

UNIVERSIDAD DE MATANZAS  
Facultad de Ciencias Técnicas  
Departamento de Informática



Trabajo de Diploma para optar el título de Ingeniero  
Informático

**TÍTULO: Videojuego educativo orientado al desarrollo de habilidades de lógica.**

Autor: Osniel Orlando Domínguez Pérez

Tutor: Ing. José Enrique Díaz

**Matanzas, 2019**

## **Pensamiento**

**“Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”**

**Albert Einstein**

**Dedicatoria.**

**A toda mi familia que siempre están a mi lado, a mi abuela, mis madres, mi padre y mi esposa.**

## **Agradecimientos**

- **A mi familia y amigos que siempre están presentes.**
- **A mi tutor y oponente por la atención prestada.**
- **A los profesores del departamento de informática que contribuyeron con sus ideas y consejos a esta investigación.**

## **Declaración de autoría**

Yo, Osniel Orlando Domínguez Pérez declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas "Camilo Cienfuegos", especialmente a la Facultad de Informática, a que hagan el uso que estimen pertinente de él.

Y para que así conste, firmo la presente a los 18 días del mes de junio del 2019.

---

Firma del Autor

---

Firma del Tutor

# Resumen

Actualmente las habilidades de lógica son necesarias en el transcurso de cualquier carrera universitaria, especialmente en el estudio de Ingeniería Informática ya que cuenta en su plan de estudio asignaturas como programación, matemática discreta e inteligencia artificial, por lo que se requiere de una herramienta para incrementar las habilidades anteriormente mencionadas. Se decide crear un juego educativo que por sus características brindará una forma más amena para lograr el objetivo mediante problemas, acertijos y puzzles. En el presente trabajo se mostrarán algunas de las características del funcionamiento de la aplicación para lo cual se utiliza la metodología de desarrollo SUM, la misma está orientada al desarrollo de videojuegos basada en SCRUM. Además, se emplea para una mejor comprensión artefactos como diagrama de casos de usos de sistema y diagramas de actividades.

# Summary

At the present day the logic abilities are necessary in the course of any university career, especially in the study of Computer Engineering in its study plan are subjects like programming, discreet mathematics and artificial intelligence, then is required a tool to increase the previously mentioned abilities. We decide to create an educational game that will offer a more interesting form to achieve the objective by problems, riddles and puzzles. In the presently work some of the characteristics of the operation of the application will be shown for that, we use the development methodology SUM, which is a methodology guided to development of video games based on SCRUM. In addition, we use for a better understanding use artifact like case diagram and activities diagrams.

# Tabla de Contenidos

Resumen .....	I
Summary .....	I
Introducción .....	1
Capítulo I MARCO TEÓRICO-REFERENCIAL .....	4
1.1. Antecedentes.....	4
1.2. La lógica .....	5
Lógica Formal .....	8
Lógica semiótica .....	8
Lógica deóntica .....	8
Lógica modal.....	8
Lógica temporal:.....	8
Lógica proposicional.....	8
Lógica cuantificacional .....	9
Lógica de clases .....	9
Lógica simbólica.....	9
Sistemas lógicos no clásicos .....	9
Lógica difusa .....	9
Lógica relevante .....	9
Lógica cuántica.....	9
Lógica matemática .....	9
Lógica de programación .....	9
Lógica de programación en los estudiantes .....	10
Lógica de predicados o cálculo de predicados .....	10
Lógica dialéctica.....	10

1.3. Videojuegos educativos y la enseñanza mediante juegos .....	11
Videojuegos.....	11
Videojuegos educativos .....	11
Aprendizaje basado en juegos.....	12
juegos serios (Serious games).....	13
Tipos de juegos serios .....	14
Edutainment .....	14
Juegos para entrenar (Training games) .....	14
Juegos de noticias (Newsgames) .....	15
Advergames .....	15
Gamificación.....	16
Objetivos de la gamificación.....	17
1.4. Metodología de desarrollo .....	18
Metodología utilizada .....	19
Estructura o ciclo de vida .....	20
Roles.....	23
1.5. Herramientas CASE .....	24
Visual Paradigm .....	24
1.6. Herramientas y tecnologías .....	24
Plataforma de desarrollo utilizada .....	26
Lenguajes de programación.....	27
QML .....	27
Ventajas .....	27
C++ .....	27
JavaScript .....	28
Ambiente de desarrollo integrado (IDE) .....	29

QTCreator .....	29
Herramientas para la edición de imágenes .....	29
Photoshop.....	29
Conclusiones del capítulo.....	30
Capítulo II Descripción de la solución .....	31
2.1. Propuesta del sistema .....	31
2.2. Etapa 1 Concepto .....	31
Tarea 1 Definir aspectos del juego.....	32
Tarea 2: Definir aspectos técnicos .....	32
Tarea 3: Definir aspectos del negocio.....	32
2.3. Etapa 2 Planificación .....	33
Tarea 2.1 Definir Roles .....	33
Tarea 2.2 Creación del cronograma.....	33
Tarea 2.3 Creación de la especificación del producto .....	40
Tarea 2.4 Priorización y estado del producto .....	41
Tarea 2.5 Definir presupuesto y costos.....	42
Costo real.....	43
2.4. Diseño de la solución propuesta .....	44
Arquitectura .....	44
Modelo/vista .....	44
Patrones de diseño .....	45
Patrones GRASP .....	45
Abstract Factory .....	45
Diseño de la Base de datos .....	46
2.5. Análisis de riesgos.....	47
Conclusiones del capítulo.....	52

Capítulo 3: Resultados y pruebas .....	53
3.1. Pruebas del software .....	53
Plan de pruebas.....	53
Tablas de pruebas .....	54
Pruebas Beta.....	55
Resultados de las pruebas Alfa y Beta .....	56
Estado de los errores .....	56
3.2. Pruebas de rendimiento.....	57
3.3. Impacto de riesgos .....	58
3.4. Resultados obtenidos .....	59
Análisis de los resultados obtenidos .....	60
3.5. Conclusiones del capítulo .....	60
Conclusiones Generales.....	62
Recomendaciones .....	63
Anexos.....	i
Anexo 1 Tablas de Pruebas .....	i
Anexo 2 Reportes de las Pruebas Beta.....	ix
Anexo 3 Imágenes de la aplicación .....	xiii

## Índice de tablas

TABLA 1 COMPARACIÓN ENTRE VIDEOJUEGOS EDUCATIVOS Y SOFTWARE TRADICIONAL FUENTE: (JIMÉNEZ ET AL., 2010) .....	12
TABLA 2 COMPARATIVA ENTRE UNITY Y QTQUICK .....	26
TABLA 3 ROLES DEL PROYECTO. FUENTE: ELABORACIÓN PROPIA .....	33
TABLA 4 CRONOGRAMA DEL PROYECTO. FUENTE: ELABORACIÓN PROPIA .....	38

TABLA 5 DESCRIPCIÓN DE ITERACIONES. FUENTE: ELABORACIÓN PROPIA.....	40
TABLA 6 ESPECIFICACIÓN DEL PRODUCTO. CARACTERÍSTICAS FUNCIONALES. FUENTE: ELABORACIÓN PROPIA .....	41
TABLA 7 ESPECIFICACIÓN DEL PRODUCTO. CARACTERÍSTICAS NO FUNCIONALES. FUENTE: ELABORACIÓN PROPIA .....	41
TABLA 8 PRIORIZACIÓN Y ESTADO DEL PRODUCTO. FUENTE: ELABORACIÓN PROPIA .....	42
TABLA 9 EQUIPO DE TRABAJO Y SALARIO. FUENTE: ELABORACIÓN PROPIA .....	43
TABLA 10 GASTOS. FUENTE: ELABORACIÓN PROPIA.....	43
TABLA 11 ANÁLISIS DE RIESGO. FUENTE: ELABORACIÓN PROPIA.....	51
TABLA 12 PLAN DE PRUEBAS. FUENTE: ELABORACIÓN PROPIA.....	54
TABLA 13 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	55
TABLA 14 RESULTADO DE LAS PRUEBAS BETA. FUENTE: ELABORACIÓN PROPIA .....	56
TABLA 15 ESTADO DE LOS ERRORES.....	57
TABLA 16 IMPACTO DE RIESGOS. FUENTE: ELABORACIÓN PROPIA.....	59
TABLA 17 ENCUESTA APLICADA. FUENTE: ELABORACIÓN PROPIA .....	59
TABLA 18 RESULTADOS DE LA ENCUESTA. FUENTE: ELABORACIÓN PROPIA.....	60
TABLA 19 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	I
TABLA 20 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	II
TABLA 21 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	II
TABLA 22 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	III
TABLA 23 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	III
TABLA 24 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	IV
TABLA 25 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	IV
TABLA 26 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	V
TABLA 27 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	V
TABLA 28 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	VI
TABLA 29 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	VI
TABLA 30 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	VII
TABLA 31 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	VIII
TABLA 32 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	VIII

TABLA 33 TABLA DE PRUEBA. FUENTE: ELABORACIÓN PROPIA.....	IX
TABLA 34 RESULTADO DE LAS PRUEBAS BETA. FUENTE: ELABORACIÓN PROPIA.....	X
TABLA 35 RESULTADO DE LAS PRUEBAS BETA. FUENTE: ELABORACIÓN PROPIA.....	XI
TABLA 36 RESULTADO DE LAS PRUEBAS BETA. FUENTE: ELABORACIÓN PROPIA.....	XII

## Índice de ilustraciones

ILUSTRACIÓN 1 RESULTADO FINAL DE LA COMPARATIVA ENTRE METODOLOGÍAS. FUENTE:(ARTEAGA, 2014).....	19
ILUSTRACIÓN 2 FASES DEL PROCESO. FUENTE:(ACERENZA ET AL., 2009).....	22
ILUSTRACIÓN 3 DIAGRAMA DE LA METODOLOGÍA SUM MODIFICADA. FUENTE:(MURILLO-SANCHEZ ET AL., 2018).....	23
ILUSTRACIÓN 4 : PATRÓN MODELO-VISTA. FUENTE: QT-PROJECT.ORG .....	45
ILUSTRACIÓN 5 DIAGRAMA ENTIDAD-RELACIÓN. FUENTE: ELABORACIÓN PROPIA .....	46
ILUSTRACIÓN 6 MÉTODO CON MÁS TIEMPO DE CORRIDA DE LA APLICACIÓN. FUENTE: REPORTE DE PRUEBA DE QTCREATOR ANALIZE .....	58
ILUSTRACIÓN 7 IMAGEN DEL JUEGO.....	XIII
ILUSTRACIÓN 8 IMAGEN DEL JUEGO.....	XIV
ILUSTRACIÓN 9 IMAGEN DEL JUEGO.....	XV
ILUSTRACIÓN 10 IMAGEN DE LA APLICACIÓN .....	XVI

# Introducción

En la carrera de Ingeniería Informática, algunas de las asignaturas más importantes son la Introducción a la programación, programación orientada a objetos, estructura de datos, para el desarrollo de las mismas se requiere que el estudiante tenga habilidades de lógica. Según lo expuesto en ([SCIENTISTS, 2006](#); [Serna, 2013](#)), la formación en lógica se debe realizar no sólo en el marco de las matemáticas, sino también en el de la formación en Ciencias Computacionales, con un aspecto inter y multidisciplinario para lograr la eficiencia.

Según el estudio realizado por la jefa del departamento de Ingeniería Informática, Msc. Mayli Estopiñán Lantigua y el Msc Yudelkis Valderrama Garrido en ([Lantigua., 2017](#)) y ([Garrido., 2019](#)) se determinó que este problema afecta a los estudiantes de la carrera de Ingeniería Informática de la Universidad de Matanzas Camilo Cienfuegos y se le concede gran importancia ya que las habilidades de lógica son utilizadas a lo largo de los estudios en asignaturas como Introducción a la programación, programación orientada a objetos, estructura de datos, Inteligencia Artificial, y Matemática Discreta, sobre esta última asignatura el máster Alberto Edgar Vásquez en su Tesis para obtener el título de Master en Educación Matemática ([Vasquez, 2007](#)) planteó “Sin embargo, los contenidos de la lógica computacional se encuentran plasmados en los libros de matemáticas discreta, y que, en ningún momento han sido diseñados pensando en el desarrollo de la lógica y de la lógica computacional de los estudiantes de informática.”. Y también estas habilidades son necesarias posteriormente en el desarrollo como profesional de un ingeniero informático. Según ([Moroni & Señas, 2005](#)) la complejidad de los programas que se desarrollan actualmente produce la necesidad de iniciar a los alumnos en un camino que los conduzca a utilizar efectivas técnicas de programación. Es importante para ello poner énfasis en el diseño previo. “Actualmente existe una tendencia internacional a la enseñanza de lenguajes de programación, pasando por alto los elementos básicos de la Lógica de Programación, que constituyen a su vez la base sobre la cual se sustenta la programación en sí. Esto fundamentalmente se debe a que “se asume conocido el conocimiento pretérito, lo cual ha ido creando un bache de formación y de pensamiento en los educandos, que solo se ocupan del nuevo conocimiento, pero no de su sustento”, según comenta a través de entrevista por correo electrónico, el Msc. Jesús Mesa Oramas, Analista Financiero de la Dirección Financiera y de Créditos, perteneciente a la Dirección General de Tesorería de CIMEX S.A en Cuba.”([Prado](#)). El master Alberto Edgar Vásquez en su Tesis para obtener el título de Master en Educación Matemática ([Vasquez, 2007](#)) planteó “La mayoría de los libros de lógica en la actualidad están dirigidos, principalmente, al desarrollo del pensamiento matemático. Esta condición ubica en desventaja a los estudiantes de informática, pues además de la lógica matemática necesitan la

lógica computacional que es uno de los pilares en el desarrollo de la programación y del estudio de las estructuras de datos que se erige como el paso inmediato en el estudio de la programación.”

Se define entonces como **problema investigativo**:

¿Cómo desarrollar las habilidades de lógica necesarias en los estudiantes de nuevo ingreso de la carrera de Ingeniería Informática durante su paso por la universidad?

La **hipótesis** en este trabajo es que si se elabora un juego educativo se contribuirá a desarrollar las habilidades de lógica necesarias en los estudiantes de nuevo ingreso de la carrera de informática durante su paso por la universidad.

De este modo se define como **objeto de estudio** el desarrollo de las habilidades de lógica necesarias en los estudiantes de nuevo ingreso en la universidad.

Enmarcándose, por tanto, el **campo de acción** en desarrollar las habilidades de lógica necesarias en los estudiantes de nuevo ingreso de la carrera de Ingeniería Informática durante su paso por la universidad. Para resolver el problema planteado, en este trabajo se traza el siguiente **objetivo general**:

Elaboración de un videojuego educativo para contribuir al desarrollo de habilidades de lógica necesarias en los estudiantes de nuevo ingreso durante su paso por la.

#### **Objetivos específicos:**

Buscar y analizar el trabajado relacionado con el desarrollo de las habilidades de lógica necesarias en los estudiantes de nuevo ingreso en la universidad.

- como punto de partida y antecedentes de la investigación.
- Definir las habilidades de lógica necesarias durante su paso por la universidad en los estudiantes.
- Desarrollar un prototipo del juego educativo.
- Demostrar la viabilidad y funcionamiento del prototipo del juego.

Entre las **tareas** planteadas para dar cumplimiento a los objetivos propuestos se encuentran:

- Estudiar y analizar los fundamentos teóricos que sustentan la investigación.
- Aplicar métodos de investigación de Ingeniería de Software en el desarrollo de la tesis.
- Seleccionar la metodología y tecnologías a emplear para desarrollar el prototipo.
- Diseñar e implementar la solución
- Realizar pruebas al prototipo y validar los resultados.

Para darle solución al problema planteado y cumplir los objetivos trazados se utilizaron en la investigación los siguientes **métodos teóricos**:

- Lógico – Histórico: analizar los antecedentes y evolución de los juegos educativos para desarrollar la lógica.
- Análisis de documentos: se consultó literatura especializada y documentos rectores de la teoría y metodología SUM. Nos permitió realizar un análisis detallado sobre las habilidades de lógica a las que apunta nuestra investigación y de las herramientas del marco teórico.
- Modelación: para comprender los procesos de desarrollo creando abstracciones con vistas que expliquen la realidad.

Y como **método empírico** se utilizó:

- La observación: permitió recopilar datos sobre el uso de las nuevas tendencias de los estudiantes para así desarrollar el juego según sus preferencias.

Mientras que las **técnicas** para la recopilación de información utilizadas fueron:

- Entrevista: Se entrevistaron a estudiantes aleatorios para saber el estado de sus habilidades de lógica
- Encuesta: Para conocer las opiniones de los jugadores sobre la herramienta creada.

# Capítulo I MARCO TEÓRICO-REFERENCIAL.

En este capítulo se analizan las bases teóricas que sustentan y fundamentan la investigación. Se abordan elementos que ayudan a entender los procesos objeto de estudio, así como los trabajos relacionados con éstos. Se explica la metodología que permitió dar respuesta a la problemática existente, y se exponen, a su vez, las herramientas utilizadas y los argumentos que facilitaron su selección.

## 1.1. Antecedentes

En el mundo actual se han creado variedad de juegos que de alguna forma ejercitan la lógica. En su mayoría los desarrolladores persiguen como meta obtener beneficios económicos mediante compras dentro de la aplicación, mediante el pago de licencias para su uso o mediante la compra del propio juego, no priorizando el ejercitamiento o desarrollo de dicha habilidad ni con fines educativos.

Los videojuegos de lógica, también conocidos como videojuegos de inteligencia o videojuegos de puzzle, son un género de videojuegos que se caracterizan por exigir agilidad mental al jugador. Pueden involucrar problemas de lógica, matemáticas, estrategia, reconocimiento de patrones, completar palabras o hasta simple azar.

Un ejemplo de estos son los llamados juegos educativos que no son más que un material multimedia interactivo por medio del cual se puede aprender uno o varios temas. En otros trabajos a lo largo de las dos últimas décadas se presenta una visión general sobre el uso educacional de los videojuegos, describiendo su utilidad en diversas áreas del conocimiento ([Bergman, 2003](#)) ([Cavallari, 1992](#)) ([Freitas, 2005](#)) ([Egenfeldt-Nielsen, 2006](#)). Una característica importante de un juego educativo, es que el conocimiento es adquirido de una forma implícita, es decir, los jugadores no se percatan de que al estar jugando van adquiriendo una serie de conocimientos concretos, sino que se van apropiando de éstos en el transcurso natural del juego. ([Padilla & Collaso, 2012](#)) Además de todo, un juego educativo permite a los estudiantes desarrollar habilidades que de otra manera quizás no adquieran con la misma facilidad.

Existe variedad de géneros en cuanto a juegos, como son de aventura, lógica, entre otros. El proyecto a desarrollar en la Universidad de Matanzas contempla las dos categorías antes mencionadas. Se tomó como referencia los siguientes juegos:

- River Crossing iq: es un juego de problemas de lógica de cruce (Lógica de cruce).([Entertainment, 2019](#))

- Lightbot: es un juego de puzles que utiliza elementos básicos de la programación. ([LLC, 2019](#))
- Human Resource Machine: es un juego de puzle de la compañía Tomorrow Corporation. ([T. Corporation, 2019](#))

Hasta la fecha en Cuba, el equipo de desarrollo no encontró ninguna aplicación conocida con estas características, a pesar de que se han elaborado juegos educativos como:

- La Neurona: La neurona es un videojuego para móviles con el objetivo de probar nuestros conocimientos de cultura general y, sobre todo, cubana. Está basado en el popular programa televisivo El selecto club de la neurona intranquila. ("[La neurona.](#)" 2019)
- Tutu2: Lo integran cuatro mini juegos que entrenarán la memoria, la agilidad, la apreciación visual y sonora en los niños. A través de la enseñanza de vocabulario en inglés sobre diferentes temáticas ("[Tutu2.](#)" 2019)
- Equipo 009. La Invasión de los Pálidos: El pueblo en el que viven Jaimito, Samy y XP ha sido invadido por los Pálidos. Pero no se saldrán con la suya porque los tres niños son agentes, forman el Equipo 009, y con sus distintos poderes le darán una lección colorida a los paliduchos.

También se han creado aplicaciones que no son juegos como es el editor de algoritmos de Norma Moroni – Perla Señas. Este programa tiene como punto negativo el tiempo que los estudiantes tienen que dedicar a aprender el trabajo con el software según lo planteado en las conclusiones de la puesta en práctica de este programa ([Moroni & Señas, 2005](#)).

Algunas universidades pioneras en la investigación y el desarrollo de la informática educativa han realizado algunas experiencias como son: Proyecto EIDOS (Educación informática e inteligencia artificial para el desarrollo de la inteligencia humana) por la Universidad Nacional, Proyecto Apolonio (Sistema tutorial inteligente para la solución de problemas en matemáticas) por la Universidad EAFIT. La Universidad Industrial de Santander (UIS) ha realizado proyectos en tres áreas básicas: anatomía, matemáticas básicas y avanzadas. El grupo de informática educativa Uniandes ha desarrollado algunos proyectos en informática educativa en las áreas de derecho, medicina y preescolar (Meléndez Acuña, 1995)

## 1.2. La lógica

Aunque existían muchas definiciones para el campo del estudio de la lógica, no difieren esencialmente unas de otras, hay cierto consenso entre los autores de que la lógica tiene, por objeto de estudio las

leyes generales del pensamiento, y las formas de aplicar estas leyes correctamente. Aunque ha sido encontrado en la India, textos sobre ese asunto, escritos en épocas remotas, acerca de que la lógica haya nacido en la Antigua Grecia, por vuelta del siglo IV antes de cristo los primeros trabajos sobre lógica fueron debido a Permenides, Zeñao, y a un grupo conocido como "sofistas", pero el verdadero creador de la lógica es, sin duda, Aristóteles, pues fue el quien sistematizó y organizó ese conocimiento, elevándolo a la categoría de ciencia. En su obra llamada "Organum", que en su traducción libre significa ferramenta, Aristóteles estableció principios tan generales y tan sólidos que dominó el pensamiento occidental durante dos mil años, y hasta hoy han sido considerados válidos. Aristóteles tenía como objetivo la búsqueda de la verdad, y por eso procuraba caracterizar los instrumentos que servían de razón para esa búsqueda. En otras palabras, Aristóteles preocupaba por las formas de raciocinio, que, a partir de conocimientos considerado verdaderos, permitían obtener nuevos conocimientos. Creó, para la lógica un formulario de leyes generales de un grupo de conceptos y juicios que llevarían al descubrimiento de nuevas verdades.(Carvalho, 2007)

Según el diccionario de sinónimos y antónimos ("Diccionario de Sinónimos y Antónimos") sinónimos de lógica son: Razón, razonamiento, dialéctica, método. Y según (Prado) plantea que el término proviene del griego LOGOS que significa: Idea, Palabra, Regularidad. También (Prado) menciona que Omar Iván Trejos Buriticá, Decano de Ingenierías de la Universidad Tecnológica de Pereira en Colombia y Máster en Informática Educativa; en su libro La Esencia de la Lógica de Programación la define como "la forma más OBVIA y más FÁCIL de hacer algo" (Trejos, 2004).

En el libro (Pascual, 2005) plantea el autor que la lógica es la ciencia que estudia el razonamiento, pero no desde una perspectiva psicológica, es decir, no busca entender los efectos que producen los razonamientos sobre sus propios autores, ni mucho menos entender el comportamiento de los propios autores, sino que estudia el producto o resultado de la actividad de razonar.

El doctor Redmond Walter plantea en su libro (Walter, 1999) "La lógica es la ciencia que estudia (y el arte que pone en práctica) la validez de los argumentos. Lo hace identificando las reglas que permiten la inferencia de consecuencias. "

"La lógica como tal no formula enunciados significativos, sino que se limita a sacar deducciones de unas premisas, pero en realidad no añade nada nuevo a estas últimas. Aunque la lógica no añada nada nuevo a las proposiciones a las que se aplica, esto no significa que con ella no se puedan adquirir nuevos conocimientos. Y por supuesto, la lógica también es fuente de diversión y entretenimiento."(Drösser, 2013)

La lógica se ha convertido en uno de los fundamentos matemáticos y en una base formal indispensable en todo informático. La formalización del conocimiento y la automatización de las formas de razonamiento son primordiales en muchas áreas de la informática. “La importancia de la lógica en los currículos de Informática va tomando cuerpo propio debido a sus aplicaciones en contextos específicos tales como la programación, la ingeniería del software, el diseño de sistemas de bases de datos y la inteligencia artificial.”(Vasquez, 2007)

Según lo planteado en (Medina) podremos desarrollar la lógica mediante: Problemas de matemáticas, interrogación socrática, demostraciones científicas, ejercicios para resolver problemas lógicos, clasificaciones y agrupaciones, creación de códigos, juegos y rompecabezas de lógica, lenguaje de programación, cuantificaciones, presentación lógica de los temas, heurística.

Según (Delavy, 2017) la lógica es la técnica utilizada para desarrollar instrucciones en una secuencia para lograr determinado objetivo.

Es una ciencia formal que estudia los principios de la demostración e inferencia válida. Así como el objeto de estudio tradicional de la química es la materia, y el de la biología la vida, el de la lógica es la inferencia.(TENJICA, 2016). La inferencia es el proceso por el cual se derivan conclusiones a partir de premisas. La lógica investiga los fundamentos por los cuales algunas inferencias son aceptables, y otras no. Cuando una inferencia es aceptable, lo es por su estructura lógica, y no por el contenido específico del argumento o el lenguaje utilizado.

Podemos clasificar los tipos de lógica desde dos puntos de vista, la lógica clásica y la moderna. Sin embargo, dicha clasificación sólo sirve para efectos históricos, de ahí que mejor proponemos subdividir, los distintos tipos de lógica, respecto a los objetos que trata, adaptando la estructura de la lógica dada por (TENJICA, 2016).

La Lógica Formal es conocida también como lógica clásica o aristotélica, se imputa al filósofo Aristóteles ser el creador de la misma, aunque ya existían antecedentes en Parmenides y Zeñao. Así mismo con el paso del tiempo, con la evolución de algunas corrientes matemáticas, específicamente las aportaciones realizadas por los matemáticos Euler y Boole al álgebra, se da inicio a la Lógica Moderna, Lógica Matemática, Lógica Simbólica o Logística y posteriormente a partir de 1960, la Lógica de Programación. De esta lógica moderna, se desprenden las siguientes: la semiótica, lógica deóntica, modal, cuantificacional y proposicional, lógica de clases.

## **Lógica Formal**

Según algunos la lógica clásica estaría constituida por un conjunto de cálculos lógicos equivalentes al cálculo presentado por Bertrand Russell y Alfred N. Whitehead en sus Principia Mathematica (1910-1913). Otro punto de vista es el que dice que la lógica clásica es aquella que se desarrolla desde Aristóteles hasta las aportaciones de Alfred Tarski, hacia mediados de los años 30 del siglo XX. Cualquiera de las dos definiciones nos viene bien, pues resulta que todos los sistemas lógicos a los que llamamos “lógica clásica” comparten la propiedad de ser equivalentes al cálculo de Principia y, a su vez, todos ellos son producto y han nacido gracias a las aportaciones realizadas por los lógicos desde Aristóteles a Tarski. ([TENJICA, 2016](#))

## **Lógica semiótica**

Es la lógica de los símbolos y se divide en tres partes: sintaxis, semántica y pragmática. La primera trata de las relaciones de los símbolos entre sí, prescindiendo de su contenido. La segunda trata de las relaciones entre el símbolo y lo que significa. La tercera trata de las relaciones entre el símbolo y el sujeto que lo utiliza. ([TENJICA, 2016](#))

## **Lógica deóntica**

Se formaliza a través de conceptos relacionados con el deber. Este tipo de lógica se utiliza en el Derecho, infiriéndose del mismo, la denominada Lógica de las normas. se ocupa de las nociones morales de obligación y permisibilidad. ([TENJICA, 2016](#))

## **Lógica modal**

Las lógicas modales están diseñadas para tratar con expresiones que califican la verdad de los juicios. Así, por ejemplo, la expresión «siempre» califica a un juicio verdadero como verdadero en cualquier momento, es decir, siempre. No es lo mismo decir «está lloviendo» que decir «siempre está lloviendo». Trata con las nociones de necesidad, posibilidad, imposibilidad y contingencia. ([TENJICA, 2016](#))

## **Lógica temporal:**

Abarca operadores temporales como «siempre», «nunca», «antes», «después», etc. ([TENJICA, 2016](#))

## **Lógica proposicional**

Analiza los razonamientos formalmente válidos partiendo de proposiciones y conectivas proposicionales (operadores lógicos). Es el nivel más básico de análisis lógico. Se analizan las relaciones que se dan entre los enunciados o las proposiciones; es, pues, una lógica interproposicional, no intraproposicional. ([TENJICA, 2016](#))

## **Lógica cuantificacional**

Estudia de manera más detallada los predicados a través del uso de cuantificadores que expresan cantidad (todos o algunos). ([TENJICA, 2016](#))

## **Lógica de clases**

Relaciona conceptos con propiedades (sujeto y predicado), estudia además las implicaciones de unas clases con otras, las cuales suelen ser representados gráficamente mediante círculos (mejor conocidos como diagramas de Venn) empleando la denominada “álgebra booleana”. ([TENJICA, 2016](#))

## **Lógica simbólica**

Emplea un lenguaje artificial en la que simboliza las proposiciones generalmente con las letras p, q, r, s, t utilizando de operadores lógicos, también llamados conectores, funtores, juntores, para poder construir formulas operando sobre las variables proposicionales y las proposiciones complejas. ([TENJICA, 2016](#))

## **Sistemas lógicos no clásicos**

Los sistemas lógicos no clásicos son aquellos que rechazan uno o varios de los principios de las lógicas clásicas ([TENJICA, 2016](#)). Algunos de estos sistemas son:

**Lógica difusa:** es una lógica plurivalente que rechaza el principio del tercero excluido y propone un número infinito de valores de verdad. ([TENJICA, 2016](#))

**Lógica relevante:** es una lógica paraconsistente que evita el principio de explosión al exigir que para que un argumento sea válido, las premisas y la conclusión deben compartir al menos una variable proposicional. ([TENJICA, 2016](#))

**Lógica cuántica:** Desarrollada para lidiar con razonamientos en el campo de la mecánica cuántica; su característica más notable es el rechazo de la propiedad distributiva. ([TENJICA, 2016](#))

## **Lógica matemática**

“Es la capacidad para usar los números de manera efectiva y de razonar adecuadamente. Incluye la sensibilidad a los esquemas y relaciones lógicas, las afirmaciones y las proposiciones, las funciones y otras abstracciones relacionadas.” ([Gardner, 1987](#)).

## **Lógica de programación**

Es la base de todo el conocimiento en programación, ya que con la lógica aprendes a escribir un código para que la computadora interprete correctamente. Es decir que aprendes a comunicarte con la máquina

a partir de un lenguaje. Según ([Delavy, 2017](#)) Es la organización y planificación de instrucciones en un algoritmo, con el objetivo de tornar viable la implementación de un programa o software. La lógica de la programación es la organización coherente de las instrucciones del programa para que su objetivo sea alcanzado. Ese es el objetivo de todo programador, crear un algoritmo, una estructura, capaz de realizar operaciones y ser ejecutado por una máquina.

### **Lógica de programación en los estudiantes**

“El estudiante está aprendiendo a solucionar problemas y a diseñar sus algoritmos bajo una lógica tradicional (secuencial y procedimental) y luego, para desarrollar los componentes de software, tiene que aprender a modelar de nuevo, pero esta vez desde el punto de vista objetual, al momento de su aplicación”.([GITOO, CASTRO, MURILLO, & PALACIO, 2001](#)) Esto evidencia que, ante las actuales metodologías de enseñanza, hay una desarticulación entre la lógica de programación y las estructuras de datos, frente a los lenguajes de programación orientados por objetos, que son los que requiere la industria del software.

### **Lógica de predicados o cálculo de predicados**

Es un sistema formal diseñado para estudiar la inferencia en los lenguajes de primer orden. Los lenguajes de primer orden son, a su vez, lenguajes formales con cuantificadores que alcanzan sólo a variables de individuo, y con predicados y funciones cuyos argumentos son sólo constantes o variables de individuo.

Con la lógica de predicados podemos representar conceptualizaciones que contienen relaciones entre objetos (como las relaciones «padre» e «hijo»). Ahora bien, a veces la conceptualización también expresa relaciones entre relaciones, o propiedades (relaciones de grado 1) de relaciones. Por ejemplo, «padre es una relación familiar». A la lógica que sólo permite representar relaciones entre objetos se le llama de primer orden, la que permite relaciones entre relaciones, de segundo orden, y así sucesivamente. ([TENJICA, 2016](#))

### **Lógica dialéctica**

Finalmente existe otro tipo de lógica que es la dialéctica, aunque ésta no la podemos considerar como integrante de la lógica moderna, pues esta “no tiene un contenido formal, sino ideológico; ni es “pasiva” como la lógica formal, sino que es activa, al obtener principios racionales a través de la interpretación de la historia”([TENJICA, 2016](#)), utilizando como su estructura en su discurso, la tesis, seguida de la antítesis y su respectiva conclusión denominada síntesis; teniendo sus antecedentes desde los griegos con SOCRATES y PLATÓN quienes la concibieron como una técnica de discusión y de obtención de

conclusiones, siendo la misma también estudiada y empleada por algunos filósofos como KANT, HEGEL, MARX, entre otros más.

De los mencionados anteriormente en la aplicación el equipo de desarrollo se centrará en la lógica modal, proposicional, cuantificacional, de clases, matemática y de programación por sus aplicaciones en cualquier carrera de ingeniería especialmente en la ingeniería informática.

### **1.3. Videojuegos educativos y la enseñanza mediante juegos**

#### **Videojuegos**

Los videojuegos han surgido hace más de 40 años. El término “video juegos” refiere a un juego digital que requiere una interacción entre el humano y algún tipo de hardware, el cual provee una retroalimentación visual en un dispositivo de video.

Según Serdar Aslan en “Digital Educational Games: Methodologies for Development and Software Quality” ([Aslan, 2016](#)) los juegos digitales empezaron con El Pong en 1970 y ha crecido exponencialmente, el mercado vale billones de dólares y estaba previsto para el 2017 alcanzar los 78 billones.

Estos pueden ser categorizados según ([Aslan, 2016](#)) como acción, aventura, pelea, puzzles, juegos de rol, deporte y estrategia. Actualmente los desarrolladores combinan dichas categorías para formar nuevos juegos más interesantes.

#### **Videojuegos educativos**

Un videojuego educativo es un material multimedia interactivo por medio del cual se puede aprender uno o varios temas. Una característica importante de un videojuego educativo, es que el conocimiento es adquirido de una forma implícita, es decir, los jugadores no se percatan de que al estar jugando van adquiriendo una serie de conocimientos concretos, sino que se van apropiando de éstos en el transcurso natural del videojuego. Introducir el videojuego como un material didáctico en el aula ha sido tema de discusión durante muchos años, en donde podemos encontrar sus partidarios y detractores; estos últimos impulsados por etiquetamientos generalizados e injustificados hacia los videojuegos con temáticas de violencia, adicción, aislamientos y sexismos; pero realmente no hay estudios científicos que demuestren que el uso de este tipo de videojuegos pueda desencadenar conductas agresivas o patológicas en los jugadores. Según ([Ferguson, 2010](#)), es todo lo contrario, el videojuego actuará como un medio para descargar tensiones produciendo un efecto tranquilizador que disminuirá las posibilidades del jugador para cometer un acto violento, "(...) ¿Es el videojuego el que desencadena conductas violentas o son jugadores violentos los que acceden a este tipo de contenidos?"([Ferguson, 2010](#))

Actualmente los videojuegos están siendo utilizados en los centros educativos. Además de todo, un videojuego educativo permite a los estudiantes desarrollar habilidades que de otra manera quizás no adquieran con la misma facilidad.

La informática educativa nace en los años 80, y desde 1990 aumentó su utilización gracias a diferentes acciones adelantadas por el gobierno nacional y el Servicio Nacional de Aprendizaje (SENA). (Jiménez, Rico, Méndez, Ceron, & Betancourt, 2010)

Los estudiantes pueden desarrollar su capacidad de encontrar diferentes maneras de resolver los problemas que encuentran a través del juego. A menudo, los jugadores podrán encontrar que requieren unas habilidades específicas más adelante en el juego, y por lo tanto están obligados a mantener y perfeccionar sus habilidades para su uso posterior. Los videojuegos suelen ofrecer premios instantáneos para tener éxito en la resolución de un problema.

Por lo tanto, los videojuegos se pueden utilizar como una alternativa al aula, manteniendo niveles de dificultad dependiendo de la edad y la madurez de cada niño. Si un juego tiene éxito, los jugadores se toman su tiempo para desarrollar conocimientos y las etapas del juego se reproducirán durante muchas más veces con gran atención.

<b>Videojuegos Educativos</b>	<b>Software Tradicional</b>
<b>El enfoque es la diversión con aprendizaje</b>	El enfoque es la funcionalidad
<b>Están diseñados para vivir una experiencia</b>	Están pensados para el desarrollo de procesos eficientes
<b>Facilidad en el uso y exigente en el nivel de estrategia, competencia y récords</b>	Facilidad de uso
<b>Exigente en el diseño de interfaz</b>	Interfaz de usuario amigable

*Tabla 1 Comparación entre videojuegos educativos y software tradicional Fuente: (Jiménez et al., 2010)*

### **Aprendizaje basado en juegos**

Una de las principales aportaciones de los juegos al desarrollo humano es el aprendizaje que se adquiere a través de ellos. Según (Margarida Romero, 2015) los juegos son una forma de aprendizaje activo que permite al estudiante tener un control de las actividades del juego y emprender interacciones. Aprendizaje basado en juegos estimula actividades de aprendizaje construyendo compromisos y retos para cumplir los objetivos de aprendizaje.

Muchas de las características combinadas y diseñadas efectivamente en juegos educativos puede enseñar muchas cosas en una forma cautivadora y motivante. Juegos pueden ser usados de una forma de expandir las habilidades cognitivas, como también en forma de plataforma para el desarrollo de nuevas habilidades o ejercitación de las mismas. Juegos también pueden ser usados para enseñar a viejos sujetos en nuevas formas. ([SCIENTISTS, 2006](#))

Además de todas las habilidades que permiten desarrollar, hay un importante peso en la adquisición de valores de una forma intuitiva a través de los procesos de experimentación. Esta es una diferencia muy fundamental que los distingue de las formas de educación clásica, como la escuela. Los videojuegos están contruidos para poder experimentar y permiten ver las consecuencias de los actos del jugador de una forma casi inmediata, pero protegiéndole de los efectos que tendrían en el mundo real. Esto fomenta que el usuario pueda jugar con posiciones morales y éticas con el fin de aprender no sólo qué sino el porqué de esas posiciones. Asimismo, los videojuegos son benévolos, al permitir reiniciar la situación en caso de llegar a un punto muerto o haber cometido una equivocación fatal, lo que lleva a una situación de aprendizaje sin precedentes. ([SCIENTISTS, 2006](#))

### **juegos serios (Serious games)**

“Juegos en los cuales la educación (en sus formas variadas) es el objetivo primario, por encima del entretenimiento” ([Margarida Romero, 2015](#)). Los juegos serios son basados en la idea de conectar un propósito serio a conocimientos y tecnologías de la industria de los videojuegos. Estos juegos se expanden por muchos campos como son la educación, terapia, propaganda, defensa, investigaciones donde utilizan el atractivo de los juegos para servir a un propósito serio.

En ([Tardón, 2014](#)) se define a los juegos serios como una evolución o especialización del videojuego de entretenimiento, no una categoría estanca independiente. También plantea que son aquellos programas que comparten todas las características de los videojuegos de entretenimiento, pero incorporando al objetivo lúdico que exclusivamente tienen éstos, el de crear un impacto directo en los valores, actitudes o habilidades del usuario que modifique su experiencia externa al videojuego.

El concepto de jugar serio es una de las principales críticas a los videojuegos serios. Desde ciertos sectores del desarrollo de videojuegos de entretenimiento se duda de la pertinencia de los videojuegos serios como videojuegos, ya que, en algunos casos, no cumplen las premisas de nuestra definición. ([Tardón, 2014](#)) Este problema entre serio y entretenimiento está creado por una incorrecta dicotomía de las dimensiones de ambos conceptos, al entender que lo contrario de serio es divertido, siendo un tema recurrente en el ámbito del juego. Si nos centramos en los términos serio y divertido, ambos son extremos de dimensiones diferentes y que no interactúan entre sí. El problema se ha creado porque, en

la práctica, una parte muy importante de lo serio en la sociedad actual se ha presentado normalmente en un formato de sobriedad y aburrimiento, y lo divertido ha tendido hacia un ámbito percibido como más banal. Pero sin duda, se defiende que la mejor forma de que lo serio sea efectivo e impacte en el usuario es a través de la diversión. Para conseguir ese objetivo los videojuegos son un medio ideal.

### **Tipos de juegos serios**

Para definir los tipos de juegos serios nos basaremos en su objetivo al igual que Carlos Gonzales en su libro digital "VIDEOJUEGOS PARA LA TRANSFORMACIÓN SOCIAL." (Tardón, 2014)

#### **Edutainment**

Son videojuegos serios cuyo objetivo es la transmisión de conceptos, ideas y conocimientos. Siempre han estado muy vinculados a la educación formal, aunque también se han ido extendiendo a la no formal. Habitualmente han sido patrocinados desde el ámbito público, al ser la aplicación de los videojuegos en el aula un proyecto de diversas administraciones en los últimos años (SCIENTISTS, 2006).

En la actualidad se ha diversificado a otras áreas, siendo un importante foco de desarrollo los videojuegos para la salud (Martínez. & Martín., 2012), cuyo objetivo es dotar de herramientas al ámbito hospitalario o al de prevención de riesgos para la salud pública, utilizando un medio que pueda atraer e impactar a la generación digital. En los videojuegos para el aprendizaje de conceptos no sólo se han utilizado videojuegos serios, sino que existen muchos proyectos cuyo objetivo ha sido aplicar videojuegos de entretenimiento como herramienta de formación.

#### **Juegos para entrenar (Training games)**

Este tipo de videojuegos serios ha sido muy fomentado desde las empresas privadas, que han visto en ellos una forma más barata de poder formar a sus empleados (SCIENTISTS, 2006). Son videojuegos cuyo objetivo es el entrenamiento de habilidades que posteriormente será necesario generalizar desde el plano virtual al mundo real. El hecho de poder proporcionar experiencia práctica en situaciones que se pueden encontrar en su ámbito laboral, pero sin los riesgos que puede suponer un error en el mundo real, los ha convertido en un medio idóneo de entrenamiento. Una última línea de training games, se refiere a aquellos videojuegos serios cuyo objetivo es la alfabetización digital y el fomento de las habilidades necesarias para el uso efectivo de elementos informáticos. Los trainings games son videojuegos que aportan el valor motivacional del videojuego unido al potencial como simulación, siendo una herramienta muy potente en el ámbito del coaching.

## **Juegos de noticias (Newsgames)**

Los newsgames están muy vinculados a los nuevos modelos de periodismo, sobre todo al periodismo interactivo. Utilizan la potencialidad de participación con el fin de conseguir un impacto mayor y más significativo del mensaje, además de una mejor comprensión de la situación a través del proceso de juego. Este tipo de videojuego está siendo desarrollado desde diversos ámbitos: desde el más obvio, como pueden ser los medios de comunicación a campañas de sensibilización de ONGs o creadores de videojuegos que quieren hacer llegar su opinión sobre un tema. Los newsgames se suelen dividir en dos grandes grupos: los videojuegos de noticia y videojuegos documentales. Existen situaciones en las que los newsgames y el edutainment están separados por una frontera muy fina. Es lógico al tratar ambos, en muchas ocasiones, los mismos conceptos desde diversas perspectivas. Se ha buscado una forma sencilla de diferenciarlos al entender que, mientras el edutainment intenta educar al usuario en algo, principalmente conceptos, los newsgames buscan más informar y hacer reflexionar al usuario, impactando sobre todo el marco de los valores.

## **Advergames**

El último tipo de videojuego serio es el que ha surgido de los proyectos de publicidad interactiva. Este tipo tiene como premisa que la participación del usuario final en una actividad publicitaria hace que el mensaje sea más significativo. Al observar la potencia de implicación de los usuarios de los videojuegos, y el acceso que supondría a una generación que está huyendo de los medios tradicionales, forzaron a las empresas de comunicación a ampliar sus estrategias hacia este sector. En su artículo sobre la interacción de la publicidad con los videojuegos, ([Noguero, 2010](#)) afirma que la potencia del advergaming está basada en los siguientes puntos:

- a) Alta exposición a la marca e integración de la marca, al estar los usuarios dentro de un juego con todo el imaginario de ella.
- b) Atención máxima del usuario, un efecto colateral de la estructura de los videojuegos.
- c) Predisposición positiva por parte del usuario, sobre todo en la generación digital.
- d) Interactividad o reactancia con el público, ya que los videojuegos son productos participativos.
- e) Memorabilidad, al involucrar al usuario en la acción lo que ocurre se recuerda mejor.
- f) Virilidad, o difusión del mensaje a través de los mismos usuarios o de boca a boca. Este tipo de programas también suponen una cierta duda sobre la ética de su desarrollo, al estar dirigidos normalmente a menores de edad, pero esto es un problema clásico del marketing.

Para poder diferenciar entre las cuatro categorías anteriores podría entenderse que, ante el concepto de “lavarse las manos es saludable”, los distintos tipos de videojuego serían los siguientes: un edutainment se centraría en que el usuario entendiera que al lavarse las manos se matan las bacterias; el training game en cómo lavarse de forma correcta las manos; el newsgame en cómo se expande una enfermedad al no lavarse las manos por el contagio en un país concreto del tercer mundo; y el advergame en presentar una marca concreta de jabón que es mucho más eficaz que las otras. El tema es el mismo pero la perspectiva diferente.

## **Gamificación**

La gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional. Sirve para absorber conocimientos, mejorar alguna habilidad, recompensar acciones concretas. Ha tenido una enorme popularidad en los últimos años, sobre todo en entornos digitales y educativos. “Con una evolución cuidada, la gamificación puede convertirse en la ingeniería memética del siglo XXI” (Tardón, 2014).

Es un movimiento de reciente creación que utiliza las estructuras de jugabilidad de los videojuegos para aplicarlas al mundo real (Lee & Hammer, 2014).

La gamificación se basa en el uso de elementos del diseño de videojuegos en contextos que no son de juego para hacer que un producto, servicio o aplicación sea más divertido, atractivo y motivador (Ortiz-Colón, Jordán, & Agredal, 2018).

De forma mayoritaria, los autores coinciden en señalar la gamificación como un factor fundamental para aumentar la motivación de los usuarios. Una de las claves principales al aplicarla es que los alumnos tengan perfectamente asimiladas las dinámicas de juego, ellas tienen por objeto implicar al alumno a jugar y seguir adelante en la consecución de sus objetivos mientras se realiza la actividad.

En función de la dinámica que se persiga, el profesor deberá explotar más unas u otras. Por ejemplo, si el maestro busca despertar el interés por el juego en el alumno deberá aplicar la dinámica de la recompensa. Si, por otra parte, busca atraer el interés sobre la actividad el maestro puede aplicar la dinámica de la competición que, aunque no es vista como una cualidad positiva, es un buen instrumento en el ámbito educativo.

Esta última, tiene la ventaja de poder realizarse de forma individual, por parejas o en grupo. También utiliza el cooperativismo, que no es más que otra forma de competir, pero, en este caso, en grupo. Esta misma actividad cuenta también con la dinámica de la solidaridad al fomentar la ayuda mutua entre compañeros de una manera altruista.

En los estudios sobre el tema ([Tardón, 2014](#)),([Lee & Hammer, 2014](#)) se afirma que existen cinco características principales de las técnicas de gamificación:

- a) Convierte tareas tediosas y-o aburridas en atractivas.
- b) Fomenta la participación del usuario.
- c) Tiene unos fuertes vínculos con la estructura social y lo social media, al tener sus orígenes una fuerte relación con la web social.
- d) Fideliza a los usuarios a partir de refuerzos y castigos.

“La gamificación puede hacer de la educación una actividad inmersiva, que provoque en los alumnos una sensación de dedicación absoluta”. ([Ortiz-Colón et al., 2018](#))

### **Objetivos de la gamificación**

La idea de la gamificación consiste en intentar transformar “aquello que se debe hacer” en “lo que se está haciendo” y posteriormente en “lo que se quiere hacer”, en situaciones en las que entre estas categorías existe un abismo ([Lee & Hammer, 2014](#)) y de esta forma mejorar la implicación del usuario en la tarea; esta mayor implicación y motivación siempre va de la mano de un aumento neto de rendimiento y calidad.

Cualquier actividad realizada en contexto de la gamificación busca lograr tres claros objetivos: por un lado, la fidelización con el alumno al crear un vínculo con el contenido que se está trabajando. Por otro lado, busca ser una herramienta contra el aburrimiento y motivarles. Finalmente, quiere optimizar y recompensar al alumno en aquellas tareas en las que no hay ningún incentivo más que el propio aprendizaje.

El objetivo principal de la gamificación es conseguir la implicación voluntaria de las personas en las actividades propuestas, creando estructuras de refuerzos y la visualización de los procesos sumergidos, fomentando la actitud lúdica que favorezca conductas colaborativas y competitivas entre usuarios. ([Smith-Robbins, 2011](#))

Esta trata de conseguir que las tareas que normalmente son realizadas sin mucho entusiasmo, se conviertan en situaciones más agradables y significativas, impulsando que el usuario llegue al “estado de ánimo que corresponde al juego (...) el arrebató y entusiasmo”. ([Tardón, 2014](#))

## 1.4. Metodología de desarrollo

Se mencionarán las metodologías más destacadas con una breve reseña de cada una:

- RUP (Rational Unified Process): es un proceso de desarrollo de software. Es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo el Proceso Unificado es más que un proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes organizaciones y proyectos de gran tamaño.([Jacobson, Booch, & Rumbaugh, 2000](#)). Tiene como desventaja que los miembros de su equipo deben ser expertos en su campo para desarrollar un software bajo esta metodología, el proceso de desarrollo es demasiado complejo y desorganizado, en la reducción de los proyectos de vanguardia que utilizan las nuevas tecnologías, la reutilización de componentes no será posible, la integración en el proceso de desarrollo de software, en teoría parece una buena cosa. Pero en particular los grandes proyectos de desarrollo con flujos múltiples que sólo servirá para aumentar la confusión y causar más problemas durante las etapas de la prueba([Molina, 2011](#))
- XP (extreme programming): es una metodología de desarrollo que busca la simplicidad y ligereza en desarrollo de un proyecto de software; por esta razón esta metodología es apropiada para proyectos en los que durante su desarrollo pueden cambiar las necesidades del cliente y así mismo es necesario actualizar todo el proceso de ingeniería desarrollado; esta metodología busca adaptarse rápidamente al cambio disminuyendo todo lo posible los costes de: tiempo, talento humano, equipos, etc. En esta metodología cambia un poco a la metodología RUP, la implementación de los casos de uso por las: Historias de usuario, las cuales básicamente son la descripción de las necesidades del cliente resaltando sus necesidades y los riesgos que estos generan. En base a estas historias se definen las tareas a desarrollar, las cuales se planifican en un tiempo por la prioridad que requiere en el proyecto, estas tareas son asignadas al grupo de desarrollo del proyecto. ([Molina, 2011](#)). Sus desventajas son que es factible en proyectos a corto plazo, el desarrollo de software es riesgoso y difícil de controlar, se rediseña todo el tiempo, dejando el código en el estado más simple e incrementándose el tiempo del desarrollo, delimita el alcance del proyecto al cliente, la falta de documentación que no se hace , debido a que todos saben del proyecto, no permite recoger la experiencia para próximos proyectos.([Arteaga, 2014](#))
- Scrum: Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden

emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar. El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. Las reglas de Scrum se describen en el presente documento. Las estrategias específicas para usar el marco de trabajo Scrum son diversas y están descritas en otros lugares (Schwaber & Sutherland, 2013). Sus desventajas son que, si no se define una fecha, los clientes pueden pedir nuevas funcionalidades, si una tarea no está bien definida puede incrementar costes y tiempos, solo funciona bien en equipos pequeños y ágiles.

En una investigación realizada por José Gaspar (Arteaga, 2014) donde comparó varias metodologías entre ellas las mencionadas anteriormente dio como resultado la siguiente gráfica:



Ilustración 1 Resultado final de la comparativa entre metodologías. Fuente: (Arteaga, 2014)

Teniendo en cuenta el resultado de la investigación se decidió utilizar una adaptación de SCRUM para el desarrollo de juegos llamada SUM.

### Metodología utilizada

La metodología SUM para videojuegos tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo. SUM fue concebida para que se adapte a equipos

multidisciplinarios pequeños (de tres a siete integrantes que trabajan en un mismo lugar físico o estén distribuidos), y para proyectos cortos (menores a un año de duración) con alto grado de participación del cliente. (Acerenza et al., 2009)

SUM adapta para videojuegos la estructura y roles de Scrum. Se utiliza esta metodología ya que brinda flexibilidad para definir el ciclo de vida y puede ser combinado fácilmente con otras metodologías para adaptarse a distintas realidades.

### **Estructura o ciclo de vida**

- **Concepto:** Se define el tipo de videojuego que se realizará, determinando a grandes rasgos las características que este tendrá, la Jugabilidad, la historia, ambientación, etc. Se toma principal atención a puntos como la Jugabilidad, ya que esto permitirá crear las mecánicas básicas del videojuego y tener un rumbo de programación para el mismo. Implica definir aspectos de negocio (público objetivo, modelo de negocio), de elementos de juego (principales características, Jugabilidad, personajes e historia entre otros) y técnicos (lenguajes y herramientas para el desarrollo). El concepto del videojuego se construye a partir de ideas y propuestas de cada rol involucrado sobre los aspectos a definir. Las propuestas se refinan a través de reuniones y se analiza su factibilidad con pruebas de concepto. Esta fase finaliza cuando se tiene el concepto validado entre todas las partes involucradas.
- **Planificación:** La fase tiene como objetivo principal planificar las restantes fases del proyecto. Para ello es necesario definir el cronograma del proyecto junto con sus principales hitos, conformar el equipo para la fase de elaboración de acuerdo a las necesidades técnicas del proyecto, determinar y tercerizar las tareas que el equipo no pueda cumplir, definir el presupuesto y especificar las características. Esto último consiste en describir, estimar y priorizar cada una de las características funcionales y no funcionales que definen el videojuego. Una característica funcional representa una funcionalidad del videojuego desde el punto de vista del usuario final, mientras que, una característica no funcional representa una propiedad o cualidad que el videojuego debe presentar. La planificación que se obtiene en esta fase es flexible ya que en cada iteración de la fase de elaboración se puede modificar para adaptarse a los cambios y reflejar la situación actual del proyecto.
- **Elaboración:** Este es el punto más importante de la metodología, ya que de este punto depende el éxito o fracaso del desarrollo de un videojuego. El objetivo de esta fase es implementar el videojuego. Para ello se trabaja en forma iterativa e incremental para lograr una versión ejecutable del videojuego al finalizar cada iteración. Estas se dividen en tres etapas, en la primera

se planifican los objetivos a cumplir, las métricas a utilizar en el seguimiento, las características a implementar y las tareas necesarias para ello. En la segunda se desarrollan las características planificadas a través de la ejecución de las tareas que la componen. Al mismo tiempo se realiza el seguimiento para mantener la visión y el control de la iteración en base a los objetivos planteados. La tercera y última implica la evaluación del estado del videojuego y de lo ocurrido en el transcurso de la iteración para actualizar el plan de proyecto respecto a la situación actual. Con esta forma de trabajo se puede evaluar el avance del proyecto, lo cual permite realizar cambios a tiempo y tomar decisiones para cumplir con los plazos planificados. Además, la experiencia adquirida permite mejorar la forma de trabajo en cada iteración y aumentar la productividad. En este punto se plantean objetivos para el avance del desarrollo y las tareas necesarias para lograr estos objetivos, se controla el avance en base al cronograma y a los objetivos planteados, además se evalúa el videojuego, y se corrigen errores y problemas que van surgiendo en base a la experiencia adquirida, además esto permite optimizar el trabajo.

- **Beta:** Es una versión de prueba del videojuego, con esta versión de prueba se puede testear el funcionamiento del videojuego, verificar los bugs existentes y corregirlos, además de verificar que la Jugabilidad y los eventos del videojuego funcionen correctamente. Otra característica de esta parte del desarrollo es que se puede publicar la beta para obtener una retroalimentación por parte de los usuarios que están interesados en el videojuego, con la versión Beta se analiza la diversión que genera el videojuego, el Jugabilidad, la curva de aprendizaje, la curva de dificultad entre otras cosas.
- **Cierre:** El cierre es el paso final de esta metodología, se llega a esta fase cuando se lanza la versión final del videojuego, durante esta fase se evalúan los problemas, éxitos, soluciones, cumplimiento de objetivos y se realiza una retroalimentación de todo el proceso de creación del videojuego. Esta fase tiene como objetivos entregar la versión final del videojuego al cliente según las formas establecidas y evaluar el desarrollo del proyecto.
- **Gestión de Riesgos:** Es un punto que se toma en cuenta durante todo el proceso de creación del videojuego, tiene como objetivo minimizar el impacto de problemas, establecer una probabilidad e impacto de posibles riesgos existentes, además, permite prever posibles problemas de programación y establecer métodos de solución para estos problemas. En el caso de la programación de un videojuego se debe tomar especial atención a los errores más comunes que se realizan al momento de desarrollar un videojuego, estos errores son los denominados “loopholes” (ventajas que el jugador puede tener en base a bugs, glitches, etc.) y los “Dead Ends” (Callejones sin salida que imposibilitan el avance en el videojuego). Esto se debe a que distintos

riesgos pueden ocurrir en cualquiera de las fases, por lo cual siempre debe existir un seguimiento de los mismos.

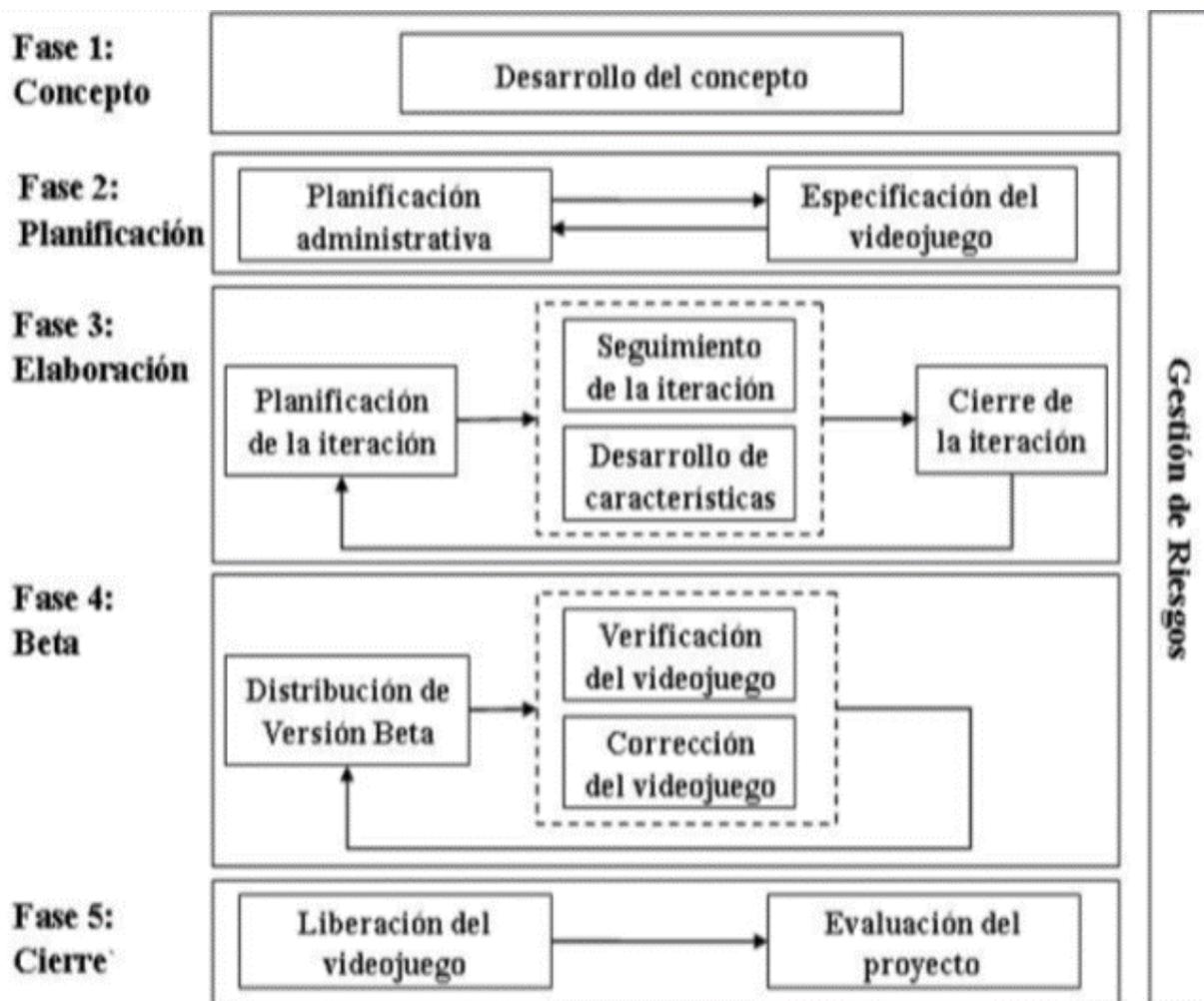


Ilustración 2 Fases del proceso. Fuente: (Acerenza et al., 2009)

En la Décima Séptima Conferencia Iberoamericana en Sistemas, Cibernética e Informática en el 2018 (MURILLO-SANCHEZ et al., 2018) el equipo de Xavier-Alexis Murillo - Sánchez, Andrés-Leonardo Gutiérrez-Rocha, Alan-Wilfor Ibáñez-Illanes, Jorge-Armando Quiroz-Pérez, Guillermo Sahonero-Alvarez, Fabio-Richard Diaz-Palacio modificaron la metodología adicionándole el Product Backlog en la etapa de planificación, priorizando de esta forma algunas tareas más importantes que otras en la etapa de elaboración. Asimismo, se implementaron las etapas de validación y presentación de resultados y se cambió la etapa de Beta por Alfa, esto debido a que la Alfa es un desarrollo más pequeño, lo que permitía tener mayor retroalimentación del desarrollo y generar documentación en base a los resultados

obtenidos. Esto permite modificar sin retroceder sustancialmente en el desarrollo del videojuego en caso de tener una mala recepción.

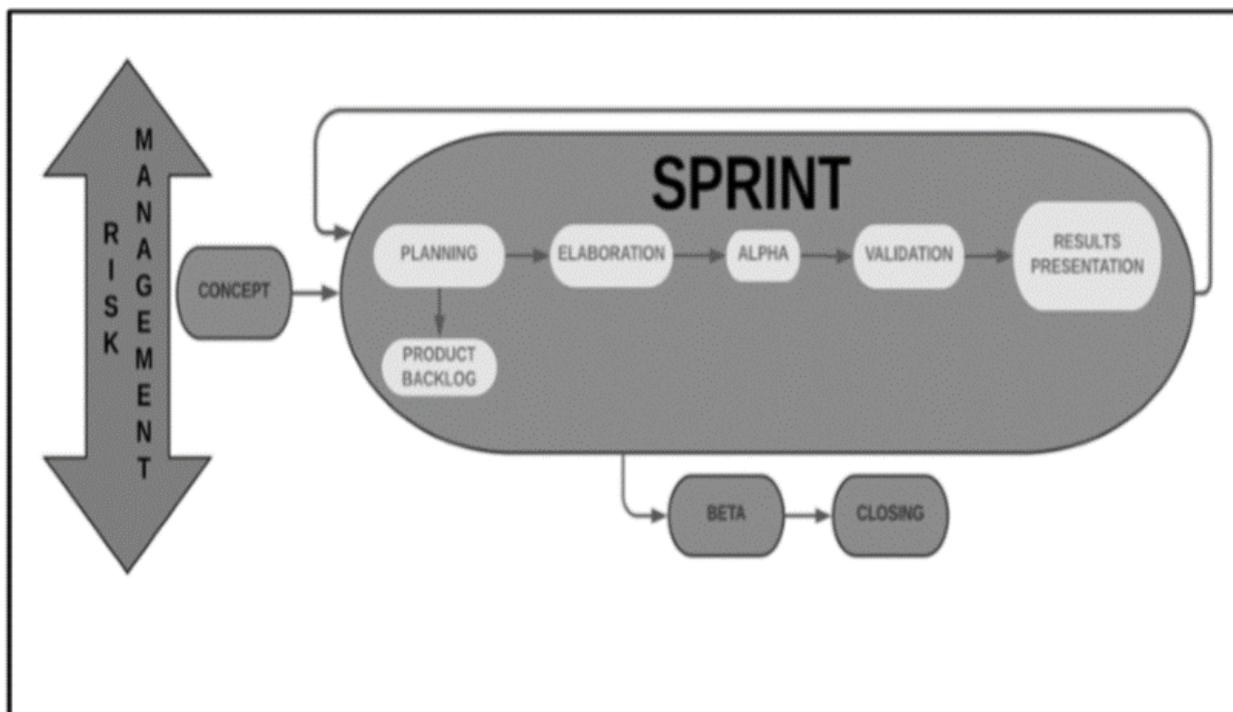


Ilustración 3 Diagrama de la metodología SUM modificada. Fuente:(MURILLO-SANCHEZ et al., 2018)

## Roles

La metodología define cuatro roles: equipo de desarrollo, productor interno, cliente y verificador beta. El productor interno y el cliente se corresponden en forma directa con los roles de Scrum Master y Product Owner de Scrum respectivamente. El equipo de desarrollo tiene las características del Scrum team, pero a diferencia de Scrum se definen subroles dentro del equipo. Estos se corresponden con los que se utilizan habitualmente en la industria local y son los de programador, artista gráfico, artista sonoro y diseñador de juego. Es necesario esta definición ya que se requiere una alta especialización para satisfacer las distintas disciplinas que involucra del desarrollo de videojuegos, aspecto no contemplado en Scrum. El rol de verificador beta no está presente en Scrum, pero sí se detecta su existencia en el relevamiento de la realidad local y en la industria del videojuego en general. Su responsabilidad es la de realizar la verificación funcional del videojuego y comunicar su resultado.

## 1.5. Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. Constituyen un soporte automatizado para el desarrollo y mantenimiento del software. Se pueden ver como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.

### Visual Paradigm

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones que utiliza modelado UML. Es ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. ([V. P. Corporation, 2019](#))

Ventajas del Visual Paradigm:

- Navegación intuitiva entre la escritura del código y su visualización.
- Potente generador de informes en formato PDF/HTML.
- Documentación automática Ad-hoc.
- Ambiente visualmente superior de modelado.
- Sofisticado diagramador automático.
- Sincronización de código fuente en tiempo real

## 1.6. Herramientas y tecnologías

El desarrollo de videojuegos es una tarea compleja y costosa, llegándose a invertir años y millones de dólares en su desarrollo, y esto ocurre también, aunque en menor medida, en el desarrollo de videojuegos para dispositivos móviles inteligentes. Estos costes y tiempos de desarrollo se ven aumentados por la existencia de varias plataformas distintas dentro de los “smartphones”, siendo las principales iOS y Android. Las aplicaciones para la plataforma iOS se escriben en el lenguaje de programación Objective-C, mientras que las aplicaciones para Android se programan mediante Java, aunque también permite el uso de otros lenguajes como C/C++, JavaScript. Además, existen cada vez más modelos de dispositivos, con distintos tamaños de pantalla, resoluciones, CPUs y GPUs y memoria

RAM, lo que dificulta comprobar el correcto funcionamiento de la aplicación en todos los teléfonos móviles.

Para intentar solventar este problema han surgido varios SDKs (Software Development Kit, o kit de desarrollo de software) y motores de videojuegos multiplataforma, permitiendo reutilizar la mayor parte del código al desarrollar para las distintas plataformas que soportan, algunos ejemplos de este tipo de SDK son Corona SDK, Marmalade, GameSalad o Android SDK. Un motor de juego cuya popularidad se ha ido incrementando rápidamente es Unity, habiendo sido usado para juego como Bad Piggies de Rovio o Wasteland 2 de inXile Entertainment.

La tabla a continuación muestra una comparativa entre Unity y QtQuick

	Unity	QtQuick
<b>Espacio ocupado por los proyectos</b>	Los proyectos se expanden enseguida en tamaño, especialmente si se usan assets complejos o modelos 3D de alta resolución, por no hablar de sonido de alta calidad.	Los proyectos desarrollados no demandan gran cantidad de tamaño ya que este no utiliza gráficos complejos
<b>Rendimiento</b>	Unity 3D es una herramienta compleja, que simplemente estando en modo edición tiene varias tareas activas, el desarrollo utilizando este motor de juego requiere de un hardware con altas prestaciones	QtQuick nos permite con su IDE QtCreator crear proyectos en máquinas con un hardware de gama baja y media.
<b>Gestión de las versiones</b>	Cuando se pasó de la versión 4 a la 5 de Unity hubo muchos problemas. Muchos elementos e instrucciones quedaron como obsoletos, y a veces, si un juego es complejo, reconvertirlo es tedioso. Esto es algo que pasa con todas las herramientas, pero en Unity 3D, al tener programas complejos, lo es más. Además, si un asset importante de un tercero que usamos ya no tiene mantenimiento, tendremos que readaptarlo nosotros, o cambiarlo por otro que haga una	Al pasar de una versión a otra los componentes creados en las anteriores versiones se pueden incorporar sin problemas haciendo el código y los componentes reusables en otros proyectos.

	funcionalidad similar. Otro ejemplo de problema se ha dado con la versión 5.3, que ha dejado obsoletas más instrucciones y la gestión de niveles, entre otras cosas. Muchas personas deciden quedarse en versiones anteriores, pero claro, luego dejan de mantenerse y tendremos más problemas.	
<b>Multiplataforma</b>	Si	Si
<b>Política de precios y costes</b>	Entramos en un tema subjetivo, pero que crea mucha polémica. Este producto tiene una versión gratuita que era bastante limitada, pero que ahora permite, al menos, poner el logo de cada uno. Luego hay una versión profesional y otra para empresas.. Este es un software propietario	Este es totalmente libre y de código abierto

*Tabla 2 Comparativa entre Unity y QtQuick*

Teniendo en cuenta el hardware de desarrollo, y que su IDE por preferencia es QtCreator el cual no requiere licencias, el equipo de desarrollo decidió utilizar como plataforma de desarrollo QtQuick.

### **Plataforma de desarrollo utilizada**

QtQuick es una aplicación multiplataforma y marco de interfaz de usuario para desarrolladores que utilizan C++, QML, CSS y JavaScript como lenguaje. QtCreator es el entorno de desarrollo que lo soporta. Además, es una colección de tecnologías diseñadas para ayudar a los desarrolladores a crear interfaces de usuario modernas, fluidas e intuitivas que son usadas en teléfonos móviles, tabletas y otros dispositivos portables. QtQuick consta de un completo conjunto de elementos para interfaz de usuario y un lenguaje declarativo para describir las escenas e interfaces. Además, cuenta con una colección de APIs de C++ utilizadas para integrar funciones de alto nivel. Sus herramientas de soporte se desarrollan como un proyecto de código abierto regido por un modelo meritocrático. Entre sus ventajas cuenta con componentes que permiten ser fácilmente estilizados, obteniendo flexibilidad visual. QtQuick se puede usar bajo el código abierto (LGPL v2.1) o términos comerciales. [\(JRyannel, 2015\)](#)

## Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el funcionamiento y el comportamiento físico y lógico de la máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

### QML

El módulo de QTQML proporciona un framework para el desarrollo de aplicaciones y bibliotecas con el lenguaje QML. Además, define y permite a los desarrolladores de aplicaciones ampliar el lenguaje QML con tipos de datos personalizados e integrar código QML con JavaScript y C++. Se debe tener en cuenta que el módulo de QTQML proporciona solo el lenguaje y la infraestructura para QML, y no un lienzo visual o motor de renderizado para la creación de interfaces de usuario. Estos se brindan por el módulo QTQuick, que proporciona muchos componentes visuales, un framework para animación y mucho más para la construcción de interfaces de usuario con QML.

El lenguaje QML se usa principalmente para aplicaciones móviles, donde la entrada táctil, las animaciones y una buena experiencia de usuario son cruciales. QML es una versión adaptada para QT de una tendencia hacia lenguajes declarativos en las interfaces de usuario. La gran diferencia en cuanto tecnologías anteriores es el uso de JSON en vez de XML ([JRYannel, 2015](#))

### Ventajas

- Velocidad de desarrollo
- Impulsa la separación entre la presentación y la lógica
- Una mejor integración entre los programadores y diseñadores
- Amigable a diseñadores CSS/JavaScript
- Cambios a interfaz de usuario no requieren re compilación

### C++

Es un lenguaje de programación muy utilizado mundialmente en la actualidad, surge en la década de los ochenta por parte de Bjarne Stroustrup, con el objetivo de mejorar el lenguaje de programación anteriormente publicado, el Lenguaje C, este es un lenguaje híbrido desde el punto de vista de los lenguajes orientados a objetos. Una de las mejores particularidades del lenguaje C++ es la de poder redefinir sus operadores. Su nombre viene a darse en 1983 por Rick Mascitti, el cual significa (C++) "incremento de C", refiriendo a este como una actualización al lenguaje C.

Algunas de sus características distintivas clave incluyen:

- Muy claro, sintaxis legible.
- La orientación a objetos intuitiva.
- Expresión natural del código de procedimiento.
- Extensas bibliotecas estándar y módulos de terceros para la mayoría de las tareas.
- Extensiones y módulos fácilmente escritos en C, C + " (o Java para Jython, o. NET

para IronPython).

- Integrable dentro de las aplicaciones como una interfaz de scripting.
- Poderoso y rápido.
- Multiplataforma (Linux | Unix, Windows, OS | 2, Mac, entre otros).
- Fácil aprendizaje.
- Libre uso y distribuibles, incluso para uso comercial (23).

## JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas webs dinámicas. Una página web dinámica es aquella que incorpora efectos como animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. En la actualidad es uno de los lenguajes de programación más usados debido a sus ventajas entre las cuales están: técnicamente, JavaScript es un lenguaje de programación interpretada, por lo que no es necesario compilar los programas para ejecutarlos, en otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios por lo que el tiempo de ejecución estará cerca de una aplicación creada en C o C++; cuenta con JSON (JavaScript object Notation) que se utiliza en casi todas las aplicaciones web modernas, tanto para la comunicación, como para la persistencia de los datos; soporta características como la encapsulación, polimorfismo, la herencia múltiple y la composición; su sintaxis es inmediatamente familiar a cualquiera que tenga experiencia en C, C++, Java, PHP entre otros; casi todo en JavaScript es un objeto, incluyendo las funciones, debido a esa característica, las funciones se pueden utilizar en casi cualquier lugar y además el código JavaScript, en virtud de su ubicuidad, es el código reutilizable más portátil alrededor del mundo. ([JRyannel, 2015](#); [Lindley, 2012](#))

## **Ambiente de desarrollo integrado (IDE)**

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

### **QTCreator**

Es un IDE creado por Trolltech para el desarrollo de aplicaciones con bibliotecas Qt. Esta distribuido bajo los términos de GNU Lesser General Public License, entre otras. Es un software libre y de código abierto.

QTCreator incluye una amplia gama de herramientas útiles entre las, cuales se encuentran:

- SmartCode Editor: es un editor de código que ofrece resaltado de sintaxis, así como de código.
- Qt Project Generating Wizard: este asistente permite al usuario generar un proyecto para una aplicación de consola, una aplicación con interfaz de usuario o una biblioteca de C++, entre otros.
- Qt Help Integration: se puede acceder fácilmente a toda la documentación haciendo clic en el botón de ayuda dentro del mismo IDE
- QT Designer Integration: la forma de la interfaz de usuario y escenas se puede diseñar fácilmente. Simplemente se hace clic sobre un archivo ".ui" en el explorador para poner en marcha la integración.
- Locator: una potente herramienta de navegación que permite al usuario localizar los archivos y el uso mínimo de clases de teclado.
- Support for qmake's .pro file format: el archivo. Pro se usa como descripción del proyecto

### **Herramientas para la edición de imágenes**

**Photoshop:** Programa informático de edición de imágenes. Su nombre completo es Adobe Photoshop y está desarrollado por la empresa Adobe Systems Incorporated. Funciona en los sistemas operativos Apple Macintosh y Windows. Permite modificar imágenes digitalizadas, especialmente fotografías. También se utiliza para crear y editar imágenes (por ejemplo, logotipos) y gráficos. La forma, la luz, el color y el fondo son algunos de los aspectos que esta herramienta permite editar. El nombre comercial de este producto se puede traducir del inglés como 'taller de fotos'. ([Significados.com](http://Significados.com), 2019)

## Conclusiones del capítulo

Para concluir, los videojuegos serios y la gamificación han desmembrado toda la potencia de transformación social que se intuía en los videojuegos de entretenimiento, como se pudo observar en el este capítulo, y se han focalizado cada uno en aspectos distintos de dicha potencialidad. En tanto que los videojuegos serios se han convertido en una eficaz herramienta para enviar mensajes de tipo social, la gamificación ha mostrado su potencial para la transformación de procesos y percepciones.

Se realizó una descripción de las tecnologías escogidas para la implementación de la solución, las cuales son: como plataforma de desarrollo QtQuick.

Como lenguajes de programación se escogió C++ QML y JavaScript para el trabajo con QTQuick, así como el IDE de desarrollo QtCreator debido a las facilidades que brinda para los lenguajes escogidos.

Como herramienta de modelado se seleccionó Visual Paradigm, ya que se caracteriza por ser muy intuitiva.

Como metodología de desarrollo se seleccionó SUM por ser una metodología ágil, que propicia varias ventajas para proyectos pequeños.

# Capítulo II Descripción de la solución

En el presente capítulo se interpretan las necesidades del sistema especificándolas mediante las características funcionales y no funcionales. Además, se detallan brevemente los artefactos de la metodología de desarrollo SUM, seleccionada para guiar el desarrollo del proyecto, según cada etapa de esta.

## 2.1. Propuesta del sistema

La solución que se propone abarca la creación de una aplicación para dispositivos móviles para lograr en los estudiantes de nuevo ingreso a la universidad, o en el jugador, el desarrollo de las habilidades de lógica mencionadas en el capítulo anterior. Este se ejecutará sobre dispositivos móviles con el sistema operativo Android, con posibilidades de extender su compatibilidad a IOS y a Windows

En dicha aplicación se unificarán conceptos de juegos como Lightbot y Human Resource Machine en donde la lógica de programación guía el juego, pero adicionándole también problemas donde otros tipos de lógica están involucrados como la lógica modal, lógica de clases, lógica proposicional, lógica simbólica que en la carrera de Ingeniería Informática especialmente en el primer año son fundamentales para asignaturas como Introducción a la programación, Matemática discreta e Inteligencia Artificial.

Los estudiantes pueden desarrollar su capacidad de encontrar diferentes maneras de resolver los problemas que encuentran a través del juego. A menudo, los jugadores podrán encontrar que requieren unas habilidades específicas más adelante en el juego, y por lo tanto están obligados a mantener y perfeccionar sus habilidades para su uso posterior.

## 2.2. Etapa 1 Concepto

En esta etapa se definen los principales aspectos del juego como visión, genero, ambientación, entre otros. También se definen las plataformas, tecnologías y herramientas para su desarrollo. Los aspectos de negocio posibles y a que publico apunta.

## **Tarea 1 Definir aspectos del juego**

**Visión:** El juego tendrá como objetivo incrementar las habilidades de lógica en los estudiantes mediante acertijos o puzzles. El personaje para poder escapar de la mazmorra deberá poder adquirir cierta cantidad de llaves distribuidas por los distintos niveles de esta, para poder acceder a dichas llaves los jugadores deberán abrir distintas puertas protegidas con puzzles, acertijos y problemas de lógica, y luego con estas llaves abrir las puertas de cada nivel hasta llegar a la salida.

**Género:** Es un juego con mezcla entre puzzle y aventura. Este será un juego educativo.

**Jugabilidad:** El personaje será libre de escoger el camino para llegar a la puerta de salida de la mazmorra, pero para llegar a esta tendrá que resolver los problemas, acertijos y puzzles que presentan cada puerta durante su recorrido. Algunas de las puertas tendrán acceso a otros niveles y otras a espacios donde se guardan las llaves y otras ventajas.

**Características:** El juego permitirá el desarrollo de habilidades de lógica mediante los acertijos, puzzles y problemas, de forma más divertida. Se desarrollará para dispositivos móviles por lo cual se tendrá más fácil acceso a este en cualquier momento o lugar. El juego dispondrá de diferentes rutas para llegar a la puerta de salida. El grado de dificultad se incrementará a medida que el personaje se acerque a la puerta de salida.

**Ambientación:** El juego se desarrollará en una mazmorra con diferentes niveles.

**Público al que apunta:** Jóvenes de enseñanza superior

## **Tarea 2: Definir aspectos técnicos**

**Plataforma objetivo:** Dispositivos móviles con sistema operativo Android.

**Tecnología y herramientas:** El juego se desarrollará utilizando lenguajes como QML, JavaScript y C++ haciendo uso del entorno de desarrollo QtCreator y herramientas para el diseño y edición de imágenes como Photoshop. También se utilizará Visual Paradigm y Sprintometer

## **Tarea 3: Definir aspectos del negocio.**

**Audiencia objetivo:** Estudiantes de carreras universitarias donde las habilidades de lógica sean fundamentales en especial estudiantes de nuevo ingreso a las mismas.

**Modelos de negocio:** Este juego no generara ninguna ganancia monetaria.

## 2.3. Etapa 2 Planificación

En esta etapa se definen los integrantes del equipo de para el resto del proyecto. Determinar cuántas iteraciones se realizarán y especificar los hilos que se deben cumplir. Calcular el presupuesto del proyecto. Especificar las características y realizar una estimación de tiempo del proyecto.

### Tarea 2.1 Definir Roles

En esta tarea se definen los roles de los integrantes del equipo según la metodología seleccionada. Para ello se elaboró la siguiente tabla.

Roles	Encargado
<b>Diseñador</b>	Osniel O. Domínguez
<b>Programador</b>	Osniel O. Domínguez
<b>Artista gráfico</b>	Osniel O. Domínguez
<b>Artista sonoro</b>	Osniel O. Domínguez
<b>Productor interno</b>	Osniel O. Domínguez
<b>Cliente</b>	Walfredo Gonzales
<b>Verificador Beta</b>	José Enrique Diaz

*Tabla 3 Roles del proyecto. Fuente: Elaboración propia*

### Tarea 2.2 Creación del cronograma

En esta tarea se define el cronograma que debe seguir el equipo según la metodología seleccionada. Para ello se elaboró la siguiente tabla.

Fase o tarea	Asignada a	Inicio	No. Días
Fase 1 Concepto		mes/día/año	
<b>Análisis previo</b>	Osniel O. Domínguez	10/01/18	7
<b>Definir aspectos de negocio</b>	Osniel O. Domínguez	10/10/18	3
<b>Definir aspectos del juego</b>	Osniel O. Domínguez	10/13/18	3
<b>Definir aspectos técnicos</b>	Osniel O. Domínguez	10/16/18	3
<b>Reunión con el cliente</b>	Osniel O. Domínguez	10/19/18	1
<b>Realizar cambios planteados en la reunión</b>	Osniel O. Domínguez	10/20/18	5
Fase 2 Planificación			
<b>Definir el cronograma del proyecto</b>	Osniel O. Domínguez	11/01/18	10
<b>Realizar la especificación de características</b>	Osniel O. Domínguez	11/11/18	5
<b>Elaborar Diseño de la base de datos</b>	Osniel O. Domínguez	11/16/18	5
<b>Reunión con el cliente</b>	Osniel O. Domínguez	11/21/18	1
<b>Realizar cambios planteados en la reunión</b>	Osniel O. Domínguez	11/22/18	7

Fase 3 Elaboración				
<b>Iteración 1</b>	Osniel Domínguez	O.	01/07/19	30
<b>Reunión con el cliente</b>	Osniel Domínguez	O.	02/06/19	2
<b>Realizar cambios planteados en la reunión</b>	Osniel Domínguez	O.	02/08/19	4
<b>Primera entrega</b>	Osniel Domínguez	O.	02/12/19	7
<b>Planificación de la próxima iteración</b>	Osniel Domínguez	O.	02/12/19	1
<b>Iteración 2</b>	Osniel Domínguez	O.	02/13/19	21
<b>Reunión con el cliente</b>	Osniel Domínguez	O.	03/06/19	2
<b>Realizar cambios planteados en la reunión</b>	Osniel Domínguez	O.	03/08/19	4
<b>Segunda entrega</b>	Osniel Domínguez	O.	03/12/19	1
<b>Planificación de la próxima iteración</b>	Osniel Domínguez	O.	03/13/19	1
<b>Iteración 3</b>	Osniel Domínguez	O.	03/14/19	21
<b>Reunión con el cliente</b>	Osniel Domínguez	O.	04/04/19	2

<b>Realizar cambios planteados en la reunión</b>	Osniel Domínguez	O.	04/06/19	3
<b>Tercera entrega</b>	Osniel Domínguez	O.	04/09/19	1
<b>Planificación de la próxima iteración</b>	Osniel Domínguez	O.	04/09/19	1
<b>Iteración 4</b>	Osniel Domínguez	O.	04/10/19	26
<b>Reunión con el cliente</b>	Osniel Domínguez	O.	05/06/19	2
<b>Realizar cambios planteados en la reunión</b>	Osniel Domínguez	O.	05/08/19	6
<b>Cuarta entrega</b>	Osniel Domínguez	O.	05/14/19	7
<b>Planificación de la próxima iteración</b>	Osniel Domínguez	O.	05/14/19	1
<b>Iteración 5</b>	Osniel Domínguez	O.	05/15/19	10
<b>Reunión con el cliente</b>	Osniel Domínguez	O.	05/25/19	2
<b>Realizar cambios planteados en la reunión</b>	Osniel Domínguez	O.	05/27/19	6
Fase 4 Beta				
<b>Verificación del videojuego</b>	Osniel Domínguez	O.	05/14/19	10

<b>Corrección del videojuego</b>	Osniel Domínguez	O.	05/24/19	15
Fase 5 Cierre				
<b>Presentación del software</b>	Osniel Domínguez	O.	06/04/19	3

Tabla 4 Cronograma del proyecto. Fuente: Elaboración propia

Dentro de cada iteración se definieron varias tareas, representándolas en la descripción de las iteraciones, homologa a los sprint backlog de scrum.

No. Iteración	Tarea	Asignada a
<b>1</b>	Elaborar Diseño de la base de datos	Programador
	Diseño y edición de imágenes para los componentes del mapa	Diseñador/ Artista Grafico
	Creación de los componentes del mapa	Programador
	Diseño de los mapas	Diseñador
	Creación de los mapas	Programador
	Pruebas al mapa	Verificador Beta
	Diseño y edición de imágenes para el Sprite del personaje	Diseñador/ Artista Grafico
<b>2</b>	Programar movimientos del personaje	Programador
	Crear animaciones al movimiento del personaje	Programador/ Artista Grafico

	Probar movimientos de personaje	Verificador Beta
	Diseño y edición de imágenes para la página principal	Diseñador/ Artista Grafico
	Diseño de la página principal	Diseñador
	Creación de la página principal	Programador
<b>3</b>	Diseño y edición de imágenes para el menú	Diseñador/ Artista Grafico
	Diseñar menú	Diseñador
	Crear menú	Programador
	Probar menú	Verificador Beta
	Diseño y edición de imágenes para los problemas de texto	Diseñador/ Artista Grafico
	Diseño de los problemas de texto	Diseñador
	Creación de los problemas de texto	Programador
<b>4</b>	Diseño y edición de imágenes para los problemas de Algoritmización	Diseñador/ Artista Grafico
	Diseño de los problemas de Algoritmización	Programador
	Creación de los problemas de Algoritmización	Programador
	Diseño y edición de imágenes para los problemas variados (Torres de Hanoi, Cruzando, ...)	Diseñador/ Artista Grafico
	Diseño de los problemas variados (Torres de Hanoi, Cruzando, ...)	Diseñador

	Creación de los problemas variados (Torres de Hanoi, Cruzando, ...)	Programador
5	Diseño y edición de imágenes para la puerta cifrada y cofres	Diseñador/ Artista Grafico
	Diseño de la puerta cifrada y cofres	Diseñador
	Creación de la puerta cifrada y cofres	Programador
	Creación de las funciones de guardar	Programador
	Creación de las funciones de cargar	Programador
	Probar funciones de cargado y guardado	Verificador Beta

Tabla 5 Descripción de iteraciones. Fuente: Elaboración propia

### Tarea 2.3 Creación de la especificación del producto

En esta tarea se define la especificación del producto según la metodología seleccionada. Donde se definen las características funcionales y no funcionales del videojuego tomando como base el concepto. La descripción de cada característica es breve, pero contiene suficiente detalle como para poder estimar su prioridad Para ello se elaboró las siguientes tablas.

Listado de Características	Descripción
<b>Mapas</b>	Los mapas a medida que el jugador abra la puerta de salida deben ir hacia otro mapa. Dentro de cada mapa la organización de los componentes debe permitirle al jugador moverse por este.
<b>Movimiento</b>	Es el movimiento del jugador dentro de cada mapa
<b>Problemas</b>	Son los problemas, acertijos, puzles o minijuegos para abrir las puertas o lograr algún objetivo

<b>Cargar</b>	El menú principal debe dar opciones para empezar una nueva partida o continuar la anterior
<b>Guardar</b>	El juego debe permitirle al jugador guardar el progreso
<b>Base de Datos</b>	La base de datos debe ser local y no requerir demasiado consumo de memoria
<b>Objetos interactivos</b>	Estos objetos serían los cofres, las pistas y las llaves que se encuentran en el mundo

Tabla 6 Especificación del producto. Características funcionales. Fuente: Elaboración propia

Características no funcionales	Descripción
<b>Rendimiento</b>	La aplicación deberá estar optimizada para dispositivos móviles de gama media y baja
<b>Diseño</b>	El diseño de la aplicación tiene que primar los tiempos de respuesta sobre los recursos de video.
<b>software</b>	La aplicación debe ser optimizada para todos los dispositivos con sistema operativo Android en especial los de gama baja o media
<b>Portabilidad</b>	La aplicación debe ser portable

Tabla 7 Especificación del producto. Características no funcionales. Fuente: Elaboración propia

#### Tarea 2.4 Priorización y estado del producto

En esta tarea se define la prioridad y el estado de cada característica de la especificación del producto determinando así el nivel de importancia. Para ello se elaboró la siguiente tabla.

Prioridad	% Realizado	Id	Característica	Responsable
3	0	1	Mapas	Osniel O. Domínguez
2	0	2	Movimiento	Osniel O. Domínguez
5	0	3	Problemas	Osniel O. Domínguez
3	0	4	Cargar	Osniel O. Domínguez
3	0	5	Guardar	Osniel O. Domínguez
2	0	6	Base de Datos	Osniel O. Domínguez
3	0	7	Objetos interactivos	Osniel O. Domínguez

Tabla 8 Priorización y estado del producto. Fuente: Elaboración propia

### Tarea 2.5 Definir presupuesto y costos

Durante la etapa de planificación de la metodología de desarrollo de software SUM fue necesario estimar el tiempo de implementación del proyecto, por lo que se obtiene al comenzar, una estimación inicial de su duración. Esta se realizó basándose en las características funcionales resultantes de las sucesivas entrevistas con el cliente, la experiencia del programador en el trabajo con el lenguaje de programación, el entorno de desarrollo, el conocimiento sobre el tema de investigación y las técnicas de programación necesarias para resolver el problema.

Teniendo en cuenta los meses de desarrollo del proyecto y los meses de investigación y estudio, el proceso de desarrollo abarca un tiempo estimado de 9 meses. Considerando que los encuentros con los tutores y con el cliente fueron de 30 días en 9 meses, el tiempo de duración es de 1 mes. A partir de los datos anteriores se calculó el valor cuantitativo del costo del proyecto utilizando la fórmula de Bohem:

$$\text{Costo} = \text{CH} * \text{SM} * \text{TD}$$

Donde CH: Cantidad de hombres. SM: Salario mensual por persona. TD: Tiempo de desarrollo total estimado

La siguiente muestra el salario mensual de cada miembro del equipo de trabajo, considerando 5 meses de duración del proyecto para el equipo de desarrollo y 1 mes para el resto del equipo de trabajo, el costo inicial obtenido es de \$2700.00cup

Miembros	Salario mensual(cup)	Tiempo de desarrollo(meses)	Costo
<b>Osniel O. Domínguez Pérez</b>	\$100.00	9	\$900.00
<b>Dr. Walfredo Gonzales</b>	\$1000.00	1	\$1000.00
<b>Ing. José Enrique Diaz Ramos</b>	\$800.00	1	\$800.00
<b>Total</b>			2700.00

Tabla 9 Equipo de trabajo y salario. Fuente: Elaboración propia

### Costo real

Para calcular el costo real del proyecto, se tuvo en cuenta determinados gastos realizados durante su desarrollo. La tabla siguiente muestra una relación aproximada entre aspectos que influyeron durante los 9 meses de trabajo y el gasto que ocasionaron.

Aspecto	Gasto mensual	Gasto total
Compra de tarjetas de internet nauta	\$50.00	\$450
Costo del móvil	\$50.00	\$450.00
Transporte	\$50.00	\$450.00
Gastos de papel	\$12.00	\$108.00
Tarifa eléctrica	\$30.00	\$270.00
<b>Total</b>		\$1728.00

Tabla 10 Gastos. Fuente: elaboración propia

Al adicionarle estos gastos al costo inicial se estima que el costo monetario real de LogiLab es de aproximadamente \$4428.00. CUP

## **2.4. Diseño de la solución propuesta**

Se diseñará una solución que se ejecute sobre un dispositivo móvil, que permita al usuario ejercitar sus habilidades de lógicas mencionadas en el capítulo 1. Enlazando esta con una base de datos local donde se guardarán todos los datos del juego.

### **Arquitectura**

La arquitectura es la organización fundamental de un sistema materializándose en sus componentes, las relaciones entre ellos, el ambiente y los principios guiando su diseño y evolución (Maie, Emery, & Hilliard). En la aplicación se identifica la necesidad de utilización del patrón arquitectónicos Modelo-Vista. A continuación, se explicará con más detalle, donde se reflejan en el sistema desarrollado.

### **Modelo/vista**

Modelo-vista-controlador es un patrón de diseño que se originó de Smalltalk y que es usado comúnmente al construir interfaces de usuario. Si la vista y el controlador son combinados el resultado es la arquitectura modelo/vista. Esta también separa la forma en que la información es almacenada de la forma en que se presenta al usuario, pero provee un marco de trabajo más simple basado en los mismos principios. Esta separación permite presentar la misma información en diferentes vistas e implementar nuevos tipos de vistas sin cambiar las estructuras subyacentes. El modelo se comunica con la fuente de datos, brindando una interfaz para los otros componentes de la arquitectura. La naturaleza de la comunicación depende del tipo de fuente de datos y de la forma en que es implementado el modelo. La vista obtiene Índices del modelo, estos se refieren a objetos de datos. Al proveer estos índices al modelo la vista puede obtener la información de la fuente. En una vista normal, un delegado edita el objeto de datos, cuando este es editado el delegado se comunica directamente con el modelo usando los índices del modelo (DIGIA). De manera general esta arquitectura se representa de la siguiente forma:

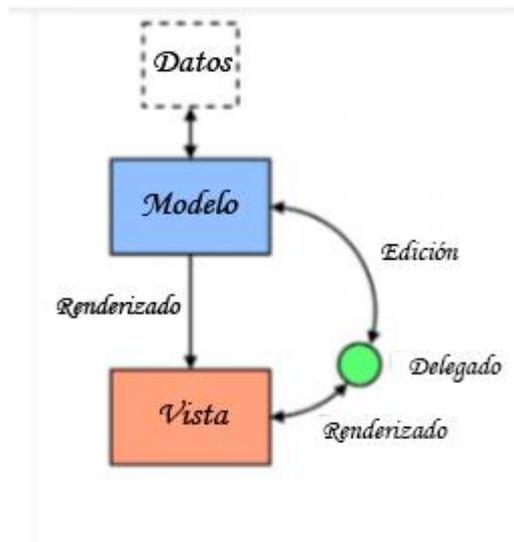


Ilustración 4 : Patrón Modelo-Vista. Fuente: qt-project.org

## Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).

## Patrones GRASP

GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para sugerir la importancia de aprehender (grasping en inglés) estos principios para diseñar con éxito el software orientado a objetos. Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Es importante entender y ser capaces de aplicar estos principios durante la creación de los diagramas de interacción porque un desarrollador de software con poca experiencia en la tecnología de objetos necesita dominar estos principios tan rápido como sea posible; constituyen la base de cómo se diseñará el sistema

## Abstract Factory

Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas. El patrón de diseño Abstract Factory aborda el problema de la creación de familias de

objetos (como por ejemplo interfaces gráficas) que comparten toda una serie de características comunes en los objetos que componen dichas familias.

En nuestro caso se da en la creación de familias de componentes gráficos en las cuales los elementos de la interfaz se mantienen constantes (por ejemplo, componentes de bloques y puertas, botones, cajas de texto ...) pero el dibujado de dichos elementos puede delegarse en distintas familias de forma que, en función de la fábrica seleccionada obtenemos unos componentes u otros.

### Diseño de la Base de datos

Nuestra base de datos cuenta con tres tablas. La tabla Mapa guardara los datos relacionados con los mapas, es decir, para cada mapa, la organización de los componentes, las celdas ocultas, si está activo o no. La tabla de Personaje tendrá la posición en la que se encuentra en el mapa, la cantidad de llaves encontradas y la puntuación. La tabla de Problemas, tendrá el texto del problema, las opciones, la opción correcta, si está resuelto o no, y el tipo de problema.

A continuación, se muestra el diagrama entidad relación de la base de datos de la aplicación

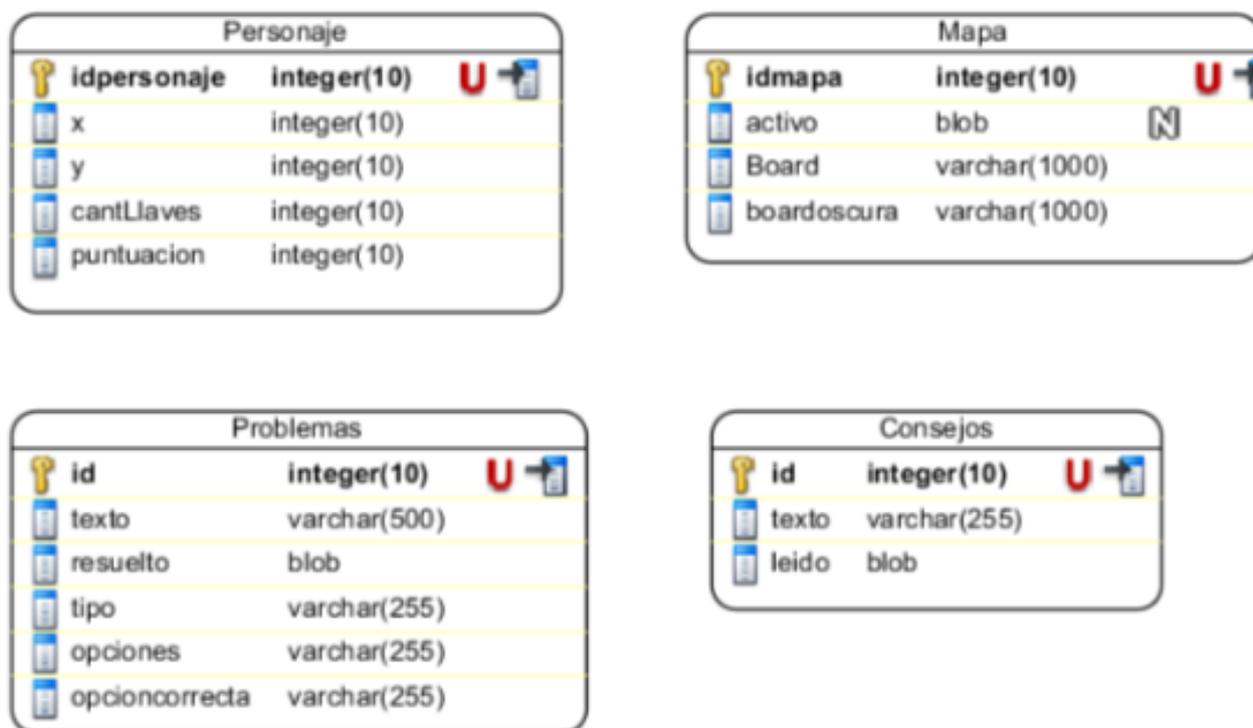


Ilustración 5 Diagrama Entidad-Relación. Fuente: Elaboración propia

## **2.5. Análisis de riesgos**

Este proceso se toma en cuenta durante todo el proceso de creación del videojuego, tiene como objetivo minimizar el impacto de problemas, establecer una probabilidad e impacto de posibles riesgos existentes, además, permite prever posibles problemas de programación y establecer métodos de solución para estos problemas.

A continuación, se muestra una tabla con los detectados.

ID Riesgo	Fecha Identificado	Descripción corta	Descripción	Impacto	Probabilidad (%)	Magnitud	Estrategia de mitigación	Estrategias de contingencia
1	1/10/2018	El juego puede tener carencias en la parte gráfica	No existen artistas gráficos profesionales por lo que los diseños y edición de las imágenes quedaran por debajo de la media	5	70	3.5	Reducir al máximo la cantidad de assets diferentes que se tengan que crear.	Buscar gráficos gratuitos en internet. (generalmente no vienen con texturas y se pierde calidad y originalidad).
2	6/10/2018	Pueden faltar sonidos cuando se termine el juego	Nunca se pensó en cómo vamos a hacer los sonidos. Si bien puede ser importante, tal vez no sea algo	1	70	0.7	Se deben crear tareas dentro de desarrollo para añadir sonidos.	Buscar sonidos en internet

			determinante en el juego, ya que es un caso de estudio, si se toma el juego como prototipo.					
3	6/10/2018	El proceso puede estar errado	No tenemos experiencia en el desarrollo de videojuegos. Puede ocurrir que no funcione bien y el juego se retrase, incrementar el retrabajo, entre otras cosas. El impacto	4	30	1.2	Pedir asesoramiento a los tutores. Anotar cada desviación que veamos del proceso lo antes posible e intentar corregirla.	

			puede ser grande.					
4	6/10/2018	Retraso en los tiempos de entrega	Como tenemos falta de experiencia podemos estimar mal, agregar demasiadas cosas, etc. y esto puede llevar a que el juego no se termine en tiempo y forma. Además, tenemos exámenes en diciembre y/o febrero.	1	50	0.5	Acotar los requerimientos, pero intentar cerrar algo jugable. Ajustar lo mejor posible las estimaciones.	

5	6/10/2018	La tecnología puede no responder a las necesidades del juego	Como nunca probamos la tecnología puede pasar que sea complicado o imposible implementar algún componente.	1	20	0.2	Realizar prototipos.	Cambiar requerimientos que sean difíciles de implementar.
---	-----------	--	--	---	----	-----	----------------------	---

Tabla 11 Análisis de riesgo. Fuente: elaboración propia

## **Conclusiones del capítulo**

En este capítulo se mostraron las etapas de concepto y planificación según la metodología SUM con sus artefactos, donde se definen los aspectos de la aplicación tanto técnicos como de negocio. Se muestra el cronograma del desarrollo de LogiLab, se definen los roles y la especificación del producto, se definen los presupuestos y costos y se muestra el análisis de riesgo realizado.

# Capítulo 3: Resultados y pruebas

En este capítulo se realizan las pruebas al software que permiten comprobar la calidad de este producto, lo que constituye uno de los pasos más importantes en el desarrollo de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de mostrar un error no descubierto hasta entonces y con el fin de verificar la fiabilidad y calidad de la aplicación como un todo.

## 3.1. Pruebas del software

El testeo es la actividad realizada para evaluar la calidad del producto y para mejorar identificando los defectos y problemas. Las pruebas de software consisten en la verificación dinámica del comportamiento del programa en una lista finita de casos de prueba, seleccionadas del dominio de las ejecuciones del usuario contra el resultado esperado. (Abran, Moore, Bourque, Dupuis, & Tripp, 2004)

“Si el software se desarrolla como un producto que va a ser usado por muchos clientes, no es práctico realizar pruebas de aceptación formales con cada uno de ellos. La mayoría de los constructores de productos de software usan un proceso llamado prueba alfa y prueba beta para descubrir errores que al parecer sólo el usuario final es capaz de encontrar.” (Pressman, 2010)

### Plan de pruebas

Según (Loraiza, 2010) el plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento. Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está entregando al cliente

Número	Componente u objetivo	Prueba a realizar
1	Botón Comenzar Partida en la pantalla principal	Test Comenzar
2	Botón Continuar Partida en la pantalla principal	Test Continuar

3	Botón Guardar Partida en la pantalla de menú	Test Guardar
4	Botón Progresos en la pantalla de menú	Test Progresos
5	Botón Consejos en la pantalla de menú	Test Consejos
6	Botón Salir en la pantalla de menú	Test Salir
7	Puerta con problema de texto	Test Accionar puerta con problema de texto
8	Puerta con problema de algoritmización	Test Accionar puerta con problema de algoritmización
9	Puerta con problema de Torre de Hanoi	Test Accionar puerta con problema de Torre de Hanoi
10	Puerta con problema de cruce	Test Accionar puerta con problema de cruce
11	Problema de texto	Test Problema de texto
12	Problema de algoritmización	Test Problema de algoritmización
13	Problema de Torre de Hanoi	Test Problema de Torre de Hanoi
14	Problema de cruce	Test Problema de cruce
15	Cofre de consejos	Test Cofre de consejos
16	Cofre de llaves	Test Cofre de llaves

*Tabla 12 Plan de Pruebas. Fuente: Elaboración propia*

### **Tablas de pruebas**

A continuación, se muestra una de las pruebas realizadas en el plan de pruebas. Para ver más, vea Anexo 1 Tablas de Pruebas

Tabla de Prueba	
Número de Prueba	1
Requerimiento	Tener la aplicación instalada
Objetivo	Probar el botón comenzar partida
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Inicie el juego con la historia correctamente
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

Tabla 13 Tabla de prueba. Fuente: Elaboración propia

Los errores detectados durante las pruebas anteriores, se solucionaron y se desplego una nueva versión para someterla a las pruebas Beta

### Pruebas Beta

La prueba alfa se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales. El software se usa en un escenario natural con el desarrollador “mirando sobre el hombro” de los usuarios y registrando los errores y problemas de uso. Las pruebas alfa se realizan en un ambiente controlado. La prueba beta se realiza en uno o más sitios del usuario final. A diferencia de la prueba alfa, por lo general el desarrollador no está presente. Por tanto, la prueba beta es una aplicación “en vivo” del software en un ambiente que no puede controlar el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que se encuentran durante la prueba beta y los reporta al desarrollador periódicamente. Como resultado de los problemas reportados durante las pruebas beta, es posible hacer modificaciones y luego preparar la liberación del producto de software a toda la base de clientes.(Pressman, 2010) En nuestra aplicación realizamos pruebas alfas durante el desarrollo y

pruebas Beta al terminar la aplicación. A continuación, mostramos uno de los reportes de las pruebas Betas al videojuego. Para ver más ver Anexo 2 Reportes de las Pruebas Beta

### Resultados de las pruebas Alfa y Beta

Id del fallo	1
Título	No puedo acceder un cofre
Descripción	Este cofre está dentro de un espacio que no tiene como entrar en el
Número de la versión usada	1.1
Fecha de la versión usada	05/14/19
Nombre del tester	José Enrique Díaz
Persona asignada al fallo	Osniel O. Domínguez
Tipo de fallo	Fallo en el diseño
Zona en la que aparece el fallo	En el segundo mapa
Pasos para llegar al fallo	<ul style="list-style-type: none"> <li>• Inicie el juego</li> <li>• Avance hasta el segundo mapa</li> <li>• El muro está cerrado completamente</li> </ul>
Prioridad	media

Tabla 14 Resultado de las pruebas Beta. Fuente: Elaboración propia

### Estado de los errores

En total se encontraron en las pruebas anteriores a la primera versión de la aplicación siete fallos, a continuación, se muestra una tabla con el estado de cada fallo

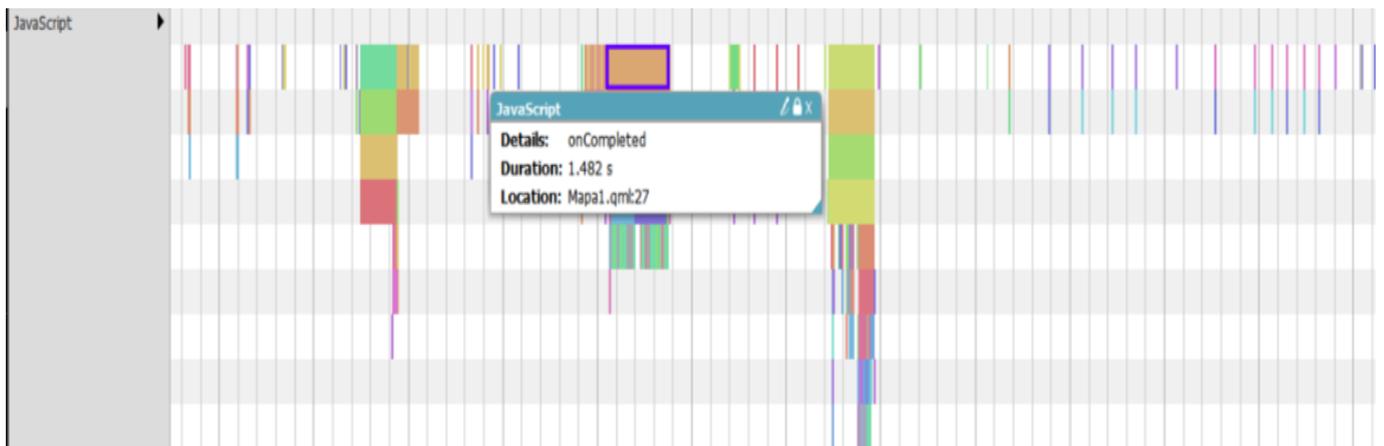
Id del fallo	Título del fallo	Estado	Detalles
1	No puedo acceder un cofre	Solucionado	Se cambio la estructura del mapa
2	La puerta para cambiar de mundo está mal ubicada	Solucionado	Se cambio la estructura del mapa
3	Mal giradas las puertas	Solucionado	Se cambiaron de puertas horizontales a verticales
4	El cofre no se quita	Solucionado	Se identifico en el código una variable con el nombre incorrecto, se sustituyó por el adecuado
5	Mal giradas las puertas	Solucionado	Se cambiaron de puertas verticales a horizontales
6	Mal giradas las puertas	Solucionado	Se cambiaron de puertas verticales a horizontales
7	Mal giradas las puertas	Solucionado	Se cambiaron de puertas horizontales a verticales

Tabla 15 Estado de los errores

### 3.2. Pruebas de rendimiento

Se utilizó la herramienta de testeo que QtCreator nos proporciona para ver el tiempo de carga y de creación de los componentes y métodos.

Al concluir este análisis el método que más tiempo de corrida tiene es el `onCompleted ()` del componente `Mapa1.qml`, que se encarga de cargar a todos los componentes dentro del mapa, y aun así no sobrepasa los 1.5 segundos tal y como se muestra en la Ilustración 6 Método con más tiempo de corrida de la aplicación. Fuente: Reporte de prueba de QtCreator Analize a continuación



*Ilustración 6 Método con más tiempo de corrida de la aplicación. Fuente: Reporte de prueba de QTCreator Analize*

Se realizó una prueba de rendimiento al microprocesador de un dispositivo móvil marca Blu DashX con sistema Android 4.4 con 959mb de memoria RAM y CPU ARM Córtes a 1300 MHz durante aproximadamente 2 horas de juego continuo y el dispositivo en ningún momento llegó a tener el microprocesador por encima del cincuenta por ciento de uso. En las imágenes del Anexo 3 Imágenes de la se muestran las fotos tomadas durante ese proceso.

### 3.3. Impacto de riesgos

Id Riesgo	Impacto en la aplicación
1	A pesar de que no existían artistas gráficos las imágenes están en la media
2	Se introdujeron la menor cantidad de sonidos a la aplicación, pero estos tienen buena calidad sonora
3	Se contó con los consejos de varios profesores de la universidad con experiencia metodológica. No influyó en la aplicación
4	Se tuvieron que corregir un poco los tiempos de estimación durante el desarrollo, por la adición de nuevos componentes a petición del cliente, pero las entregas se

	realizaron como estaban planeadas sin retrasos al igual que las reuniones con los clientes.
<b>5</b>	No tuvo impacto ninguno en la aplicación. Se implementaron correctamente todos los componentes planificados

Tabla 16 Impacto de riesgos. Fuente: Elaboración propia

### 3.4. Resultados obtenidos

Después de las pruebas realizadas, y reparados los errores encontrados, se lanzó una versión estable. A varios de los jugadores se les pidió rellenar una encuesta antes y después de jugar la aplicación. A continuación, se muestra dicha encuesta.

Preguntas	Marca con una x la puntuación				
	1	2	3	4	5
<b>Historia</b>					
<b>Jugabilidad</b>					
<b>Rapidez</b>					
<b>Diseño</b>					
<b>Experiencia</b>					
<b>Problemas</b>					
<b>Impacto en el jugador</b>					

Tabla 17 Encuesta aplicada. Fuente: Elaboración propia

Se obtuvo un promedio de:

Preguntas	Promedio
Historia	3
Jugabilidad	3
Rapidez	4
Diseño	3
Experiencia	4
Problemas	4
Impacto en el jugador	4

Tabla 18 Resultados de la encuesta. Fuente: Elaboración propia

### **Análisis de los resultados obtenidos**

Después de desarrollar todo un proceso de pruebas con un nivel medio de sencillez se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las realizadas, se solucionaron varios problemas que impedían el cumplimiento de los requisitos fundamentales del sistema en cuestión. Las pruebas fueron planeadas y ejecutadas de forma que abarcara completamente el software, permitiendo esto obtener una aplicación que se cuyas funciones se encuentran en correspondencia con las especificaciones acordadas y que además cumple con los requerimientos de rendimiento. El desarrollo de la aplicación cumple las expectativas trazadas al inicio del proyecto.

### **3.5. Conclusiones del capítulo**

Las pruebas realizadas utilizando las técnicas anteriormente planteadas fueron de gran importancia para demostrar el buen funcionamiento del software y el cumplimiento de las peticiones del cliente. La implementación del software y la aplicación de las pruebas de validación con resultados satisfactorios demostraron que el software elaborado cumple con los requerimientos especificados por el cliente. Esta aplicación les dará a los jugadores la herramienta para elevar las habilidades de lógica, y en este capítulo

se garantizó que fuese sin errores en el juego y cumpliera con las expectativas de los jugadores a través de las encuestas realizadas a los jugadores Beta.

# Conclusiones Generales

- Se buscó y analizó el trabajado relacionado con el objeto de estudio y antecedentes de la investigación.
- Se definieron las habilidades de lógica necesarias para la investigación
- Se desarrollo un prototipo del juego educativo.
- Se demostró la viabilidad y funcionamiento del prototipo del juego.

# Recomendaciones

- Poner en marcha la aplicación en la Universidad de Matanzas como herramienta para los profesores de Matemática discreta e Introducción a la Programación
- Ampliar la aplicación, incorporándole nuevas funcionalidades y problemas
- Incorporar una función que de un reporte de estadística por jugador donde se informe el nivel de cada habilidad
- Extender su uso a otras Universidades

## Bibliografía

- Abran, A., Moore, J. W., Bourque, P., Dupuis, R., & Tripp, L. L. (2004). *Guide to the Software Engineering Body of Knowledge*: Angela Burgess Group Managing Editor.
- Una Metodología para Desarrollo de Videojuegos*, (2009).
- Arteaga, J. G. (2014). *ESTUDIO COMPARATIVO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE*. Universidad de Nariño, San Juan de Pasto.
- Aslan, S. (2016 ). *Digital Educational Games: Methodologies for Development and Software Quality*
- Bergman, P. (2003). *Digital Games and Learning: A Research Overview*.
- Carvalho, P. F. P. d. (2007). *Apostila de Lógica de Programação - ALGORITMOS -*
- Cavallari, J., Hedberg, Harper, B. (1992). Adventure games in education: A review. . *Australian Journal of Educational Technology.*, 172-184.
- Corporation, T. (2019). Human Resource Machine. from <https://tomorrowcorporation.com/humanresourcemachine>
- Corporation, V. P. (2019). Visual Paradigm 2019, from <https://www.visual-paradigm.com/>
- Delavy, E. (2017). Lógica de programación: el primer paso para aprender a programar.
- Diccionario de Sinónimos y Antónimos (8va ed.). Cuba: Editorial Científico-Técnica.
- DIGIA. Model/View Programming. 2019, from <http://qt-project.org/doc/qt-4.8/qtquick.html>
- Drösser, C. (2013). *La Seducción de la Lógica* (Planeta Ed. Primera edición ed.). Barcelona
- Egenfeldt-Nielsen. (2006). *Overview of research on the educational use of video games*. (Vol. 1).
- Entertainment, C. (2019). River crossing iq. from <https://play.google.com/store/apps/details?id=com.coscia.quasong.riveriq&hl=en>
- Ferguson, C. J. (2010). Blazing Angels or Resident Evil? Can Violent Video Games Be a Force for Good? *Review of General Psychology*, 14.
- Freitas, D. (2005). Learning through Play. Using educational games and simulations to support post-16 learners. *London Learning and Skills Research Centre*.
- Gardner, H. (1987). *teoría de las Inteligencias Múltiples*: Universidad Harvard
- Garrido., I. Y. V. (2019). *Contribución de la Lógica Matemática a la formación inicial del Ingeniero Informático*. (Máster en Matemática Educativa. Tesis presentada en opción al título académico de Máster en Matemática Educativa.), Universidad de Matanzas sede "Camilo Cienfuegos".
- GITOO, G. d. I. e. T. O. p. O.-., CASTRO, C. A. C., MURILLO, E. E. L., & PALACIO, J. C. (2001). Hacia una nueva forma de enseñar la lógica de programación. Retrieved from Revista académica y científica de UNIREMINGTON website: <https://www.researchgate.net/publication/258441405>
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*
- Jiménez, Á. I., Rico, D. P., Méndez, C. M., Ceron, S., & Betancourt, S. P. (2010). Procesos y técnicas de

- ingeniería de Software  
para la Modelación de  
Videojuegos. *Inventum*, 9.
- JRyannel, J. (2015). Qt5 Cadaques.
- La neurona. (2019, 6/10/2019). 2019, from [http://cubapk.com/store/cu.vertex\\_ica.laneurona](http://cubapk.com/store/cu.vertex_ica.laneurona)
- Lantigua., I. M. E. (2017). *Sistema de acciones para contribuir a la formación del Ingeniero Informático a partir de las relaciones interdisciplinarias de Matemáticas Discretas con otras asignaturas bases*. (Máster en Matemática Educativa. Tesis presentada en opción al título académico de Máster en Matemática Educativa.), Universidad de Matanzas sede "Camilo Cienfuegos".
- Lee, J., & Hammer, J. (2014). Gamification in Education: What, How, Why Bother? *Academic Exchange Quarterly*.
- Lindley, C. (2012). <JavaScript\_Succinctly.pdf>
- LLC, S. (2019). Ligthbot.
- Loraiza. (2010). Plan de Pruebas de Software.
- Maie, M. W., Emery, D., & Hilliard, R. Recommended Practice for Architectural Description for Software Intensive Systems.
- Margarida Romero, M. U., and Michela Ott. (2015). Can Serious Games Contribute to Developing and Sustaining 21st Century Skills? *Vol. 10(2)* 148-177
- Martínez., J. B., & Martín., J. P. (2012). Uso de los videojuegos en el tratamiento contra el dolor. *Revista de Estudios de Juventud*.
- Medina, M. P. L. La educación y la teoría de las inteligencias múltiples de Howard Gardner
- Molina, D. E. A. (2011). *Especificacion tecnologica en desarrollo de aplicaciones de dispositivos moviles*.
- Moroni, N., & Señas, P. (2005). Estrategias para la enseñanza de la programación Primeras Jornadas de Educación en Informática y TICS en Argentina.
- MURILLO-SANCHEZ, X.-A., GUTIÉRREZ-ROCHA, A.-L., IBAÑEZ-ILLANES, A.-W., QUIROZ-PEREZ, J.-A., SAHONERO-ALVAREZ, G., & DÍAZ-PALACIOS, F.-R. (2018). *Implementación de la metodología SUM modificada para el desarrollo de videojuegos orientados al aprendizaje en Bolivia* Paper presented at the Décima Séptima Conferencia Iberoamericana en Sistemas, Cibernética e Informática
- Noguero, A. M. (2010). ADVERGAMING:CONCEPTO, TIPOLOGÍA,ESTRATEGIAS Y EVOLUCIÓN HISTÓRICA. *REVISTA DE COMUNICACIÓN Y NUEVAS TECNOLOGÍAS*.
- Ortiz-Colón, A.-M., Jordán, J., & Agredal, M. (2018). Gamificación en educación: una panorámica sobre el estado de la cuestión. *Educ. Pesqui*.
- Padilla, N., & Collaso. (2012). Videojuegos Educativos: Teorías y Propuestas Para el Aprendizaje en Grupo. *Ciencia e Ingeniería Neogranadina*.
- Pascual, I. J. (2005). *Lógica Simbólica para Informáticos* (Primera edición ed.). Primera edición.

- Prado, M. A. R. La lógica de programación como instrumento. Retrieved from <http://www.ilustrados.com/> website:
- Pressman, R. S. (2010). *Ingeniería del software*  
*UN ENFOQUE PRÁCTICO*
- Schwaber, K., & Sutherland, J. (2013). *La Guía de Scrum*
- SCIENTISTS, F. O. A. (2006). Harnessing the power of video games for learning. Summit on Educational Games.
- Serna, E. M. (2013). *Revista educación en ingeniería*.
- Significados.com. (2019, 04/03/2019). Photoshop. Retrieved 14 de junio de 2019,, 2019
- Smith-Robbins, S. (2011). "This Game Sucks":  
How to Improve the  
Gamification of Education. *New horizons*.
- Tardón, C. G. (2014). *VIDEOJUEGOS PARA LA TRANSFORMACIÓN SOCIAL* Bilbao.
- TENJICA, C. S. C. (2016). *LOGICA Y ALGORITMOS*  
*LOGICAS, TIPOS Y SUS DERIVADAS*. (INGENIERIA DE SISTEMAS SEGUNDO SEMESTRE),  
UNIVERSIDAD CUNDINAMARCA, CUNDINAMARCA.
- Trejos, O. I. (2004). *La Esencia de la Lógica de Programación*.
- Tutu2. (2019, 2019). from <http://cubapk.com/store/com.dnis3d.tutu4Game#>
- Vasquez, A. E. (2007). *Una alternativa didáctica en la enseñanza de la lógica para estudiantes de primero de bachillerato en computación*. (Master), Universidad pedagógica nacional Francisco Morazan San Pedro Sula.
- Walter, R. (1999). *Lógica simbólica para todos*. (U. Veracruzana Ed. Primera edición ed.). Xalapa, Veracruz.

# Anexos

## Anexo 1 Tablas de Pruebas

Tabla de Prueba	
Número de Prueba	1
Requerimiento	Tener la aplicación instalada
Objetivo	Probar el botón comenzar partida
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Inicie el juego con la historia correctamente
Resultado obtenido	Satisfactorio(x) insatisfactorio()

Tabla 19 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	2
Requerimiento	Tener la aplicación instalada
Objetivo	Probar el botón continuar partida
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android

Personal	Equipo de SUM
Resultado esperado	Inicie el juego en el mapa donde se encontraba el jugador antes de salir, la sombra sobre el mapa debe estar despejada según lo explorado por el jugador en el anterior juego
Resultado obtenido	Satisfactorio() insatisfactorio(x)

Tabla 20 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	3
Requerimiento	Tener la aplicación instalada
Objetivo	Probar el botón guarda partida en la pantalla del menu
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	El juego se guarda correctamente en la Base de datos
Resultado obtenido	Satisfactorio() insatisfactorio(x)

Tabla 21 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	4
Requerimiento	Tener la aplicación instalada
Objetivo	Probar el botón Progresos en la pantalla de menú
Hardware	Celular Android Blu Dash x de gama media-baja

Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	El juego muestra la pantalla de progresos
Resultado obtenido	Satisfactorio(x) insatisfactorio()

Tabla 22 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	5
Requerimiento	Tener la aplicación instalada
Objetivo	Probar el botón Consejos en la pantalla de menú
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	El juego muestra la pantalla de consejos
Resultado obtenido	Satisfactorio(x) insatisfactorio()

Tabla 23 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	6
Requerimiento	Tener la aplicación instalada

Objetivo	Probar el botón Salir en la pantalla de menú
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	El juego se termina.
Resultado obtenido	Satisfactorio(x) insatisfactorio()

Tabla 24 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	7
Requerimiento	Tener la aplicación instalada
Objetivo	Probar Accionar puerta con problema de texto
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre el problema correctamente
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

Tabla 25 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	8

Requerimiento	Tener la aplicación instalada
Objetivo	Probar Accionar puerta con problema de algoritmización
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre el problema correctamente
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

Tabla 26 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	9
Requerimiento	Tener la aplicación instalada
Objetivo	Probar Accionar puerta con problema de Torre de Hanoi
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre el problema correctamente
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

Tabla 27 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	10
Requerimiento	Tener la aplicación instalada
Objetivo	Probar Accionar puerta con problema de cruce
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre el problema correctamente
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

Tabla 28 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	11
Requerimiento	Tener la aplicación instalada
Objetivo	Probar problema de texto
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre el problema correctamente con las opciones correctas
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

Tabla 29 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	12
Requerimiento	Tener la aplicación instalada
Objetivo	Probar los Problema de algoritmización
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre el problema correctamente, los componentes deben estar bien ubicados, teniendo, el jugador, acceso a la marca final en la cantidad de movimientos necesarios
Resultado obtenido	Satisfactorio() insatisfactorio(x)

Tabla 30 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	13
Requerimiento	Tener la aplicación instalada
Objetivo	Probar los Problema de Torre de Hanoi
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM

Resultado esperado	Se muestre el problema correctamente, los componentes deben estar bien ubicados, la caja grande siempre debe estar encima de una chiquita, se debe poder mover solo las de arriba.
Resultado obtenido	Satisfactorio(x) insatisfactorio()

Tabla 31 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	14
Requerimiento	Tener la aplicación instalada
Objetivo	Probar los Problema de cruce
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre el problema correctamente, los componentes deben estar bien ubicados, no se puede cruzar cuando hay dos componentes seguidos en esa dirección.
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

Tabla 32 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	15
Requerimiento	Tener la aplicación instalada
Objetivo	Probar los Cofre de consejos
Hardware	Celular Android Blu Dash x de gama media-baja

Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre la animación del consejo correctamente y se adicione a la base de datos.
Resultado obtenido	Satisfactorio(x) insatisfactorio()

Tabla 33 Tabla de prueba. Fuente: Elaboración propia

Tabla de Prueba	
Número de Prueba	16
Requerimiento	Tener la aplicación instalada
Objetivo	Probar los Cofre de llaves
Hardware	Celular Android Blu Dash x de gama media-baja
Software	Sistema operativo Android
Personal	Equipo de SUM
Resultado esperado	Se muestre la animación de la llave correctamente y se adicione a la base de datos.
Resultado obtenido	Satisfactorio(x) insatisfactorio ()

## Anexo 2 Reportes de las Pruebas Beta

<b>Id del fallo</b>	2
<b>Título</b>	La puerta para cambiar de mundo está mal ubicada

Descripción	La puerta para cambiar de mundo está mal ubicada, esta queda fuera del muro
Número de la versión usada	1.1
Fecha de la versión usada	05/14/19
Nombre del tester	José Enrique Díaz
Persona asignada al fallo	Osniel O. Domínguez
Tipo de fallo	Fallo en el diseño
Zona en la que aparece el fallo	En el séptimo mapa
Pasos para llegar al fallo	<ul style="list-style-type: none"> <li>• Ejecutar el juego normalmente</li> <li>• Avanzar hasta el séptimo mundo</li> <li>• En la esquina inferior derecha se encuentra la puerta</li> </ul>
Prioridad	media

Tabla 34 Resultado de las pruebas Beta. Fuente: Elaboración propia

Id del fallo	3
Título	Mal giradas las puertas
Descripción	Las puertas del segundo mapa se encuentran horizontalmente en vez de verticalmente
Número de la versión usada	1.1
Fecha de la versión usada	05/14/19
Nombre del tester	José Enrique Díaz

Persona asignada al fallo	Osniel O. Domínguez
Tipo de fallo	Fallo en el diseño
Zona en la que aparece el fallo	En el segundo mapa
Pasos para llegar al fallo	<ul style="list-style-type: none"> <li>• Ejecutar el juego normalmente</li> <li>• Avanzar hasta el segundo mundo</li> </ul>
Prioridad	media

*Tabla 35 Resultado de las pruebas Beta. Fuente: Elaboración propia*

Id del fallo	4
Título	El cofre no se quita
Descripción	El cofre en el quinto mapa no se quita al igual que los otros cuando me da la llave
Número de la versión usada	1.1
Fecha de la versión usada	05/14/19
Nombre del tester	José Enrique Díaz
Persona asignada al fallo	Osniel O. Domínguez
Tipo de fallo	Fallo en la programación
Zona en la que aparece el fallo	En el quinto mapa
Pasos para llegar al fallo	<ul style="list-style-type: none"> <li>• Ejecutar el juego normalmente</li> <li>• Avanzar hasta el quinto mundo</li> <li>• Camine recto hacia la derecha</li> </ul>

	<ul style="list-style-type: none"><li>• Abra la puerta</li><li>• Baje directo hasta encontrar la otra puerta y pase a través de ella</li><li>• Suba hasta la esquina superior derecha donde se encuentra el cofre</li></ul>
<b>Prioridad</b>	media

*Tabla 36 Resultado de las pruebas Beta. Fuente: Elaboración propia*

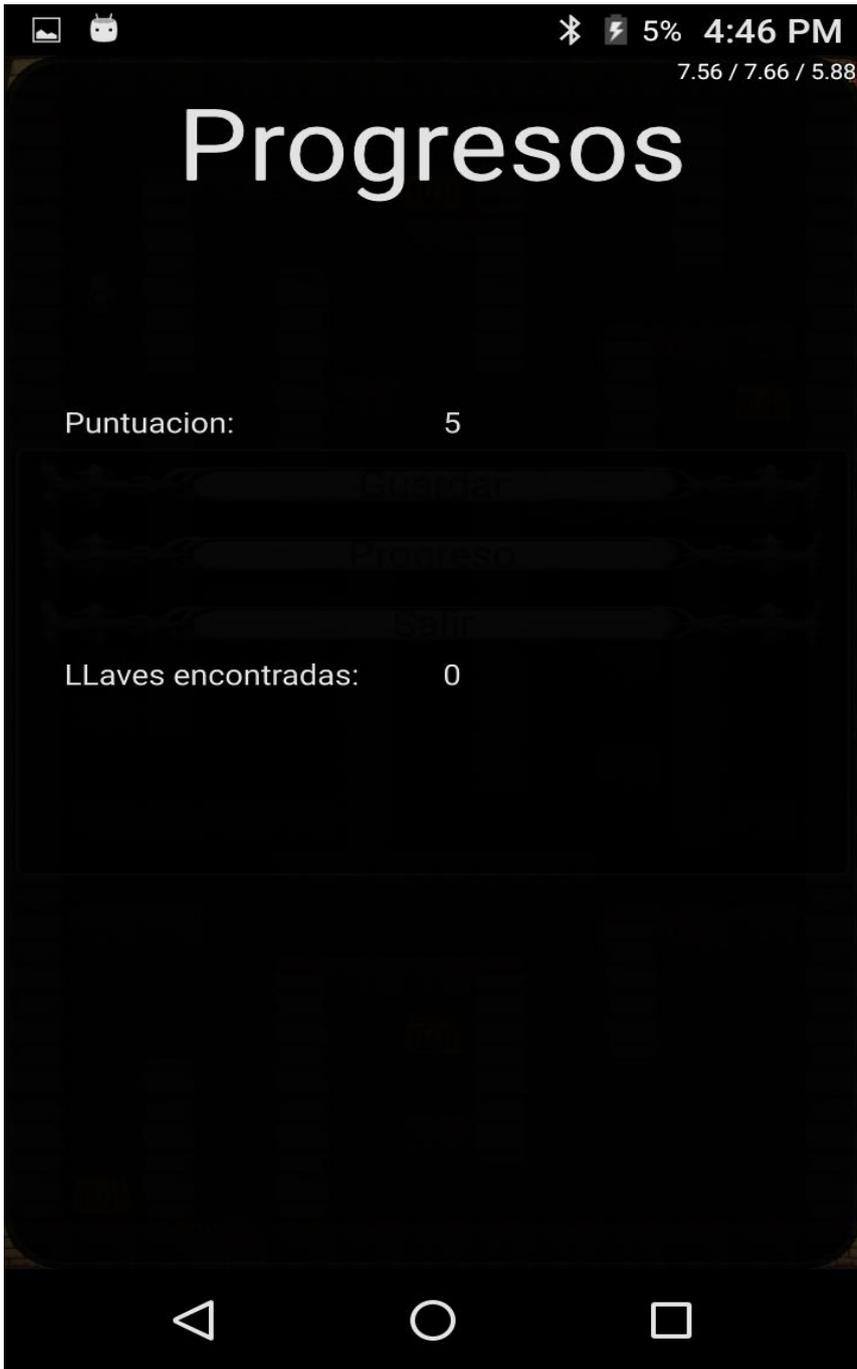
### Anexo 3 Imágenes de la aplicación



Ilustración 7 Imagen del juego.



Ilustración 8 Imagen del juego



*Ilustración 9 Imagen del juego*

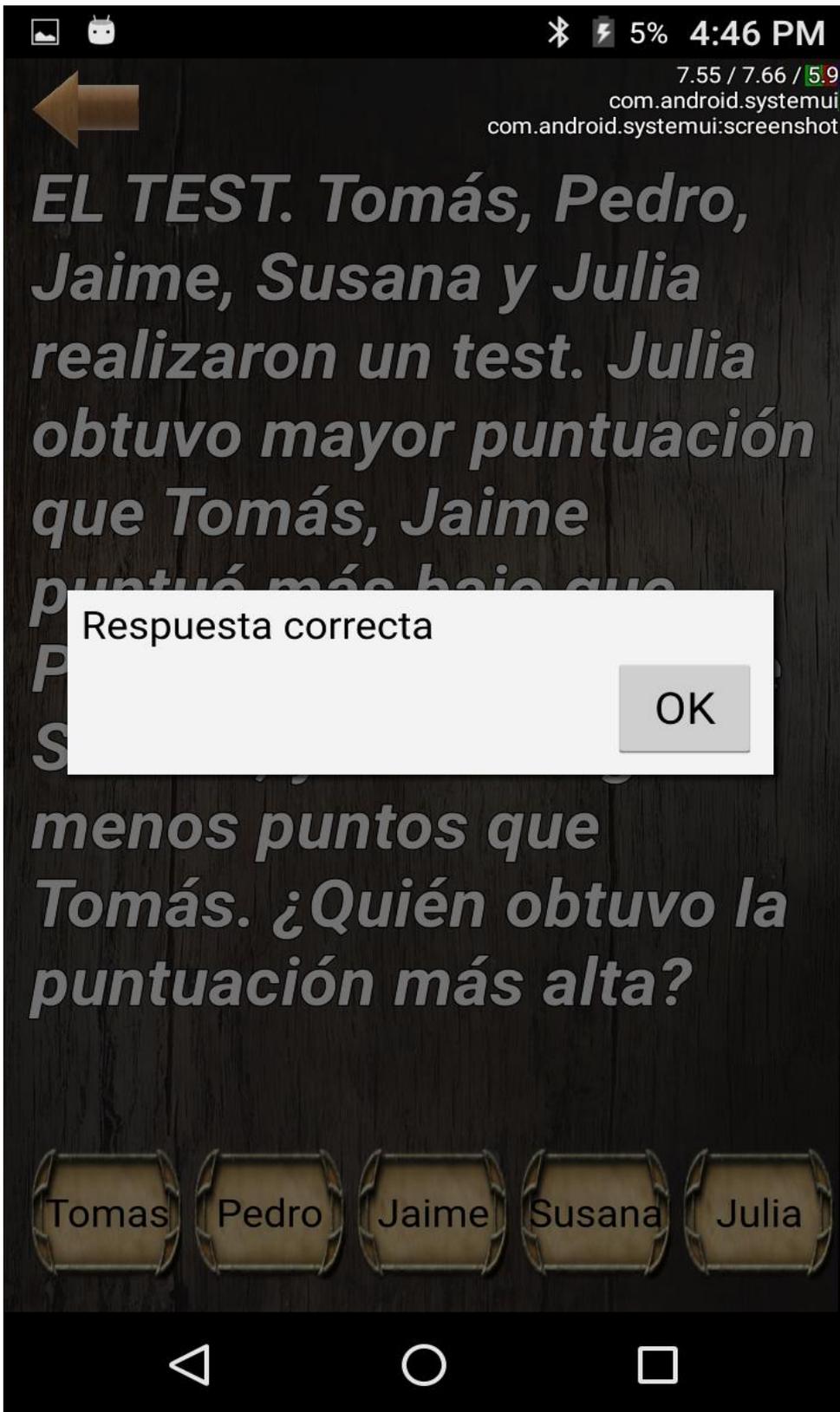


Ilustración 10 Imagen de la aplicación