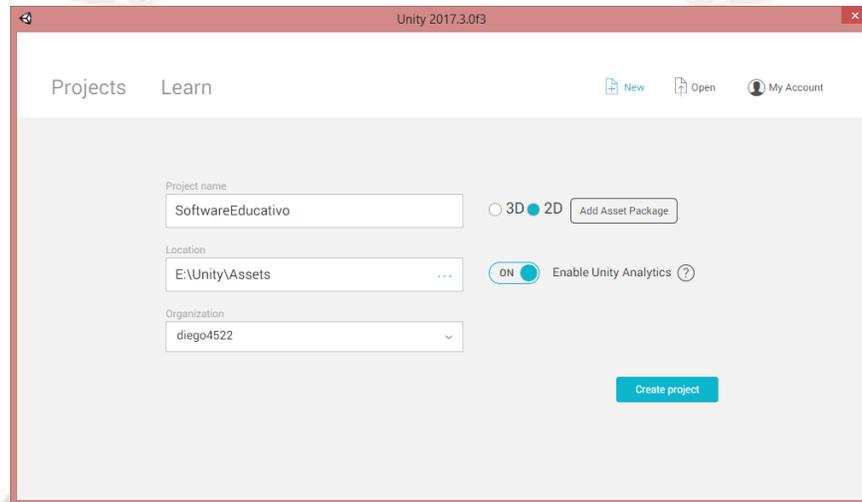


Figura 32. Creación de nuevo proyecto en Unity (Fuente: Elaboración Propia)

Una vez creado el proyecto se ingresa a la interfaz del editor de Unity la cual contiene diferentes vistas como se observa en la Figura 33.

Las vistas que tienen el proyecto son:

- **Escena:** Vista principal del juego, la cual permite mover, girar y cambiar la escala de los diferentes objetos.
- **Juego:** Vista cuando reproduce el juego, la cual permite realizar una simulación y poder usar diferentes tamaños de pantallas para poder apreciar cómo se verá la aplicación al ser compilada.
- **Jerarquía:** listado de los objetos presentes en la escena actual, permite la creación de padres e hijos entre dichos objetos.

- **Proyecto:** se puede explorar los diferentes archivos pertenecientes al proyecto, se usa para poder crear carpetas y mover los elementos.
- **Inspector:** Muestra las propiedades de cada objeto, es posible arrastrar script desde la ventana proyecto a la pantalla inspector.
- **Consola:** Está ubicada en la pestaña cercana a proyecto, se envían mensajes de errores y advertencias para reconocer donde ha ocurrido un problema.



Figura 33. Interfaz de Unity (Fuente: Elaboración Propia)

Ahora se verán otras herramientas que se usaron en conjunto con el programa Unity para el desarrollo del aplicativo, para la edición de imágenes se usó la herramienta de Gimp que es un programa libre para la manipulación de imágenes, Gimp trabaja mediante el uso de capas para separar las imágenes, su interfaz se puede observar en la Figura 34.

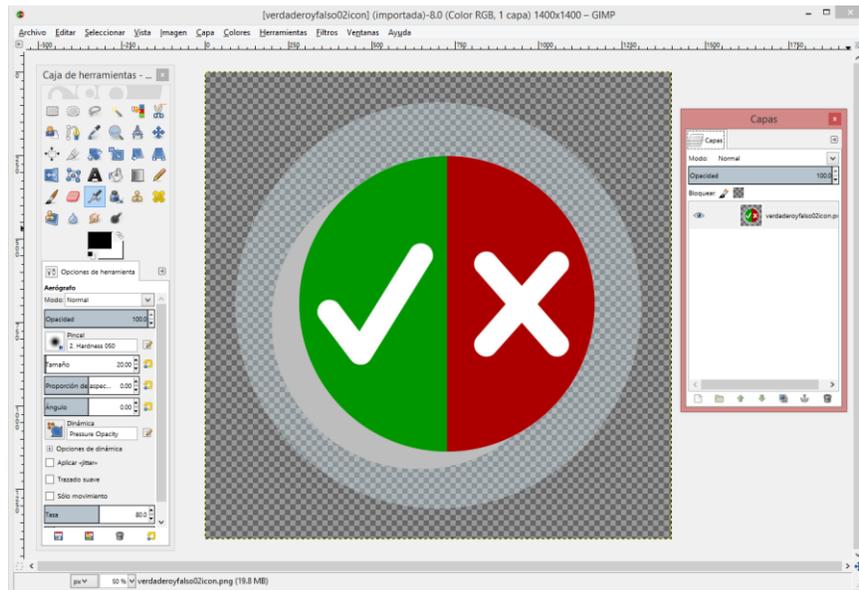


Figura 34. Interfaz de Gimp (Fuente: Elaboración Propia)

Otra herramienta que se ha usado es Audacity que es una herramienta libre para editar sonidos, sea usado para amplificar los sonidos y reducir el ruido, su interfaz se puede observar en la Figura 35.

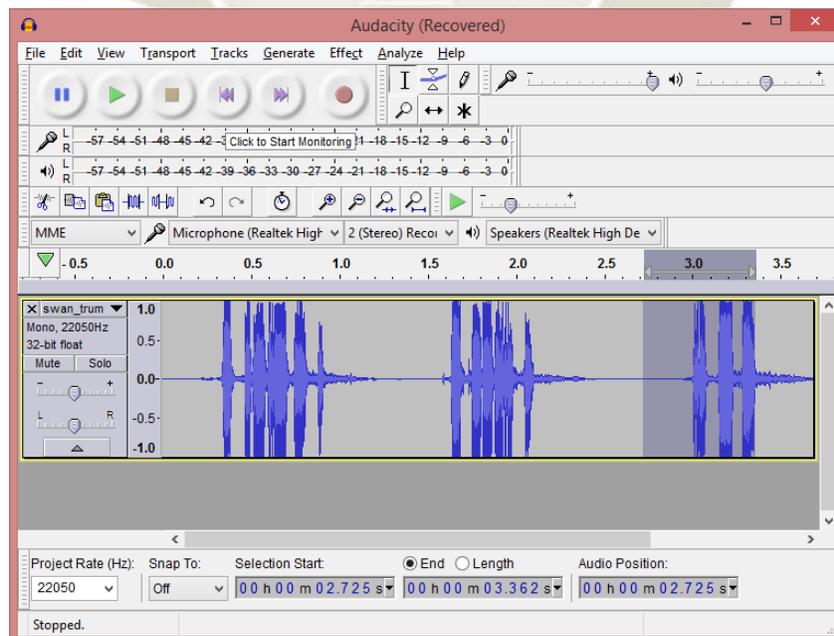


Figura 35. Interfaz de Audacity (Fuente: Elaboración Propia)

Para el desarrollo de un programa en Unity es necesario codificar las acciones a realizar lo cual se consigue mediante scripts, los cuales se crean en la ventana de proyecto haciendo clic derecho seleccionando create y luego C# Script como se muestra en la Figura 36.

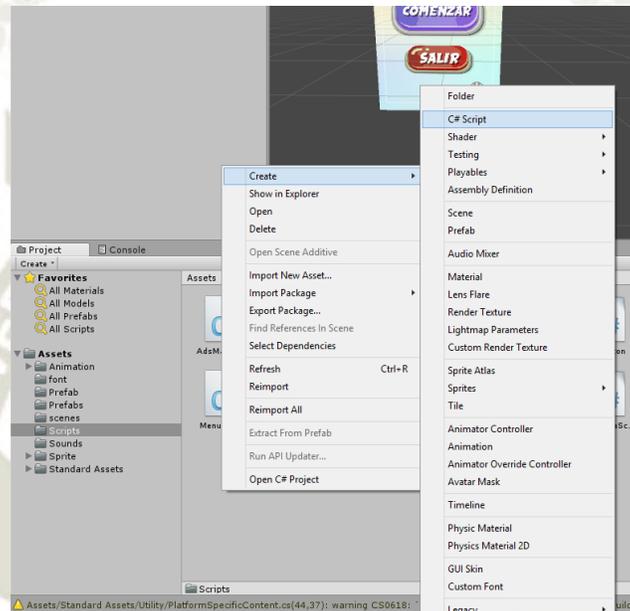


Figura 36. Creación de un script (Fuente: Elaboración Propia)

En Unity existe una serie de términos con los cuales se les nombra a los diferentes componentes y objetos en Unity los cuales son los siguientes:

Escena: cada escena es un conjunto de objetos que forma el ambiente de un juego, cada escena es guardada como un archivo único, es necesario el guardar de forma constante cada escena para que se mantenga de forma actualizada, es necesario escoger una ubicación en la vista de proyecto para determinar la ubicación del archivo de escena, finalmente para hacer el cambio de escena se hace en el código pero es necesario agregar la escena a la lista de escenas que se construirán al crear un juego, lo cual se hace

haciendo clic en file, luego build setting y se abre una ventana como se mira en la Figura 37, después es necesario arrastrar las escenas de la vista proyecto.

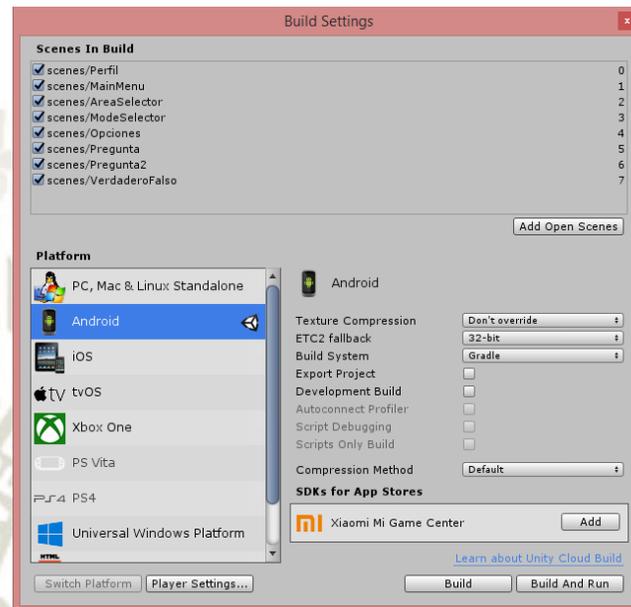


Figura 37. Ventana de Build Settings (Fuente: Elaboración Propia)

GameObjects: es cada objeto que se encuentra en la escena como personajes, cámaras o efectos especiales, tiene un conjunto de propiedades que le permite determinar sus acciones y atributos, para convertir un GameObject a una cámara, árbol o luz es necesario agregar componentes según lo necesario, todos estos componentes son agregados en la vista inspector, los GameObject están ubicados en la vista jerarquía, es posible volver a un GameObject padre de otro GameObject arrastrándolo.

Inputs: se refiere a los dispositivos de entrada como controladores, pantalla táctil, joypad, teclado y ratón; de los cuales se reconoce cuando se hace clic o se presiona botones.

Cámara: cada escena necesita una cámara la cual permite establecer que se mira en cada escena, en la vista juego se observa lo que mira la cámara.

Luego de haber creado el script es necesario nombrarlo lo cual se realiza después de haber sido creado, después se abre el script para lo cual es necesario hacerle clic dos veces con lo cual se inicializa MonoDevelop como se observa en la Figura 38.

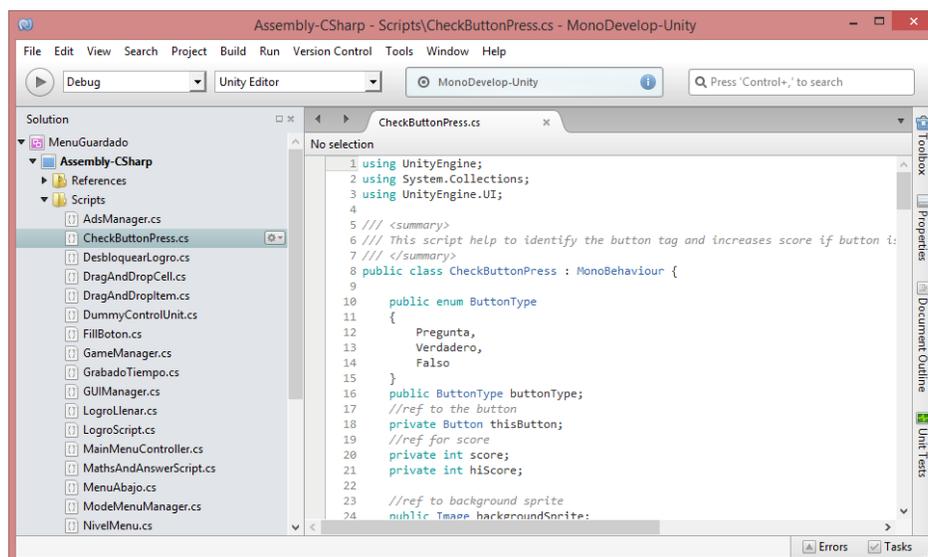


Figura 38. Interfaz de MonoDevelop (Fuente: Elaboración Propia)

Cuando se crea un script por primera vez se agregan dos funciones por defecto Start() la cual es llamada cuando se inicializa una escena con lo cual se puede agregar valores iniciales a las variables y Update() la cual es una función que se llama en cada frame que sirve como un eje central para todos los comportamientos que se desean agregar en los juegos.

La interfaz de Unity permite el agregar componentes de manera “Drag & Drop”, es decir permite el arrastrar componentes desde la ventana de proyecto a la ventana de inspector, adicionalmente también se puede agregar haciendo clic en agregar componente y luego buscar el script según el nombre con el que se ha guardado.

Luego de haber creado las diferentes escenas y haber culminado con el desarrollo del software se procede a la generación de APK, se realiza en la ventana de Build

Settings donde es necesario especificar la plataforma en la cual se va crear el ejecutable, luego seleccionamos Android como la plataforma, después se debe hacer clic en Player Settings se cambia el nombre del proyecto y luego el nombre del paquete por un nombre que represente a tu aplicación como se ve en la Figura 39.

Finalmente, se hace clic en Build And Run para instalar el aplicativo en el dispositivo, hay que tomar en cuenta que es necesario entrar como desarrollador en el celular para permitir una depuración por USB, también es necesario descargar los plugins correspondientes para poder conectar el celular en la computadora.

Caso contrario se puede hacer clic en Build y generar un archivo Apk el cual luego puede ser instalado a un celular dándole los permisos de instalar de fuentes desconocidas.

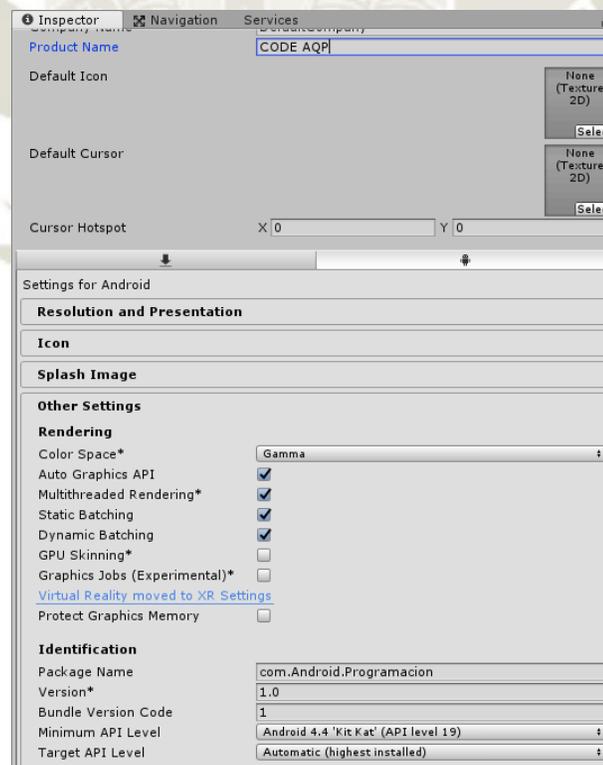


Figura 39. Ventana de Player Settings (Fuente: Elaboración Propia)

2.5 Pruebas de Ejecución

En esta sección se describe un conjunto de pruebas para demostrar el correcto funcionamiento del software educativo, donde se describe a las pruebas y cómo se utilizan.

2.5.1 Pruebas funcionales

Se determinó el uso de la caja negra como la prueba funcional para el software educativo, esto es debido a que este tipo de prueba se suele usar en videojuegos, para el uso de esta prueba se elaboró una lista de acciones para verificar, donde se pide a personas ajenas al desarrollo del software que evalué el aplicativo siguiendo una lista de verificación marcando un sí o no según corresponda como se ve en la Tabla 14, luego de realizar esta prueba se hicieron correcciones de las actividades faltantes.

Tabla 14. Tabla de verificación de la caja negra (Fuente: Elaboración Propia)

Id	Sistema	Verificar	Resultado
1	Menú	Se muestra la pantalla de menú	Si
2	Menú	Se puede ingresar la información referente al nombre	Si
3	Menú	Se puede ingresar la información referente a la edad	Si
4	Menú	Se puede ingresar la información referente al sexo	Si
5	Menú	Se puede navegar entre las pantallas	Si
6	Menú	Se muestra un mensaje cuando se cambia la escena	Si
7	Menú	Se muestra un mensaje cuando se entra a una actividad	Si
8	Menú	Se puede seleccionar la actividad que deseas entrar	Si
9	Menú	Se puede regresar al menú principal	Si
10	Menú	Se genera un mensaje cuando se complete un logro	Si
11	HUD	Se muestra el avance de las actividades	Si
12	HUD	Se puede ver el nivel actual del estudiante	No
13	HUD	Se muestra la experiencia del estudiante	No
14	HUD	Se puede ver el perfil del estudiante	Si
15	HUD	Se puede ver los logros del estudiante	Si
16	Sonido	Se puede escuchar la música de fondo	Si
17	Sonido	Se puede escuchar los efectos de sonido	Si
18	Sonido	Se puede regular el volumen de la música	No

19	Sonido	Se puede silenciar la música	Si
20	Sonido	Se escucha cuando se hace clic en los botones	Si
21	Guardado	Se puede guardar la información del perfil	Si
22	Guardado	Se puede guardar la información de los logros	Si
23	Guardado	Se guarda las preferencias de sonido	Si
24	Guardado	Se puede guardar la experiencia obtenida	Si
25	Guardado	Se puede guardar el progreso obtenido	Si
26	Guardado	Se puede cargar información del perfil	Si
27	Guardado	Se puede cargar información de los logros	Si
28	Guardado	Se puede cargar las preferencias de sonido	Si
29	Guardado	Se puede cargar la experiencia obtenida	Si
30	Guardado	Se puede cargar el progreso obtenido	Si
31	Actividad	Se muestra la pantalla de la actividad Pregunta	Si
32	Actividad	Se muestra la pantalla de la actividad Verdadero o Falso	Si
33	Actividad	Se muestra la pantalla de la actividad Conectar Conceptos	Si
34	Actividad	Se muestra el título de la pregunta	Si
35	Actividad	Se cambia los botones según la pregunta	Si
36	Actividad	Se muestra un mensaje si se responde de forma correcta	Si
37	Actividad	Se muestra un mensaje si se responde de forma incorrecta	No
38	Actividad	Se puede cambiar a la siguiente actividad	Si
39	Actividad	Se puede mostrar el avance de la actividad	Si
40	Actividad	Existe animaciones en la actividad	No

2.5.2 Pruebas de Usabilidad

Se usaron diferentes pruebas de usabilidad para determinar el nivel de usabilidad del software educativo, se definió una lista de acciones a realizar y basándonos en los resultados de dichas pruebas se hicieron modificaciones al sistema para corregir los errores, como se observa en la Tabla 15.

Tabla 15. Tabla de pruebas de usabilidad (Fuente: Elaboración Propia)

Prueba PU-01	
Acción	Realizar el ingreso del perfil
Puntos a favor	<ol style="list-style-type: none"> 1. Se verifica que solamente se ingrese datos numéricos en edad 2. Se verifica que solamente se pueda ingresar letras en el nombre
Puntos en contra	<ol style="list-style-type: none"> 1. Colocar un control para campos vacíos en edad 2. Colocar un control para campos vacíos en nombre
Prueba PU-02	
Acción	Ver el perfil del estudiante
Puntos a favor	<ol style="list-style-type: none"> 1. Se mira el nivel del estudiante 2. Se muestra una imagen para representar el género 3. Los colores contrastan con el fondo
Puntos en contra	<ol style="list-style-type: none"> 1. Se debe cambiar la letra por otro tipo más legible 2. Se debe mostrar el avance para llegar al siguiente nivel
Prueba PU-03	
Acción	Ver logros de estudiantes
Puntos a favor	<ol style="list-style-type: none"> 1. Se mira una correcta descripción de los logros 2. Los iconos diferentes ayudan a distinguir los diferentes logros 3. Los colores facilitan distinguir los logros
Puntos en contra	<ol style="list-style-type: none"> 1. Colocar un avance para completar los logros 2. Solo mostrar el icono cuando se encuentra desbloqueado el logro
Prueba PU-04	
Acción	Seleccionar una actividad
Puntos a favor	<ol style="list-style-type: none"> 1. Tienen iconos diferentes según el tipo de actividad. 2. Existe un sonido al hacer clic en los botones. 3. Se muestra un avance de la actividad
Puntos en contra	<ol style="list-style-type: none"> 1. Colocar una descripción al hacer clic a la actividad 2. Se debe mostrar niveles según la dificultad de las actividades 3. Cambiar el desplazamiento del menú de vertical a horizontal
Prueba PU-05	
Acción	Realizar la actividad de pregunta
Puntos a favor	<ol style="list-style-type: none"> 1. El título se adapta de manera correcta según la pantalla 2. Se muestra una barra de progreso de la actividad 3. Se hace una correcta reproducción de sonido de fondo 4. Se adapta las respuestas de manera correcta 5. Se muestra el logro cuando se completa la actividad
Puntos en contra	<ol style="list-style-type: none"> 1. No muestra mensajes cuando se equivoca. 2. Colocar un botón para regresar al menú principal.
Prueba PU-06	
Acción	Realizar la actividad de verdadero o falso
Puntos a favor	<ol style="list-style-type: none"> 1. Los colores diferentes ayudan a reconocer de manera rápida la opción de verdadero o falso.

	<ol style="list-style-type: none"> 2. Se genera sonidos correctos según los botones presionados. 3. Se muestra el logro cuando se completa la actividad.
Puntos en contra	<ol style="list-style-type: none"> 1. Colocar un botón para regresar al menú principal. 2. Se debe agregar un efecto cuando se seleccione una pregunta.
Prueba PU-07	
Acción	Realizar la actividad de conectar conceptos
Puntos a favor	<ol style="list-style-type: none"> 1. Se desplaza de manera correcta los conceptos. 2. Se regresa de manera correcta a su posición original si se ha cometido un error. 3. La descripción y conceptos se leen de manera correcta.
Puntos en contra	<ol style="list-style-type: none"> 1. Agregar sonidos cuando se arrastra y suelta el objeto. 2. Se debe bloquear las respuestas ya contestadas.

2.5.3 Pruebas de Aceptación

Se realizaron pruebas de aceptación las cuales ayudan a determinar si se consiguió realizar un correcto flujo de las funciones del proyecto, se determinaron una serie de acciones y verificamos si se cumple su correcto flujo, se observa las diferentes pruebas en la Tabla 16.

Tabla 16. Tabla de pruebas de aceptación (Fuente: Elaboración Propia)

Prueba PA-01	
Objetivo	Probar el funcionamiento de “Realizar el ingreso de perfil del estudiante”.
Precondición	Haber instalado la aplicación.
Descripción de la prueba	Ingresar los datos como nombre, edad y sexo
Resultados esperados	Se realizó de manera correcta el ingreso de datos y se accedió a la pantalla de menú principal
Prueba PA-02	
Objetivo	Probar el funcionamiento de “Ver el perfil del estudiante”.
Precondición	Haber ingresado información sobre el perfil
Descripción de la prueba	Luego de haber ingresado al menú principal el estudiante entra a ver su perfil.
Resultados esperados	Se muestra en la pantalla la información referente al estudiante.
Prueba PA-03	
Objetivo	Probar el funcionamiento de “Ver los logros del estudiante”.
Precondición	Haber ingresado información sobre el perfil

Descripción de la prueba	Luego de haber ingresado al menú principal el estudiante entra a ver sus logros.
Resultados esperados	Se muestra en la pantalla la información referente a los logros del estudiante.

Prueba PA-04

Objetivo	Probar el funcionamiento de “Seleccionar Actividad”.
Precondición	Haber ingresado información sobre el perfil
Descripción de la prueba	Luego de haber ingresado al menú principal el estudiante selecciona la actividad realizar.
Resultados esperados	Se muestra una pantalla con la descripción de la actividad y luego cambia de escena a la actividad seleccionada.

Prueba PA-05

Objetivo	Probar el funcionamiento de “Realizar la actividad pregunta”.
Precondición	Haber seleccionado la actividad pregunta.
Descripción de la prueba	Se selecciona la respuesta correcta, luego se vuelve a mostrar una nueva pregunta, después de completar todas las preguntas muestran un mensaje para ir a la siguiente actividad o volver al menú principal
Resultados esperados	Se muestra el puntaje obtenido, se desbloquea los logros obtenidos y luego se cambia de escena

Prueba PA-06

Objetivo	Probar el funcionamiento de “Realizar la actividad verdadero o falso”.
Precondición	Haber seleccionado la actividad pregunta.
Descripción de la prueba	Se selecciona la respuesta correcta, luego se vuelve a mostrar una nueva pregunta, después de completar todas los verdaderos y falso muestran un mensaje para ir a la siguiente actividad o volver al menú principal
Resultados esperados	Se muestra el puntaje obtenido, se desbloquea los logros obtenidos y luego se cambia de escena.

Prueba PA-06

Objetivo	Probar el funcionamiento de “Realizar la actividad conectar conceptos”.
Precondición	Haber seleccionado la actividad pregunta.
Descripción de la prueba	Se arrastra los conceptos a los lugares predeterminados, luego de haber arrastrado todos los conceptos se muestra un mensaje para ir a la siguiente actividad o volver al menú principal.
Resultados esperados	Se muestra el puntaje obtenido, se desbloquea los logros obtenidos y luego se cambia de escena.

2.6 Manuales de Usuario

Objetivo, Requisitos e instalación, Descripción del funcionamiento del sistema

Se ha determinado el uso de una

2.6.1 Objetivo

El objetivo de este manual es proporcionar al estudiante la información necesaria para utilizar la aplicación enfocada en la enseñanza de programación mediante el uso de técnicas de gamification.

Es aconsejable leer los requerimientos bases y las funciones del programa antes de su uso.

Las funciones bases del sistema son:

- El **ingreso de datos del perfil**, con lo cual se ingresa información personal del estudiante como: nombre, edad y orientación sexual.
- El **ver perfil**, donde el estudiante puede observar los datos generales del estudiante.
- El **ver logros**, contiene la información referente a los logros del jugador con lo cual se puede saber los logros obtenidos y el porcentaje pendiente para cumplir los logros.
- El **entrar a actividad**, con lo cual se navega en la sección de menú con el objetivo de entrar a la actividad deseada.
- El **resolver una pregunta**, que está compuesta con todos los pasos necesarios para completar la actividad de preguntas.

- El **resolver verdadero o falso**, contiene los diferentes pasos para completar la actividad de verdadero o falso.
- El **resolver conectar conceptos**, en esta actividad el alumno tiene que desplazar los objetos mostrados en la pantalla hasta completar la actividad.

2.6.2 Requisitos e instalación

Para el correcto funcionamiento del software educativo es necesario tener una serie de requisitos, tanto de hardware como de software.

Los requisitos previos a la instalación de software son:

- Dispositivo Android
- Nivel de API 19 'KitKat' o superior

Los requisitos mínimos de hardware serían los siguientes:

- Ram 2 GB o superior
- Procesador Snapdragon 430 octa-core a 1,4 GHz o superior
- Almacenamiento disponible para la instalación.

2.6.3 Descripción del funcionamiento del sistema

Instalación de la aplicación

Para comenzar con la instalación del programa primero es necesario copiar el APK, el cual es un paquete para el sistema operativo Android con el cual se puede instalar aplicaciones, se debe permitir la instalación de fuentes desconocidas.

Primero se debe buscar el archivo, luego es necesario hacer clic en él, de ahí se debe confirmar la instalación y después de eso se creará la aplicación como se ve en la Figura 40.

Luego de haber instalado la aplicación se creará un icono con el cual se accede a la aplicación, el icono es de la misma forma que el icono del archivo APK.

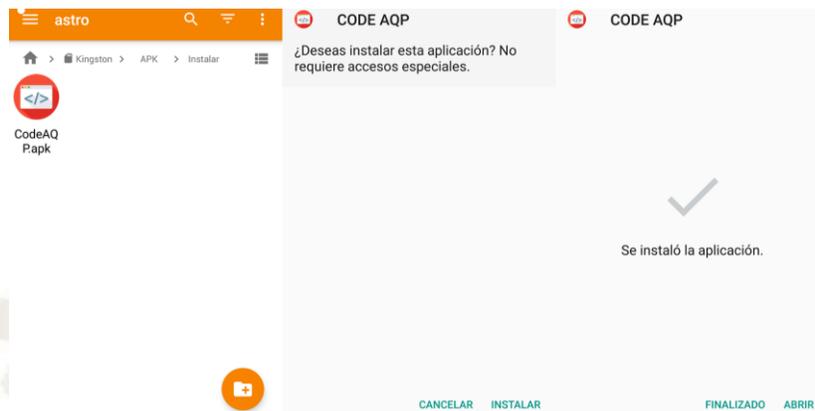


Figura 40. Instalación de la aplicación (Fuente: Elaboración Propia)

Ingreso de datos del perfil

Cuando se inicia por primera vez la aplicación, se muestra una pantalla para ingresar los datos sobre el estudiante como nombre, edad y orientación sexual como se muestra en la Figura 41.

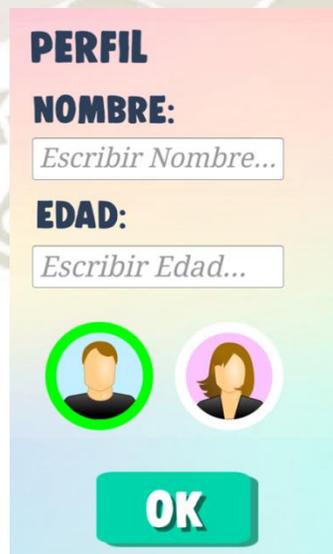


Figura 41. Pantalla principal del ingreso de datos del perfil (Fuente: Elaboración Propia)

Primero se debe ingresar el nombre y la edad, haciendo clic en la caja de texto, en el nombre se permite el ingreso de todos los caracteres, pero para la edad solamente se permite el ingreso de números como se muestra en la Figura 42.

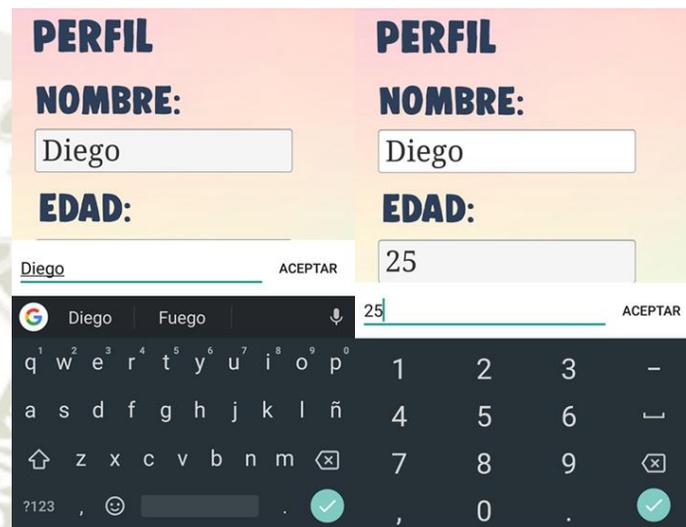


Figura 42. Ingreso de nombre y edad (Fuente: Elaboración Propia)

Finalmente, se determina la orientación sexual, haciendo clic en el símbolo correspondiente, ya sea de hombre o mujer; seguidamente se hace clic en el botón ok como se muestra en la Figura 43.

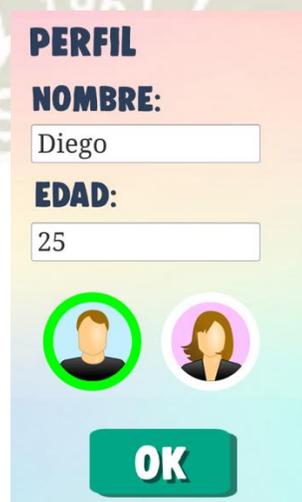


Figura 43. Selección de la orientación sexual (Fuente: Elaboración Propia)

Ver perfil

Luego de haber ingresado la información referente al perfil se entra en la pantalla del menú principal en la cual se muestra el nivel actual del jugador, su porcentaje para completar la actividad, como se observa en la Figura 44, seleccionamos la opción de perfil que es la silueta de una persona que se encuentra en la parte inferior central para entrar a la escena de perfil, como se ve en la Figura 44 donde nos brinda la información personal del usuario.



Figura 44. Menú principal de la aplicación y Perfil del alumno (Fuente: Elaboración Propia)

Ver logros

Luego de haber visto la información referente al perfil para poder entrar en la escena de logros, seleccionamos la opción de logro que es la silueta de un trofeo que se encuentra en la parte inferior derecha, como se ve en la Figura 45 donde nos brinda la

información de los diferentes logros como el título del logro, el avance del logro, una descripción del logro, una barra de avance del logro, y una imagen referente al logro.

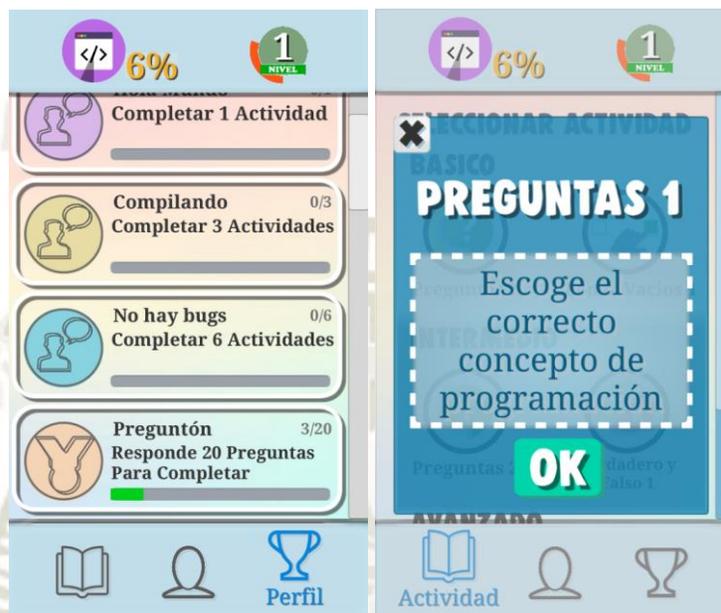


Figura 45. Logros del alumno y Descripción de la actividad preguntas (Fuente: Elaboración Propia)

Entrar a actividad

Después de haber entrado en la escena de logros para poder volver en la escena de menú principal y poder seleccionar una actividad, seleccionamos la opción de actividad que es la silueta de un libro se encuentra situado en la parte inferior izquierda, luego de haber cambiado la escena se busca la actividad que se desea realizar, la imagen de los botones se va coloreando según su avance como se muestra en la Figura 44, ahora se debe hacer clic en la actividad que se desea realizar para tener una descripción más específica de la actividad como se observa en la Figura 45, luego se hace clic en el botón Ok para entrar en la actividad seleccionada, con lo cual se carga una pantalla de cargando como se muestra en la Figura 46.



Figura 46. Pantalla de cargando para el cambio de escena e Inicio de la actividad de pregunta

(Fuente: Elaboración Propia)

Resolver una pregunta

Después de haber seleccionado la actividad de pregunta, se muestra una pantalla inicial en la cual se indica la acción a realizar como se observa en la Figura 46, en la parte superior izquierda hay un botón que permite retroceder, en la parte superior derecha se mira el puntaje obtenido contando las respuestas respondidas de forma correcta; cuando se presiona el botón de retroceder se abre un menú en cual se debe confirmar si se desea salir de la aplicación como se observa en la Figura 47, en que caso que se presione el botón si, se regresa a la escena de menú principal, en caso se presione el botón de no, se cerrará el menú y volverá a la actividad; en la parte superior central se ve una barra que rastrea el progreso de la actividad.

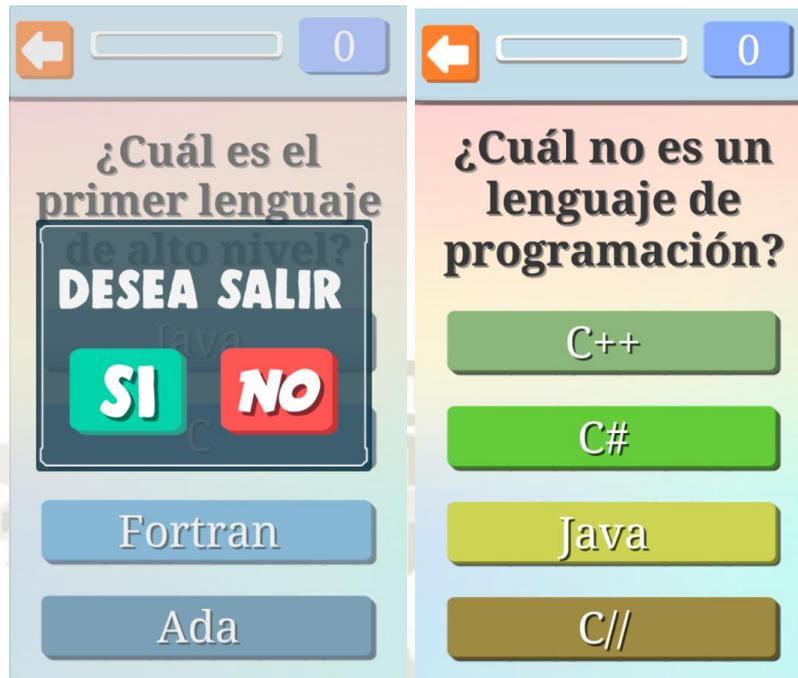


Figura 47. Pantalla para salir de la actividad y Actividad de pregunta (Fuente: Elaboración Propia)

Luego de unos segundos aparece en la pantalla la pregunta que se debe resolver y una serie de opciones para responder como en la Figura 47, luego se debe seleccionar una de las opciones, en caso de responder de forma correcta se muestra un mensaje de correcto y se explica por qué es correcta la opción escogida aumentando el puntaje de la parte superior derecha e incrementando la barra de progreso; en caso que se escoja la respuesta incorrecta se muestra una pantalla de error y se explica la razón porque es incorrecto y se muestra un avance en la barra de progreso, pero no se incrementa ningún punto como se muestra en la Figura 48.

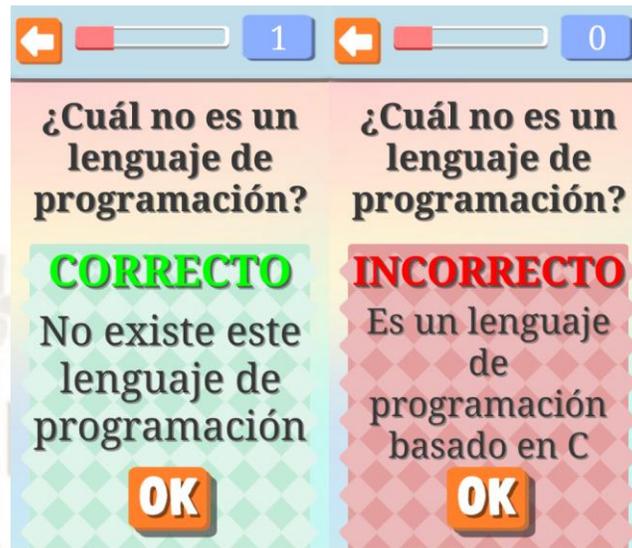


Figura 48. Respuesta correcta e incorrecta de la actividad pregunta (Fuente: Elaboración Propia)

Si se cumple todos los objetivos de un logro se muestra un mensaje especificando el logro objetivo, como se muestra en la Figura 49, el cual aparece con un sonido que referencia a que se ha desbloqueado un logro y después se muestra el nombre del logro, después de un momento el logro desaparece.

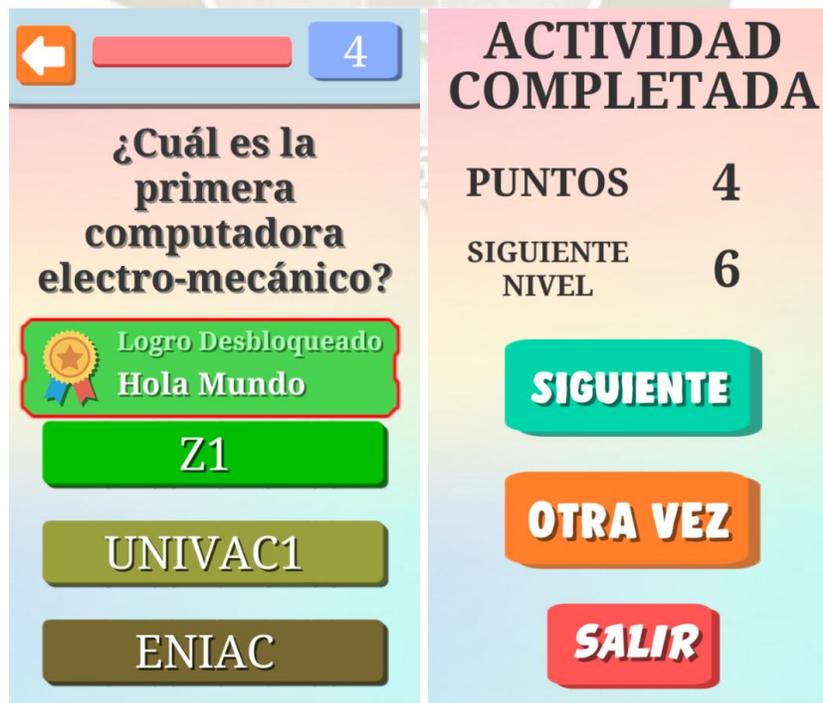


Figura 49. Mensaje de logro desbloqueado y Salir de la actividad (Fuente: Elaboración Propia)

Cuando se termina la actividad se muestra una pantalla que permite ir a la siguiente actividad, volver a realizar la actividad o salir al menú principal como se muestra en la Figura 50.

Resolver verdadero o falso

Después de haber seleccionado la actividad de verdadero o falso, se muestra una pantalla inicial en la cual se indica la acción a realizar como se observa en la Figura 50, en la parte superior izquierda hay un botón que permite retroceder, en la parte superior derecha se mira el puntaje obtenido contando las respuestas respondidas de forma correcta, en la parte superior central se ve una barra que rastrea el progreso de la actividad.

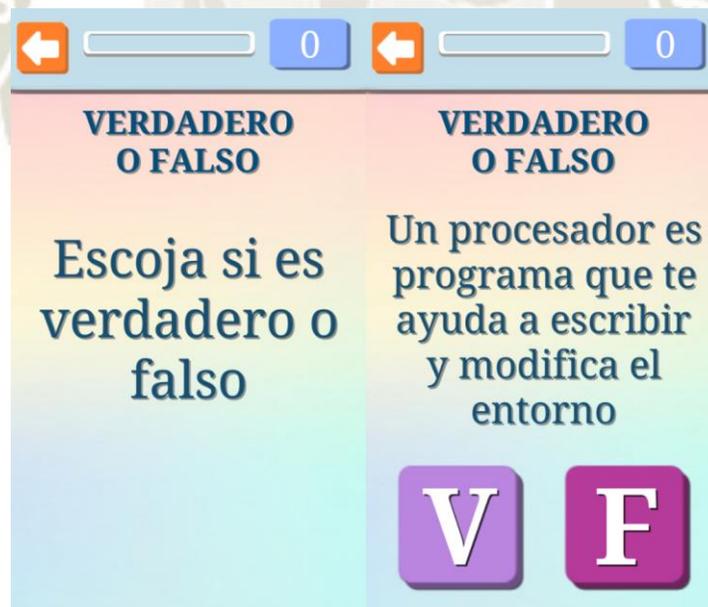


Figura 50. Inicio de la actividad de verdadero o falso (Fuente: Elaboración Propia)

Se debe seleccionar una opción verdadero o falso, en caso de responder de forma correcta se cambia el color del botón a verde y se muestra el puntaje en la parte superior derecha e incrementando la barra de progreso; en caso que se escoja la respuesta

incorrecta se cambia el color de botón a rojo y se muestra un avance en la barra de progreso, pero no se incrementa ningún punto como se muestra en la Figura 51.



Figura 51. Respuesta correcta e incorrecta de la actividad verdadero o falso (Fuente: Elaboración Propia)

Cuando se termina la actividad se muestra una pantalla que permite ir a la siguiente actividad, volver a realizar la actividad o salir al menú principal como se muestra en la figura 49.

Resolver conectar conceptos

Después de haber seleccionado la actividad de conectar conceptos, se muestra una pantalla inicial en la cual se indica la acción a realizar como se observa en la Figura 52, en la parte superior izquierda hay un botón que permite retroceder, en la parte superior derecha se mira el puntaje obtenido contando las respuestas respondidas de forma correcta, en la parte superior central se ve una barra que rastrea el progreso de la actividad.

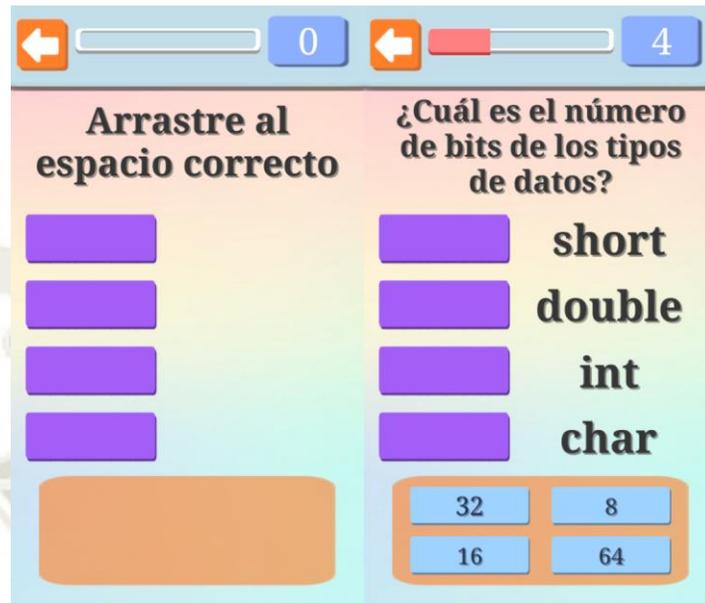


Figura 52. Inicio de la actividad de conectar conceptos (Fuente: Elaboración Propia)

Se debe arrastrar los datos como se observa en la Figura 53, cuando se está arrastrando un dato aparece el nombre del dato en la posición del dedo, para que se pueda identificar a dónde se arrastra.



Figura 53. Arrastrar datos en la actividad conectar conceptos (Fuente: Elaboración Propia)

En caso de arrastrar el concepto de forma correcta se cambia el color del botón a verde, se mueve el dato en la nueva posición y se muestra el puntaje en la parte superior derecha; en caso que se arrastre a otra posición se cambia el color de botón a rojo, se regresa el concepto a su posición original, pero no se incrementa ningún punto como se muestra en la figura 54.

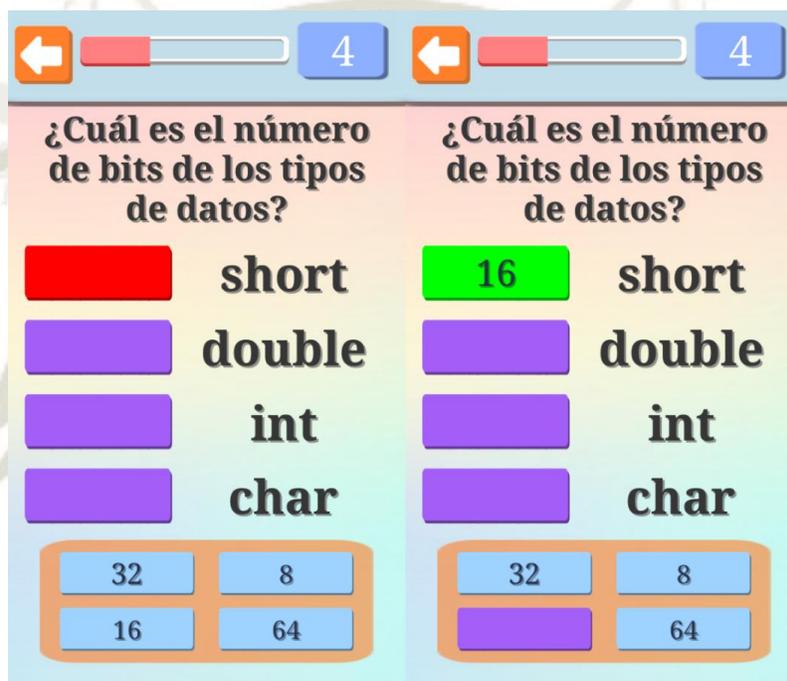


Figura 54. Respuesta correcta e incorrecta de la actividad conectar conceptos (Fuente: Elaboración Propia)

Finalmente, cuando se termine de completar de conectar todos los conceptos, se pintan todos los conceptos de verde, se incrementa la barra de progreso y se muestra un mensaje de correcto para comenzar la siguiente pregunta para conectar conceptos como se muestra en la Figura 55.



Figura 55. Completar ejercicio de la actividad conectar conceptos (Fuente: Elaboración Propia)

2.7 Validación de la aplicación

2.7.1 Caso de Estudio

Se realizó un caso de estudio con alumnos de la Universidad Católica de Santa María, en dicho caso de estudio fueron evaluados 22 estudiantes pertenecientes a la carrera de Ingeniería de Sistemas, para realizar la evaluación primero se realizó un pre-test en cual se dieron 9 preguntas referentes a programación en Java, luego los alumnos procedieron a usar la aplicación, luego de terminar de usar la aplicación a los alumnos se les pidió volver a realizar una evaluación pos-test para evaluar el conocimiento adquirido finalmente se les realizó una encuesta a los alumnos con preguntas sobre la usabilidad de la aplicación donde se evaluaron las siguientes heurísticas:

1. Visibilidad en la aplicación.
2. Coincidencia con el mundo real.

3. Control y libertad.
4. Consistencia en la aplicación.
5. Prevención de errores.
6. Intuitivo.
7. Flexibilidad.
8. Estética Minimalista.

Luego de realizar el caso de estudio se pudo observar en la Figura 56 y la Figura 57 un aumento de las respuestas correctas por parte de los estudiantes luego de haber utilizado la aplicación.

Finalmente en la Figura 58 se determinó que más del 75% de los estudiantes ha encontrado fácil el uso de la aplicación.

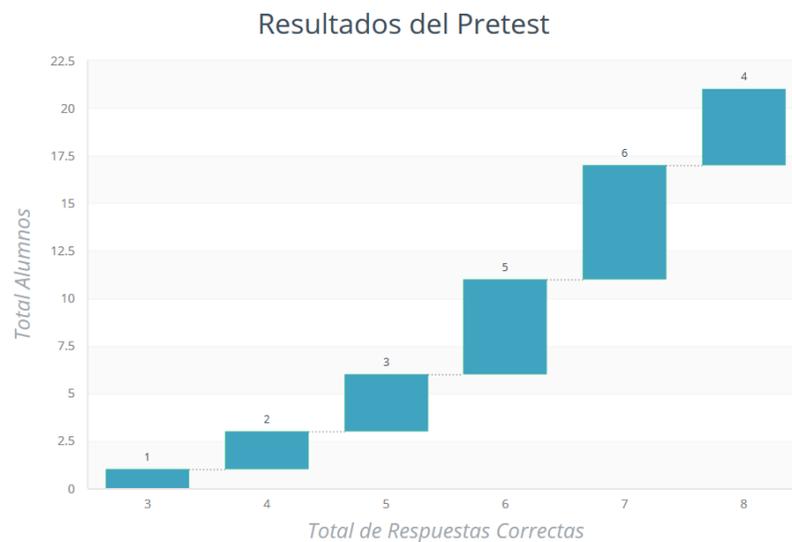


Figura 56. Resultados del Pre-test (Fuente: Elaboración Propia)

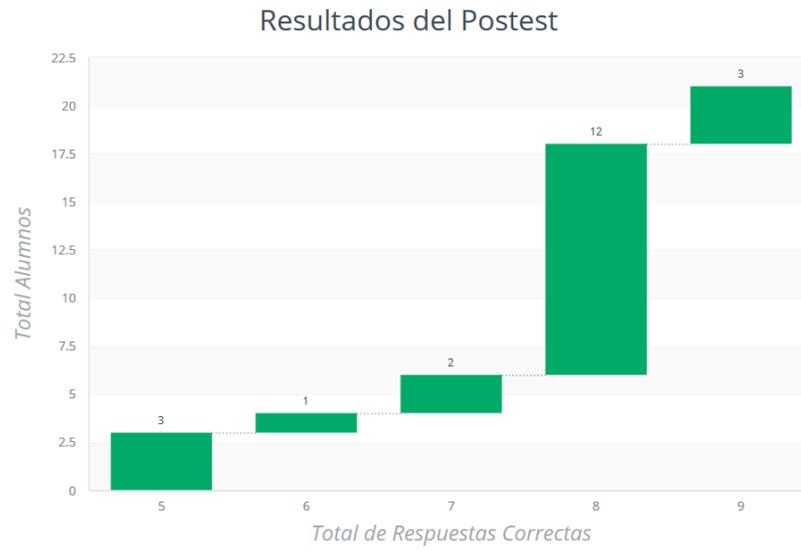


Figura 57. Resultados del Pos-test (Fuente: Elaboración Propia)

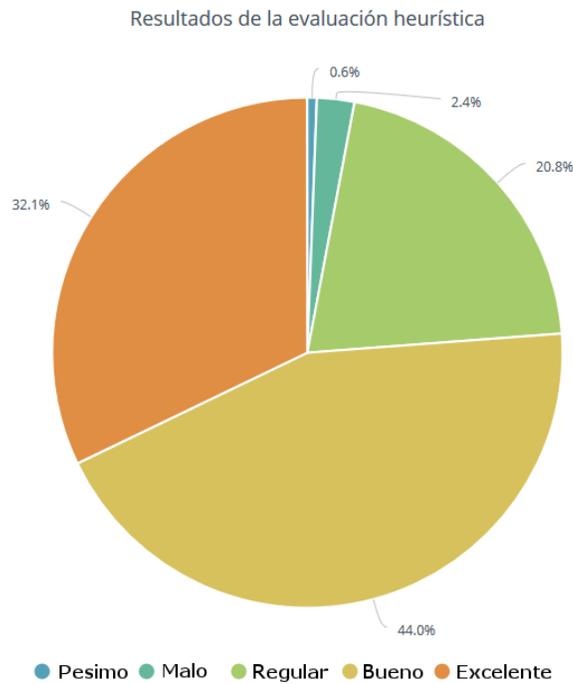


Figura 58. Resultados de la evaluación heurística (Fuente: Elaboración Propia)

3. CONCLUSIONES

Según los objetivos del proyecto se concluyó lo siguiente:

- Primero la metodología de programación escogida XP permitió realizar una correcta especificación de requisitos, con la cual basándonos en las historias de usuarios y las entrevistas se pudieron determinar los requerimientos funcionales y no funcionales de la aplicación con facilidad, facilitando el proceso de programación en los tiempos establecidos.
- Segundo las técnicas de gamification escogidas permitieron mejorar la usabilidad de la aplicación y la motivación de los estudiantes según se vio en los resultados obtenidos.
- Tercero el realizar las especificaciones de los requisitos de la aplicación se pudo establecer plazos y tiempos correctos para el desarrollo de la aplicación.
- Cuarto el uso de un diseño intuitivo y minimalista facilitó el uso de la aplicación por parte de los nuevos usuarios.
- Quinto se implementó la aplicación mediante la herramienta Unity, el cual facilitó el proceso de construcción del software educativo, con lo que se llegó a culminar la programación de la aplicación en los tiempos establecidos.
- Sexto el desarrollo de los componentes de la aplicación fue una buena forma de trabajo, debido a que permitió separar las diferentes acciones

permitiendo tener componentes reutilizables en otros proyectos con lo cual es posible reducir el tiempo de programación, adicionalmente la integración de estos componentes fue hecha de manera rápida gracias al uso de patrones de programación como singleton.

- Séptimo debido al uso de software libre este proyecto es económicamente y técnicamente viable con lo cual se pudo determinar que es posible realizar un software educativo sin la necesidad de licencias.

Finalmente luego de realizar la prueba de la aplicación con los usuarios se pudieron las siguientes conclusiones:

- Los usuarios que usaron la aplicación entendieron de manera rápida el uso de la aplicación, y brindaron sus opiniones para mejorar la aplicación con las pruebas realizadas.
- El uso de gamification permitió motivar a los estudiantes, ya que el uso de niveles y porcentaje de avance ayuda a los estudiantes a notar su nivel de avance y les dio libertad de escoger la actividad que ellos deseaban realizar.

4. RECOMENDACIONES Y TRABAJOS FUTUROS

4.1. Recomendaciones

Se recomienda utilizar la presente investigación como base para elaborar una mayor cantidad de actividades, este proyecto cuenta con tres diferentes actividades, con el fin de brindar más opciones a los estudiantes de aprender conceptos de programación.

Además de la adición de más actividades, se recomienda incluir temas, con lo cual el estudiante pueda seleccionar actividades según un tema específico y de esta manera poder estudiar una mayor cantidad de conceptos sobre la programación.

Se recomienda también el uso de video dentro de la aplicación, con lo cual se permitiría dar al alumno un tutorial sobre los diferentes temas, el cual guíe al alumno y sirva como complemento a su proceso educativo.

4.2. Trabajos Futuros

Como trabajo futuro se puede incluir actividades grupales, las cuales requieren el trabajo en conjunto de dos alumnos para ser completadas, para ello sería bueno considerar las conexiones entre dispositivos móviles, ya sea mediante bluetooth o equipos conectados en una misma red Wi-Fi.

Este trabajo tiene un gran potencial a futuro, ya que puede servir como base para la enseñanza de otros cursos, donde se puede usar el diseño de la aplicación y las actividades para enseñar otras materias como matemática, lenguaje, ciencias, entre otras.

En una escala mayor este trabajo puede expandirse en una plataforma que permita seleccionar una materia en específico y luego un tema los cuales serían creados por el profesor según los temas que están realizando en clase.



5. REFERENCIAS BIBLIOGRÁFICAS

- Akilli, G. K. (2011). Games and simulations: A new approach in education. *Gaming and Simulations: Concepts, Methodologies, Tools and Applications*, 150–167.
<https://doi.org/10.4018/978-1-59904-808-6.ch078>
- Al-imamy, S., Alizadeh, J., & Nour, M. a. (2006). On the Development of a Programming Teaching Tool : The Effect of Teaching by Templates on the Learning Process. *Journal of Information Technology Education*, 5, 271–283. Retrieved from <http://www.editlib.org.ezproxy.psz.utm.my/p/111545/>
- Azmi, S., Iahad, N. A., & Ahmad, N. (2015). Gamification in online collaborative learning for programming courses: A literature review. *Researchgate*, 10(23), 18087–18094. Retrieved from https://www.researchgate.net/profile/Shahdatunnaim_Azmi/publication/290460188_Gamification_in_online_collaborative_learning_for_programming_courses_A_literature_review/links/56d4e20e08ae2cd682b94244/Gamification-in-online-collaborative-learning-for-progra
- Ballu, A., Yan, X., Blanchard, A., Clet, T., Mouton, S., & Niandou, H. (2016). Virtual Metrology Laboratory for e-Learning. *Procedia CIRP*, 43, 148–153.
<https://doi.org/10.1016/j.procir.2016.02.110>
- Buisman, A. L. D., & van Eekelen, M. C. J. D. (2014). Gamification in educational software development. In *Proceedings of the Computer Science Education Research Conference* (pp. 9–20).
- Butt, P. (2016). Students' perceptions of game-based learning using CodinGame.

- Caillois, R. (2001). *Man, Play and Games*. 1958. *Trans. Meyer Barash. Urbana and Chicago: University of Illinois Press.*
- Craighead, J., Burke, J., & Murphy, R. (2008). Using the Unity Game Engine to Develop SARGE : A Case Study. In *Itsec* (Vol. 4552, p. 366). Retrieved from <http://www.robot.uji.es/research/events/iros08/contributions/craighead.pdf>
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness. *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11*, 9. <https://doi.org/10.1145/2181037.2181040>
- Díaz Cruzado, J., & Troyano Rodríguez, Y. (2013). El potencial de la gamificación en el ámbito educativo. *III Jornadas de Innovación Docente. Innovación Educativa*, 9. Retrieved from https://fcee.us.es/sites/default/files/docencia/EL POTENCIAL DE LA GAMIFICACIÓN APLICADO AL ÁMBITO EDUCATIVO_0.pdf <https://idus.us.es/xmlui/handle/11441/59067>
- Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: a systematic mapping study Badges. *Educational Technology & Society*. Retrieved from https://scholar.google.com/citations?view_op=view_citation&continue=/scholar%3Fhl%3Den%26start%3D20%26as_sdt%3D0,33%26scilib%3D1&citilm=1&citation_for_view=EIOeQSYAAAAJ:EUQCXRtRnyEC&hl=en&oi=p%5Cnhttps://scholar.google.com/citations?view_op=view_citation&
- Fredes, C. A., Hernández, J. P., & Díaz, D. A. (2012). Potencial y Problemas de la

- Simulación en Ambientes Virtuales para el Aprendizaje. *Formación Universitaria*, 5(1), 45–56. <https://doi.org/10.4067/S0718-50062012000100006>
- Gallego-Durán, F. J., Villagrà-Arnedo, C., Llorens-Largo, F., & Molina-Carmona, R. (2017). PLMan: A game-based learning activity for teaching logic thinking and programming. *International Journal of Engineering Education*, 33(2), 807–815.
- Iqira Becerra, D. A. (2018). Implementacion del laboratorio virtual inmersivo aplicado a la enseñanza de física usando técnicas de Gamification.
- Joskowicz, I. J., & Mingus, C. (2008). Reglas y Prácticas en eXtreme Programming. *Universidad de Vigo*, 1–22.
- Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming. *Universidad de Vigo. España*, 1–22. Retrieved from <http://iie.fing.edu.uy/~josej/docs/XP - Jose Joskowicz.pdf>
- Laster, J. (2010). At Indiana U., a class on game design has students playing to win. *The Chronicle of Higher Education*.
- Minnaard, C., & Minnaard, V. (2014). Evaluación por competencias en Entornos Virtuales de Aprendizaje. *WICC 2014 XVI Workshop de Investigadores En Ciencias de La Computación*, 903–906. Retrieved from http://traffilight.bitdefender.com/info?url=http%3A//sedici.unlp.edu.ar/handle/10915/43712&language=en_US
- Orji, R., Mandryk, R. L., Vassileva, J., & Gerling, K. M. (2013). Tailoring persuasive health games to gamer type. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 2467.

<https://doi.org/10.1145/2470654.2481341>

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137–172.

<https://doi.org/10.1076/csed.13.2.137.14200>

Satratzemi, M., Xinogalos, S., & Dagdilelis, V. (2003). An environment for teaching object-oriented programming: ObjectKarel. *Proceedings - 3rd IEEE International Conference on Advanced Learning Technologies, ICALT 2003*, 342–343.

<https://doi.org/10.1109/ICALT.2003.1215114>

Sims, Z., & Bubinski, C. (2011). Codecademy. URL= *Http://Www. Codecademy. Com.*

Standard, I. (1993). IEEE Recommended Practice for Software Requirements Specifications, 830, 43.

Unity Technologies. (n.d.). Unity. Retrieved January 19, 2018, from <https://unity3d.com/es>

Anexo A: Formato de pre-test y post-test

¿Cuál de las siguientes es una palabra reservada en Java?

- a. NULL
- b. new
- c. instanceof
- d. wend

¿Cuál de las siguientes instrucciones compila sin errores?

- a. `char c="a";`
- b. `byte b=257;`
- c. `boolean b=null;`
- d. `int i=10;`

¿Cuál de los siguientes no es un identificador válido en Java?

- a. #variable
- b. \$variable
- c. _variable
- d. vari_able

Respecto a los bucles, indique cuál de las siguientes afirmaciones es falsa:

- a. Un bucle for-each puede repetirse 0 o más veces.
- b. Un bucle for-each puede aplicarse sobre cualquier clase que implemente la interfaz Iterable.
- c. El cuerpo de un bucle while siempre se ejecuta, como mínimo, una vez.
- d. Un bucle for-each puede aplicarse sobre arreglos (arrays).

¿Qué elementos cree que definen a un objeto?

- a. Sus cardinalidad y su tipo
- b. Sus atributos y sus métodos
- c. La forma en que establece comunicación e intercambia mensajes
- d. Su interfaz y los eventos asociados

¿Qué significa instanciar una clase?

- a. Duplicar una clase
- b. Eliminar una clase
- c. Crear un objeto a partir de la clase
- d. Conectar dos clases entre sí

¿Qué es una 'clase' en la programación orientada a objetos?

- a. Es un concepto similar al de 'array'
- b. Es un tipo particular de variable
- c. Es un modelo o plantilla a partir de la cual creamos objetos
- d. Es una categoría de datos ordenada secuencialmente

¿Qué significa sobrecargar un método?

- a. Editarlo para modificar su comportamiento
- b. Cambiarle el nombre dejándolo con la misma funcionalidad
- c. Crear un método con el mismo nombre pero diferentes argumentos
- d. Añadirle funcionalidades a un método

¿Cual es el valor de X ? $\text{int } x = 5 * 4 \% 3$; ¿Cuál es el valor de la operación?

- a. Error de compilación en la línea.
- b. 2
- c. 3
- d. 6

Las presentes preguntas fueron obtenidas de diversos exámenes.

REFERENCIAS BIBLIOGRÁFICAS

Cartagena99.com. (2018). [online] Available at:
[http://www.cartagena99.com/recursos/alumnos/ejercicios/Problemas.Resueltos.POO.SIN.RESPUESTAS.v1.3%20\(1\).pdf](http://www.cartagena99.com/recursos/alumnos/ejercicios/Problemas.Resueltos.POO.SIN.RESPUESTAS.v1.3%20(1).pdf) [Accessed 5 Oct. 2018].

Jchq.net. (2018). *Mock Java Certification Exam*. [online] Available at:
<http://www.jchq.net/mockexams/exam1.htm> [Accessed 5 Oct. 2018].

Zentenö (2018). Test. [online] Es.slideshare.net. Available at:
<https://es.slideshare.net/ratzenteno/test-60592685> [Accessed 5 Oct. 2018].

Anexo B: Formato de encuesta de heurística de usabilidad

Correo Electrónico: _____

Sexo _____ Edad _____

No	Heurística	Calificación Totalmente de desacuerdo/ En desacuerdo/ Neutral/ De acuerdo/ Totalmente de acuerdo
1	Visibilidad en la aplicación	
2	Coincidencia con el mundo real	
3	Control y Libertad	
4	Consistencia en la aplicación	
5	Prevención de errores	
6	Intuitivo	
7	Flexibilidad	
8	Estética Minimalista	

Anexo C: Guía para la encuesta de usabilidad

Visibilidad en la aplicación

- Se muestra toda la información necesaria en la pantalla.
- El sistema responde correctamente a las acciones, no existen acciones sorpresas o errores.

Coincidencia con el mundo real

- El lenguaje, términos y conceptos son similares al del mundo cotidiano
- Los diferentes iconos utilizados son intuitivos.
- La información está organizada en un orden natural y lógico

Control y Libertad

- Controlan correctamente la aplicación
- Pueden salir de la aplicación en cualquier momento
- Es fácil desplazarse entre escenas

Consistencia en la aplicación

- Se usan los mismos conceptos palabras y símbolos en la aplicación
- Existen estándares comunes en la aplicación

Prevención de errores

- El sistema evita que cometan errores
- Cuando comenten un error se muestra un mensaje con información del error

Intuitivo

- Los elementos con los que se interactúan son visibles
- No necesitas recordar información para navegar en la aplicación
- Las interfaces son simples y se entienden cómo usarlas

Flexibilidad

- Existen accesos directos que permiten navegar de manera más rápida
- El sistema se puede configurar

Estética Minimalista

- La aplicación no contiene información irrelevante o que sea innecesaria

Anexo D: Plan de Tesis

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Caracterización del Problema

La educación es un concepto que va cambiando y adaptándose, donde en un principio pasa de ser un modelo basado en la transmisión y observación, a un modelo activo y participativo, con lo cual se puede establecer nuevas estrategias en el aprendizaje.

Pero para lograr dicha participación de los estudiantes es necesaria la creación de una metodología que permita crear experiencias educativas concretas y vivenciales.

Por ejemplo las actividades lúdicas permiten que los alumnos generen un potencial heurístico, el cual permite mejorar las capacidades tanto de análisis como de síntesis, debido a que dan una mayor libertad al estudiante.

En el contexto educativo en las carreras científicas existe una idea de que la realización de ejercicios y trabajos permiten mejorar el aprendizaje de los estudiantes, ya que facilita la relación de lo aprendido en forma teórica con la parte práctica.

Un problema que se ha identificado en la educación actual son las limitaciones de materiales y la falta de recursos humanos, que hacen que sea difícil aprender sólo mediante ejercicios prácticos en el aula (Ballu et al., 2016).

Debido a esto, los software educativos son las herramientas educativas de la nueva sociedad del aprendizaje (Fredes et al., 2012), Por lo cual se han adoptado estrategias a

nivel institucional y de aula para que los alumnos usen adecuadamente software educativos.

Lamentablemente las lecciones realizadas en los salones no siempre son apreciadas por los alumnos, en la realización de clases actual tanto el aprendizaje como la motivación del estudiante y del profesor se puede ver afectado por estímulos externos o por una falta de atención e interés (Ballu et al., 2016).

Esto explicaría las razones porque se usa la gamification en la educación, ya que ayuda a motivar a los estudiantes, a aprender mediante elementos de juegos en entornos educativos con la meta de maximizar el entreteniendo y el compromiso del estudiante e inspirarlos a continuar aprendiendo (Dicheva et al., 2015).

El uso de la tecnología en la educación se ha vuelto una necesidad, ya que permite brindar un apoyo al estudiante en su proceso educativo, el uso de la gamification permite mejorar el proceso de diferentes tareas, al convertirlas en acciones más lúdicas, motivando a los usuarios al realizar su trabajo.

El presente plan de tesis busca el desarrollo de un software con la intención de enseñar conceptos básicos de programación mediante técnicas como la *gamification*, con la intención de mejorar el proceso educativo de cursos enfocados en la informática, usando técnicas enfocadas en mejorar la motivación de los estudiantes a aprender.

La gamification busca usar conceptos de videojuegos en ambientes no lúdicos, permitiendo mejorar la motivación de los usuarios, es por esta razón que puede ser

usado en diferentes campos como en la educación donde sirve como un complemento a los procesos realizados.

Se realizara un trabajo de diseño y creación del software educativo, con el fin de poder determinar la importancia del uso de tecnología en la educación como complemento al alumno en su proceso educativo.

1.2. Línea y Sub-línea de Investigación a la que corresponde el Problema

Línea: Sistemas de Información y Bases de Datos.

Sub-línea: E-learning.

1.3. Palabras Clave

Tecnología Educativas, Gamification, TIC, Aprendizaje, Unity.

2. OBJETIVOS DEL PROYECTO

2.1. General

Diseñar un software educativo para la enseñanza de conceptos básicos de programación usando gamification.

2.2. Específicos

Establecer las actividades educativas a realizar.

Identificar y seleccionar un conjunto de técnicas de gamification.

Desarrollar la mecánica, que permite realizar las actividades educativas.

Integrar, en el software educativo la parte visual y sonora.

3. FUNDAMENTOS TEÓRICOS

3.1. Estado del arte

Se ha realizado la verificación de la relación de estudios de investigación realizadas sobre el presente tema en la biblioteca de la Universidad Nacional de San Agustín, en la Universidad Alas Peruanas, Universidad Católica de Santa María, en las cuales no se encontraron trabajos a nivel de pre-grado, ni de nivel de post-grado sobre la materia. Se revisó las investigaciones relacionadas al uso de la gamification en la educación, donde se buscaba la dinámica del juego usada según las circunstancias.

Primero Jill Laster (Laster, 2010) decidió cambiar el formato del curso volviéndolo un juego, donde el tiempo de clases se usaba para completar misiones, pelear contra enemigos (tomar exámenes o cuestionarios) y crear (escribir documentos), dividió la clase en diferentes zonas, cambio el formato de notas por puntos, ya que permite sentir a los estudiantes que están ganando puntos por hacer las cosas bien en lugar de perderlas por respuestas incorrectas.

Después se puede ver Re-Mission donde tiene como misión contralar un nano robot el cual se mueve en el cuerpo de un paciente destruyendo las células cancerígenas mediante quimioterapia y radiografía, con el fin de mostrar el beneficio de dichas técnicas, los resultados obtenidos permitieron generar un mayor conocimiento y entendimiento del tratamiento del cáncer en los pacientes en comparación de los que no habían jugado (Orji et al., 2013).

Otro ejemplo es Help que es una plataforma para diferentes estudiantes, donde los alumnos pueden buscar respuestas a preguntas, mediante la ayuda de otros estudiantes, donde si un alumno conoce la respuesta gana un recompensa (Díaz Cruzado & Troyano Rodríguez, 2013).

Los jugadores van comenzando con un cierto puntaje el cual se va incrementando o reduciendo dependiendo de su colaboración el proyecto, ese puntaje puede ser usado para canjear objetos.

Otras empresas crearon aplicaciones similares como Google con Google Answer o Yahoo con Yahoo Answers (Orji et al., 2013).

Otro programa usado para la enseñanza de programación es CodeAcademy (Sims & Bubinski, 2011), donde se enseñan cursos de programación, los cuales son organizados e interconectados, los cuales consiste en una secuencia de ejercicios con un texto educativo que explica el tema, instrucciones indicando la tarea a realizar y un ejercicio interactivo real que se debe completar, finalmente cada estudiante gana puntos por completar cada ejercicio y se guarda cada lección aprendida como logros.

Mientras que la mayoría de plataformas educativas enfocadas en la gamification usan logros, puntos o medallas, Codingames usa juegos para ayudar el proceso educativo, su temario incluye más de 20 diferentes tipos de lenguaje de programación, donde cada alumno puede ver lo que está codificando en la pantalla (Butt, 2016).

Software Educativo. El software educativo es la herramienta tecnológica para la nueva sociedad del aprendizaje (Fredes et al., 2012; Minnaard & Minnaard, 2014) por lo que se han desarrollado estrategias en diferentes niveles, desde el nivel institucional hasta el salón de clases, con el objetivo de buscar un uso apropiado de software educativo.

La educación especialmente ha recibido mucha atención por parte de investigadores y compañías como un campo para aplicar la gamification (Buisman & van Eekelen, 2014). En la educación se ha utilizado la gamification para motivar a los estudiantes a aprender mediante el uso de elementos del diseño de juegos, donde se busca aumentar la diversión, participación y motivación del estudiante logrando que continúe aprendiendo (Azmi et al., 2015).

Gamification. El término gamification es la aplicación de elementos de juego y principios de diseños de juegos en otros contextos no relacionados con juegos.

En resumen, la gamification utiliza elementos de diseño de juegos para mejorar la motivación, la participación y la productividad del usuario (Deterding et al., 2011).

Una definición general de gamification es el uso de elementos de diseño de juegos en contextos distintos de los juegos (Deterding et al., 2011).

La gamification se relaciona con los juegos y no con jugar, donde el jugar es una forma más libre y expresiva y el juego es un conjunto de normas para conseguir una meta (Caillois, 2001), debido a estas diferencias que existe al describir un juego se ha llegado al uso de una terminología para sistematizar dichas definiciones:

Los principales elementos de diseño de juegos usados en la gamification son (Akilli, 2011):

- Mecánicas de Progresión.
- Narrativa.
- Control del Jugador.
- Retroalimentación.
- Resolución de Problemas en Forma Colaborativa.
- Oportunidades de Dominio.
- Progresión por Niveles.
- Relaciones Sociales.
- Retos.
- Música.

Gamification en la Educación. En la educación se ha utilizado la gamification para motivar a los estudiantes a aprender mediante el uso de elementos del diseño de juegos, donde se busca aumentar la diversión, participación y motivación del estudiante logrando que continúe aprendiendo (Azmi et al., 2015).

Una diferencia en el uso de gamification con el aprendizaje basado en juegos, es que no involucra al estudiante en el diseño y creación del juego, ya que en la gamification se busca que el aprendizaje se consiga en un contexto no de juegos, sino en el aula por tal razón se busca realizar la actividad de aprendizaje en un salón de clases (Akilli, 2011).

Las ventajas del uso de gamification en la educación son (Akilli, 2011):

- Libertad para prueba y error sin repercusiones negativas.
- Posibilidad de aumentar la diversión en el salón.
- Educación diferenciada según las habilidades del estudiante.
- Visualización del aprendizaje.
- Brindar un conjunto de tareas y subtareas.
- Motivar a los estudiantes a aprender.
- Libertad al estudiante para realizar su propio aprendizaje.

De los cuales no todos son necesarios usarlos, por tal razón se deben escoger en base a la actividad que se desea realizar.

4. PRESENTACIÓN DEL PROYECTO

4.1. Justificación

Debido a la importancia de utilizar nuevas tecnologías para la educación con el fin de mejorar el proceso educativo, el uso de software educativo se ha vuelto una necesidad para los estudiantes en la actualidad.

Pero dicho software no solo debe ser usado con el fin de educar a los estudiantes, sino debe buscar motivar a los estudiantes a seguir aprendiendo.

Es por esta razón que el uso de técnicas de gamification en el software educativo no solo permite crear una tecnología que sea útil para los estudiantes, sino que adicionalmente permite motivar a los estudiantes en el proceso educativo.

El aporte de dicha investigación al ser un tema novedoso en el Perú permite tener un gran impacto al aportar una nueva alternativa para la educación.

El beneficio de la creación de un prototipo y de un diseño correcto permite que dicho prototipo pueda ser incorporado a diferentes áreas beneficiando a los profesores brindándoles un diseño base para el desarrollo en otras áreas.

Un tema abordado es la educación actual, la cual puede ser mejorada cambiando la perspectiva del alumno sobre los cursos que estudian, ya que volviendo dichas actividades educativas a actividades educativas lúdicas, permiten que el alumno adquiera una mayor motivación para realizar trabajos y aprender nuevas cosas.

4.2 Resumen del Proyecto

4.2.1 Descripción del Proyecto a medio y largo plazo. Se plantea el desarrollo de un software educativo enfocado en la enseñanza de conceptos de programación, donde en una primera etapa se plantea haber desarrollado la mecánica básica de las diferentes actividades educativas que realizara el estudiante, en una segunda etapa se plantea haber adicionado los elementos de gamification los cuales ayudan a la motivación del estudiante, en una etapa final se busca unir tanto la parte visual como sonora en combinación con la mecánica básica.

4.2.2 Usuarios del Proyecto. Se plantea como usuarios a los alumnos de primer año de cursos de programación, con lo cual se les brinde un software educativo que complemente su proceso educativo.

4.2.3 Beneficios. Debido al uso de gamification brinda una ventaja en comparación con otros programas educativos, adicionalmente al ser desarrollado para dispositivos

móviles brinda una mayor accesibilidad al programa permitiendo una mayor libertad en cuanto a su uso.

4.2.4 Localización. Se realizara el desarrollo del software educativo ciudad de Arequipa, la disposición del programa desarrollado se ubicara en la tienda de Android con lo cual permite una fácil acceso para la instalación de la aplicación.

4.2.5 Análisis del futuro del Proyecto. Complemento al proceso educativo realizado, en la enseñanza de conceptos básicos de programación, una de las mayores ventajas del uso de dicho software educativo es que gracias a la información que se puede obtener de los diferentes estudiantes, es posible analizar a los estudiantes y poder apoyarlos en las áreas de poco dominio.

4.2.6 Riesgos que debemos afrontar:

Económicos. *Se deberá contar con dispositivos móviles para poder realizar las pruebas.*

Competencia. *Debido a que cada día se suben diferentes aplicaciones en Google Play Store, siempre existirán diferentes aplicaciones que busquen realizar la misma tarea.*

Tecnológicos. *Debido a que al desarrollar una aplicación en Android no existe especificaciones iguales en todos los dispositivos siempre será necesario crear una aplicación la cual se adapte a la mayoría de los equipos.*

No Tecnológicos. *Pueden existir dificultades en la búsqueda de recursos como imágenes y sonidos, debido a problemas con licencias*

5. PLAN DE IMPLANTACIÓN DEL PROYECTO

5.1 Definición del Proyecto

5.1.1 Aspectos Técnicos. Para el desarrollo del presente proyecto se usará el programa Unity que es un programa enfocado en el desarrollo de videojuegos y simulaciones, se realizarán pruebas sobre los aplicativos en tres diferentes equipos móviles, Samsung s7, Moto G5 y Asus zenfone 3 deluxe.

Se hará el desarrollo de la aplicación para Android desde la versión Android – 4.4 KitKat a adelante.

5.1.2 Aspectos Económicos. Para la implantación del proyecto será necesario la compra de una licencia de programador en Android, para poder subir la aplicación en Google Play Store, será necesario la compra de recursos tanto de imágenes como de sonidos.

5.1.3 Aspectos Comerciales. El uso de gamification brinda un atractivo para los usuarios ya que busca que el proceso educativo se vuelva más entretenido para los usuarios sin perder el lado educativo.

5.1.4 Recursos del Proyecto. El desarrollo del aplicativo se realizará por una persona, se usará una computadora Asus para la programación en un sistema operativo Windows, las pruebas se realizarán en tres diferentes dispositivos Android: Samsung s7, Moto G5 y Asus zenfone 3, se contará con una licencia de desarrollador en Android y en Unity.

6. METODOLOGÍA A EMPLEAR

Se ha determinado el uso de una metodología ágil para el desarrollo del software educativo, ya que permite acelerar el proceso de desarrollo del aplicativo mediante iteraciones que se desarrollan a lo largo todo el ciclo de vida del software, existe

distintas metodologías ágiles, para el desarrollo del software educativo se ha optado utilizar la metodología XP (Extreme Programming).

Una de las principales razones que se optó por el uso de una metodología ágil es la reducción de riesgos en el desarrollo de software debido a las iteraciones que se realizan en cortos lapsos de tiempo, una iteración puede ser comprendida como una nueva versión del programa hechas en un determinado lapso de tiempo, donde cada iteración dura entre una a cuatro semanas.

La metodología propuesta en XP está diseñada para entregar el software que los usuarios necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo (I. J. Joskowicz & Mingus, 2008).

Adicionalmente Extreme Programming es una metodología que se basa en cuatro fases importantes: exploración, planificación, iteraciones, y puesta en producción (I. J. Joskowicz & Mingus, 2008).

6.1 Fase de exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle

en cada iteración, esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado (Joskowicz, 2008).

6.2 Fase de planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, o “Release Plan”, como se detallará en la sección “Reglas y Practicas” (I. J. Joskowicz & Mingus, 2008).

6.3 Fase de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo.

Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance (I. J. Joskowicz & Mingus, 2008).

6.4 Fase de puesta en producción

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa.

En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste (“fine tuning”) (I. J. Joskowicz & Mingus, 2008).

Luego de revisar de forma más detallada la metodología XP, se pudo determinar que resultaría ser la metodología ideal, debido a que como el proyecto se va a desarrollar en un grupo pequeño, el cual necesita diferentes iteraciones para su desarrollo debido a que se realizarán diferentes reuniones con profesores para determinar las tareas a realizar y poder hacer una comprobación del nivel educativo y finalmente en la última fase se hará una revisión final del software y la evaluación que se realizara a los alumnos a nivel de software.

7. PLAN DE TRABAJO

Se muestra en la presente tabla 1 las actividades a realizar en un periodo de meses para cumplir con el trabajo establecido.

Tabla 2. *Tabla del plan de trabajo.*

Actividad	Meta Física		Meses							
	Cantidad	Unidad de Medida	1	2	3	4	5	6	7	
Investigar plataformas enfocadas en la educación con el uso de gamification	1	Mes	X							
Obtener Recursos y Modelos	1	Mes	X							
Investigar Técnicas de Gamification en la educación	1	Mes		X						
Revisión del plan académico	1	Mes		X						

Selección de técnicas de gamification	1	Mes			X				
Análisis de requisitos y especificaciones del software educativo	1	Mes			X				
Realizar bocetos de cómo será el software educativo	2	Mes			X	X			
Desarrollo del prototipo	3	Mes				X	X	X	
Realización de una evaluación pedagógica	1	Mes							X

8. REFERENCIAS BIBLIOGRÁFICAS

Akilli, G. K. (2011). Games and simulations: A new approach in education. *Gaming and Simulations: Concepts, Methodologies, Tools and Applications*, 150–167.

<https://doi.org/10.4018/978-1-59904-808-6.ch078>

Al-imamy, S., Alizadeh, J., & Nour, M. a. (2006). On the Development of a Programming Teaching Tool : The Effect of Teaching by Templates on the Learning Process.

Journal of Information Technology Education, 5, 271–283. Retrieved from

<http://www.editlib.org.ezproxy.psz.utm.my/p/111545/>

Azmi, S., Iahad, N. A., & Ahmad, N. (2015). Gamification in online collaborative learning for programming courses: A literature review. *Researchgate*, 10(23), 18087–18094.

Retrieved from

[https://www.researchgate.net/profile/Shahdatunnaim_Azmi/publication/2904601](https://www.researchgate.net/profile/Shahdatunnaim_Azmi/publication/290460188_Gamification_in_online_collaborative_learning_for_programming_courses_A_literature_review/links/56d4e20e08ae2cd682b94244/Gamification-in-online-collaborative-learning-for-progra)

[88_Gamification_in_online_collaborative_learning_for_programming_courses_A_literature_review/links/56d4e20e08ae2cd682b94244/Gamification-in-online-collaborative-learning-for-progra](https://www.researchgate.net/profile/Shahdatunnaim_Azmi/publication/290460188_Gamification_in_online_collaborative_learning_for_programming_courses_A_literature_review/links/56d4e20e08ae2cd682b94244/Gamification-in-online-collaborative-learning-for-progra)

Ballu, A., Yan, X., Blanchard, A., Clet, T., Mouton, S., & Niandou, H. (2016). Virtual

Metrology Laboratory for e-Learning. *Procedia CIRP*, 43, 148–153.

<https://doi.org/10.1016/j.procir.2016.02.110>

Buisman, A. L. D., & van Eekelen, M. C. J. D. (2014). Gamification in educational software development. In *Proceedings of the Computer Science Education Research Conference* (pp. 9–20).

Butt, P. (2016). Students' perceptions of game-based learning using CodinGame.

Caillois, R. (2001). *Man, Play and Games*. 1958. *Trans. Meyer Barash*. Urbana and Chicago: University of Illinois Press.

Craighead, J., Burke, J., & Murphy, R. (2008). Using the Unity Game Engine to Develop SARGE : A Case Study. In *Itsec* (Vol. 4552, p. 366). Retrieved from <http://www.robot.uji.es/research/events/iros08/contributions/craighead.pdf>

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness. *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11*, 9.
<https://doi.org/10.1145/2181037.2181040>

Díaz Cruzado, J., & Troyano Rodríguez, Y. (2013). El potencial de la gamificación en el ámbito educativo. *III Jornadas de Innovación Docente. Innovación Educativa*, 9.
Retrieved from https://fcee.us.es/sites/default/files/docencia/EL POTENCIAL DE LA GAMIFICACIÓN APLICADO AL ÁMBITO EDUCATIVO_0.pdf
<https://idus.us.es/xmlui/handle/11441/59067>

Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: a

systematic mapping study Badges. *Educational Technology & Society*. Retrieved from

https://scholar.google.com/citations?view_op=view_citation&continue=/scholar%3Fhl%3Den%26start%3D20%26as_sdt%3D0,33%26scilib%3D1&citilm=1&citation_for_view=EIOeQSYAAAAJ:EUQCXRtRnyEC&hl=en&oi=p%5Cnhttps://scholar.google.com/citations?view_op=view_citation&

- Fredes, C. A., Hernández, J. P., & Díaz, D. A. (2012). Potencial y Problemas de la Simulación en Ambientes Virtuales para el Aprendizaje. *Formación Universitaria*, 5(1), 45–56. <https://doi.org/10.4067/S0718-50062012000100006>
- Gallego-Durán, F. J., Villagrà-Arnedo, C., Llorens-Largo, F., & Molina-Carmona, R. (2017). PLMan: A game-based learning activity for teaching logic thinking and programming. *International Journal of Engineering Education*, 33(2), 807–815.
- Iquira Becerra, D. A. (2018). Implementación del laboratorio virtual inmersivo aplicado a la enseñanza de física usando técnicas de Gamification.
- Joskowicz, I. J., & Mingus, C. (2008). Reglas y Prácticas en eXtreme Programming. *Universidad de Vigo*, 1–22.
- Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming. *Universidad de Vigo. España*, 1–22. Retrieved from <http://iie.fing.edu.uy/~josej/docs/XP - Jose Joskowicz.pdf>
- Laster, J. (2010). At Indiana U., a class on game design has students playing to win. *The Chronicle of Higher Education*.

- Minnaard, C., & Minnaard, V. (2014). Evaluación por competencias en Entornos Virtuales de Aprendizaje. *WICC 2014 XVI Workshop de Investigadores En Ciencias de La Computación*, 903–906. Retrieved from http://trafficlight.bitdefender.com/info?url=http%3A//sedici.unlp.edu.ar/handle/10915/43712&language=en_US
- Orji, R., Mandryk, R. L., Vassileva, J., & Gerling, K. M. (2013). Tailoring persuasive health games to gamer type. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 2467. <https://doi.org/10.1145/2470654.2481341>
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137–172. <https://doi.org/10.1076/csed.13.2.137.14200>
- Satratzemi, M., Xinogalos, S., & Dagdilelis, V. (2003). An environment for teaching object-oriented programming: ObjectKarel. *Proceedings - 3rd IEEE International Conference on Advanced Learning Technologies, ICALT 2003*, 342–343. <https://doi.org/10.1109/ICALT.2003.1215114>
- Sims, Z., & Bubinski, C. (2011). Codecademy. URL= [Http://www. Codecademy. Com.](http://www.codecademy.com)
- Standard, I. (1993). IEEE Recommended Practice for Software Requirements Specifications, 830, 43.
- Unity Technologies. (n.d.). Unity. Retrieved January 19, 2018, from <https://unity3d.com/es>