

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERÍA Y ARQUITECTURA
Escuela Profesional de Ingeniería de Sistemas



**Desarrollo de un Serious Gaming para estudiantes de Primer Año
de Primaria aplicando la Metodología SUM**

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Por:
Kevin Maximiliano Arenas Cancapa

Asesor:
Ing. Gómez Apaza Roel Dante

Juliaca, agosto de 2018

DECLARACIÓN JURADA DE AUTORÍA DEL INFORME DE TESIS

Roel Dante Gomez Apaza, de la facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: "Desarrollo de un Serious Gaming para estudiantes de Primer Año de Primaria aplicando la Metodología SUM" constituye la memoria que presenta el Bachiller Kevin Maximiliano Arenas Cancapa para aspirar al título de Profesional de Ingeniero de Sistemas, cuya tesis ha sido realizada en la Universidad Peruana Unión bajo la dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca, a los 06 de febrero del 2020



Roel Dante Gomez Apaza

Desarrollo de un Serious Gaming para estudiantes de Primer Año de
Primaria aplicando la Metodología SUM

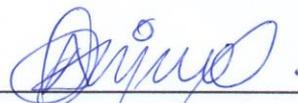
TESIS

Presentada para optar el título profesional de Ingeniero de Sistemas

JURADO CALIFICADOR



Mtro. Lennin Henry
Centurión Julca
Presidente



Ing. Angel Rosendo
Condori Coaquira
Secretario



Ing. Eder Gutierrez
Quispe
Vocal



Ing. David Mamani Pari
Vocal



Mtro. Roel Dante Gómez Apaza
Asesor

Juliaca, 03 de Diciembre del año 2019

DEDICATORIA

Esta investigación está dedicada a los niños de la comunidad de Occoro perteneciente al distrito de Achaya, porque inspirado en ellos nació la idea de este proyecto investigación, y aunque no esté orientada de manera directa para ellos espero que el desarrollo de este proyecto a gran escala o por medio de investigaciones relacionadas puedan brindar una herramienta de apoyo para sus estudiantes.

AGRADECIMIENTOS

Hay muchas personas a las que agradecer por el desarrollo de este proyecto, pero en especial quiero agradecer a Dios y al esfuerzo de mi madre por apoyarme y entenderme en cada decisión y proyecto, agradecer a mi universidad por permitirme tener tan buena experiencia universitaria, gracias a mis amigos casi hermanos que siempre estuvieron ahí cada vez que lo necesitaba, gracias a cada maestro que formo parte del desarrollo de este proyecto de tesis.

Para finalizar agradezco a quien lee este apartado y más de mi tesis, por permitir a mis experiencias, investigaciones y conocimientos, ser parte de su repertorio de información mental.

TABLA DE CONTENIDO

DEDICATORIA.....	iv
AGRADECIMIENTOS.....	v
TABLA DE CONTENIDO.....	vi
LISTA DE TABLA.....	viii
ÍNDICE DE FIGURAS.....	x
INDICE DE ANEXOS.....	xii
RESUMEN.....	xiii
ABSTRACT.....	xiii
CAPÍTULO I. EL PROBLEMA.....	15
1.1. IDENTIFICACIÓN DEL PROBLEMA.....	15
1.2. JUSTIFICACIÓN.....	17
1.3. PRESUNCION FILOSOFICA.....	18
1.4. OBJETIVOS.....	18
1.4.1. Objetivo General.....	18
1.4.2. Objetivos Específicos.....	19
CAPÍTULO II. BASES TEÓRICAS.....	20
2.1. REVICIÓN LITERARIA.....	20
2.2. MARCO TEÓRICO.....	21
2.2.1. Juegos Serios.....	21
2.2.2. Documento de Diseño de Videojuegos (GGD).....	22
2.2.3. Metodología de Desarrollo SUM.....	23
2.2.4. ¿Porque la metodología SUM?.....	26
2.2.5. Metodología de Desarrollo de Videojuegos Educativos de Pérez Marques.....	28
2.2.6. ¿Porque la metodología de Pérez Marques?.....	30
2.2.7. Godot Engine como motor de Videojuego.....	33

2.2.8. ¿Porque Godot Engine?	34
CAPÍTULO III. MATERIALES Y MÉTODOS	37
3.1. DESCRIPCIÓN DEL LUGAR DE EJECUCIÓN	37
3.2. MATERIALES E INSUMOS	37
3.2.1. Lenguaje de Programación	37
3.3. METODOLOGÍA.....	39
3.3.1. Tipo de Investigación	39
3.3.2. Arquitectura Solución.....	40
3.3.3. Diseño de la Investigación.....	40
3.4. DESARROLLO.....	40
3.4.1. CONCEPCIÓN.....	41
3.4.2. PLANIFICACIÓN	53
3.4.3. ELABORACIÓN	56
3.4.4. BETA.....	68
3.4.5. CIERRE.....	74
3.4.6. LIBERACIÓN EN LA INSTITUCIÓN EDUCATIVA.....	80
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN	86
4.1. RESULTADOS	86
4.1.1. Resultado Uno	86
4.1.2. Resultado Dos.....	86
4.1.3. Resultado Tres	87
4.1.4. Resultado Cuatro	103
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....	107
5.1. CONCLUSIONES.....	107
5.2. RECOMENDACIONES	107
CAPÍTULO VI. LISTA DE REFERENCIAS	109
CAPÍTULO VII. ANEXOS	111

LISTA DE TABLA

Tabla 1 <i>Metodologías de desarrollo Viabes para el desarrollo de Juegos</i>	26
Tabla 2 <i>Metodologías para el desarrollo de Software Educativo</i>	31
Tabla 3 <i>Motores de Desarrollo</i>	35
Tabla 4 <i>Características de Software para el Motor de Videojuego</i>	36
Tabla 5 <i>Materiales e Insumos para el desarrollo del software</i>	37
Tabla 6 <i>Identificación de Necesidades para el Juego Serio</i>	42
Tabla 7 <i>Desempeños de la competencia Resuelve Problemas de Cantidad</i>	43
Tabla 8 <i>Competencias del Programa Curricular</i>	45
Tabla 9 <i>Objetivos del videojuego de acuerdo a Competencias</i>	51
Tabla 10 <i>Presupuesto para Recursos Humanos Mensual</i>	54
Tabla 11 <i>Presupuesto para Servicios Mensual</i>	54
Tabla 12 <i>Presupuesto para Materiales un solo pago</i>	55
Tabla 13 <i>Autofinanciamiento</i>	55
Tabla 14 <i>Características Funcionales y nivel de prioridad del Juego Serio</i>	55
Tabla 15 <i>Características no Funcionales y nivel de prioridad del Juego Serio</i>	56
Tabla 16 <i>Relación de interacciones, requisitos e interfaces</i>	56
Tabla 17 <i>Seguimiento de las interacciones</i>	65
Tabla 18 <i>Versiones del videojuego</i>	67
Tabla 19 <i>Interfaces verificadas del videojuego</i>	70
Tabla 20 <i>Lista de Errores detectados y Solucionados</i>	73
Tabla 21 <i>Representación de respuestas de la encuesta a números</i>	83
Tabla 22 <i>Nivel de interés de los estudiantes para el juego serio</i>	84
Tabla 23 <i>Cantidad de Líneas de cada script dentro de la escena Menú Principal</i>	88
Tabla 24 <i>Cantidad de líneas de cada script dentro de la escena Menú Niveles</i>	89
Tabla 25 <i>Cantidad de líneas de cada script dentro de la escena Menú Principal</i>	90
Tabla 26 <i>Cantidad de líneas de cada script dentro de la escena Paleta Nivel</i>	92
Tabla 27 <i>Cantidad de líneas de cada script dentro de la escena Tablero de Instrucciones</i>	93
Tabla 28 <i>Cantidad de líneas de cada script dentro de la escena Personaje</i>	94
Tabla 29 <i>Coordenadas de movimiento para el personaje</i>	95
Tabla 30 <i>Sistema de ataque de acuerdo al tipo de enemigo</i>	97

Tabla 31	<i>Cantidad de líneas de cada script dentro de las escenas de objetos colisión....</i>	101
Tabla 32	<i>Cantidad de líneas de cada script dentro de la escena puntos</i>	102
Tabla 33	<i>Cantidad de líneas de cada script dentro de las escenas npc</i>	103

ÍNDICE DE FIGURAS

Figura 1 Nivel académico Peruano según OCDE	15
Figura 2 Nivel matemático estudiantes de Primaria	16
Figura 3 Estructura Metodología SUM	25
Figura 4 Metodología de Pérez Marqués para el desarrollo de software Educativo	29
Figura 5 Diseño de la Investigación para Serious Gaming	40
Figura 6 Metodología de desarrollo para el software	41
Figura 7 Flujo del desarrollo del concepto	46
Figura 8 Boceto movimiento de cajas en el escenario.....	47
Figura 9 Boceto movimiento por instrucciones del jugador	47
Figura 10 Boceto Modo Pelea contra npc	48
Figura 11 Boceto Interfaces del Videojuego	49
Figura 12 Boceto diseño de nivel tutorial.....	50
Figura 13 Escenario Bosque Menú Niveles Boceto	52
Figura 14 Flujo de tareas de planificación y administración.....	53
Figura 15 Flujo de la tarea desarrollo de características	56
Figura 16 Flujo de la tarea desarrollo de características	57
Figura 17 Tareas para el desarrollo del personaje	58
Figura 18 Sprite personaje principal.....	58
Figura 19 Movimiento del Personaje por touch	59
Figura 20 Colisión del Personaje.....	59
Figura 21 Escena Personaje en Godot Engine.....	60
Figura 22 Tareas para el desarrollo del tablero de instrucciones	60
Figura 23 Sprites de la escena Tablero de Instrucciones.....	61
Figura 24 Selección de Casilla Escena Tablero de Instrucciones	61
Figura 25 Dirección de movimiento del Personaje	62
Figura 26 Leer las instrucciones del Personaje	62
Figura 27 Escena Tabla de Instrucciones en Godot Engine	63
Figura 28 Tareas para el desarrollo de Tile Set.....	63
Figura 29 Sprites diseñados para Tile Set (Niveles).....	64
Figura 30 Escena Tile Set (Niveles) en Godot Engine	65
Figura 31 Flujo de monitoreo de la Interacción	65
Figura 32 Flujo del Cierre de la Interacción.....	66

Figura 33 Flujo de Interacción Beta	69
Figura 34 Flujo de tareas de planificación de la interacción	69
Figura 35 Verificación del Videojuego	70
Figura 36 Flujo de Corrección del videojuego	73
Figura 37 Flujo de Cierre del proyecto.....	75
Figura 38 Flujo de liberación del videojuego.....	75
Figura 39 Interfaz creación de KeyStore	76
Figura 40 Interfaz de exportación para Android	76
Figura 41 Interfaz de publicación de aplicación en Play Store	77
Figura 42 Interfaz aplicación publicada en Play Store	77
Figura 43 Interfaz de exportación a Sistema Operativo Windows.....	78
Figura 44 Flujo de Elaboración del Proyecto	78
Figura 45 Aplicación del juego serio en el Colegio Adventista Titicaca	82
Figura 46 Cuestionario para estudiantes.....	83
Figura 47 Nivel de interés de los estudiantes para el juego serio.....	84
Figura 48 Administración del desarrollo de características.....	87
Figura 49 Apariencia del Juego Educativo Duke	88
Figura 50 Nodos del menú principal	89
Figura 51 Animación de elementos del menú niveles.....	90
Figura 52 Identificación de niveles desbloqueados por partida guardada.....	91
Figura 53 Nodos del tablero de instrucciones	94
Figura 54 Movimiento del personaje por instrucciones	96
Figura 55 Nodos Raycast 2D de la escena personaje	96
Figura 56 Ajuste de nodo Tile Map en modo isométrico	98
Figura 57 Escena tileBloque para la construcción del niveles	99
Figura 58 Limitación de desplazamiento de la cámara del jugador dentro del nivel	100
Figura 59 Construcción de niveles de acuerdo boceto de diseño	101
Figura 60 Liberación del Juego Serio en el Colegio Adventista Titicaca	104
Figura 61 Nivel de interés de los estudiantes para el juego serio.....	105
Figura 62 Juego Serio Duke en Play Store	106

INDICE DE ANEXOS

Anexo A. Programa Curricular de Educación Primaria	111
Anexo B. Documento de Diseño del Videojuego GDD	111
Anexo C. Etapas de la Arquitectura del Software	111
Anexo D. Diagrama de Gantt	111
Anexo E. Encuestas estudiantes Primero de Primaria.....	111
Anexo F Documentos para la aplicación del software en la Institución Educativa	112

RESUMEN

En nuestro país existe una gran preocupación con respecto a la educación, específicamente en el bajo rendimiento de los estudiantes, una de las causas principales es el desinterés de los estudiantes en las materias. Con la finalidad de solucionar este problema se pretende usar una herramienta que logre captar el interés de los estudiantes. Para lograr esto se toma a los videojuegos como medio efectivo para estimular a los estudiantes y a la vez educarlos por medio de los Serious Gaming. Por esta razón el presente proyecto tiene como objetivo el desarrollo y liberación de un Serious Gaming para el reforzamiento de estudiantes de educación primaria en el área de matemática basándonos en el plan curricular del estado peruano "Programa curricular de Educación nivel Primaria" con el objetivo de presentar esta materia a los estudiantes desde un punto de vista más lúdico y de esta forma mejorar el rendimiento académico.

Para lograr este objetivo nos enfocamos en el diseño y desarrollo del Serious Gaming aplicando la metodología ágil SUM para el desarrollo del software como también la metodología de Pérez Marques para el desarrollo del concepto del videojuego.

PALABRAS CLAVE: Juegos Serios, Desarrollo de Juegos, Gamificación, Metodología SUM, Metodología de Pérez Márquez, Inteligencia lógico-matemática, Juegos Digitales, Aprendizaje basado en Juegos.

ABSTRACT

In our country there is a great concern with regard to education, specifically in the low performance of students, one of the main causes is the disinterest of students in the subjects. In order to solve this problem it is intended to use a tool that captures the interest of students. To achieve this, video games are taken as an effective means to stimulate students and at the same time educate them through Serious Gaming. For this reason the present project has the objective of the development and release of a Serious Gaming for the reinforcement of primary education students in the area of mathematics based on the curricular plan of the Peruvian state "Primary Education Curricular Program" with the objective of present this subject to students from a more playful point of view and thus improve academic performance.

To achieve this objective we focus on the design and development of Serious Gaming by applying the agile SUM methodology for software development as well as Pérez Marques methodology for the development of the concept of the video game.

KEYWORDS: Serious Games, Game Development, Gamification, SUM Methodology, Pérez Márquez Methodology.

CAPÍTULO I. EL PROBLEMA

1.1. IDENTIFICACIÓN DEL PROBLEMA

En nuestra sociedad se ha venido estableciendo, poco a poco e invisiblemente, una creciente preocupación por la educación. Ya que el nivel educativo peruano es muy bajo en comparación con otros países. En una evaluación internacional realizada el 2016 por OCDE organización que se dedica a evaluar el nivel educativo de estudiantes de distintos países que financian la educación obligatoria, encontramos el posicionamiento del Perú en un 64 puesto en educación a nivel país colocándonos en uno de los países con mayor porcentaje de estudiantes de 15 años que no superan el promedio establecido a nivel académico tanto en lectura (60%), ciencia (68%) y el segundo peor situado en matemáticas (74%). Como se presenta en el grafico siguiente (PISA, 2015).

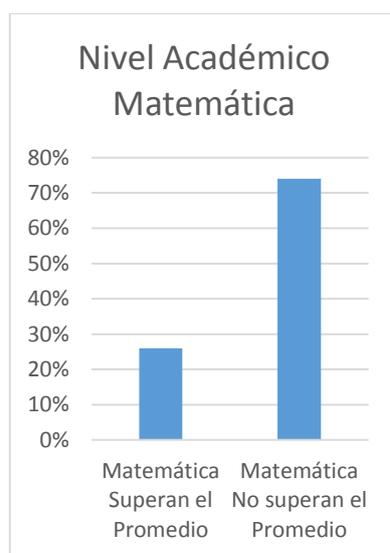


Figura 1 Nivel académico Peruano según OCDE

Fuente: (PISA, 2015)

La evaluación presentada en la figura anterior muestra el nivel educativo de los estudiantes en el área de matemática solo del grado de secundaria, el Instituto Nacional de Estadística e Informática (INEI) del Perú en el año 2016 presentó los resultados de los estudiantes de nivel primario en el área de matemática dividiéndolo en tres niveles: satisfactorio, en proceso, en inicio y previo al inicio. Como se muestra en la figura siguiente.

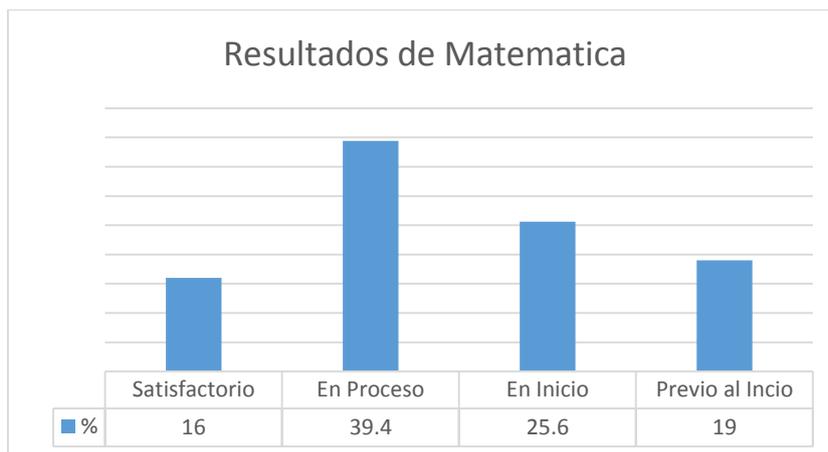


Figura 2 Nivel matemático estudiantes de Primaria

Fuente: (Ministerio de Educación Perú, 2016)

Los resultados de los estudiantes presentados en la figura anterior corresponden al sexto grado de primaria, donde se aprecia que solo el 16% de estudiantes muestra un desempeño acorde a lo que se infringe para este grado, es decir que esta cantidad de estudiantes se encontraría en posibilidades de formular y resolver problemas matemáticos involucrando los contenidos correspondientes al nivel de educación primaria, mientras que el 84% de estudiantes pertenece al total de estudiantes con limitaciones a posteriores aprendizajes. (Ministerio de Educación Perú, 2016)

Según el Ministerio de Educación Peruano en el Diseño Curricular Nacional de Educación Básica Regular *“La actividad lógico-matemática contribuye también al desarrollo del pensamiento creativo, la capacidad de análisis y de crítica, y a la formación de actitudes como la confianza en sus propias habilidades, la perseverancia en la búsqueda de soluciones y el gusto por aprender, desarrollados desde temprana edad”* (DINEIP - DINESST, 2005)

Por esta razón es importante enfocarnos en el desarrollo de una base educativa sólida en primaria, para de esta manera permitir al estudiante desenvolverse de la manera esperada en grados superiores, para lograr esto se debe identificar los factores potenciales que afecten el rendimiento de los estudiantes.

Uno de los factores que evita una buena formación educativa primaria es el desinterés de los estudiantes en las distintas materias que lleva en su carga escolar y esto sucede por distintas razones, *“Los estudiantes no encuentran utilidad a lo aprendido, las metodologías y didácticas utilizadas muchas veces no son acorde con lo que se quiere enseñar. No todos los estudiantes aprenden de la misma manera o al mismo ritmo.”* (Rafael, 2016). Esto afecta la relación entre

el estudiante y el aprendizaje cambiando la perspectiva del estudio a una obligación y no a una necesidad, ya que la perspectiva de educación de los estudiantes difiere al objetivo principal de esta que es aprender.

Como se observa en las evaluaciones realizadas por OCDE (Organización para la Cooperación y el Desarrollo Económicos) al grado de secundaria, como también el Informe de Evaluación de Matemática realizado por MINEDU (Ministerio de Educación Peruano) al sexto grado de primaria, el área a la que se le debe dar atención es de matemáticas. *“Es importante que desde la infancia se desarrolle el pensamiento lógico matemático en el niño basado en la construcción de un conjunto de competencias que le posibiliten utilizarlas en cualquier situación que se le presente ya sea escolar o no.”*(Cardoso Espinosa & Cerecedo Mercado, 2009)

Es importante considerar que el desarrollo lógico matemático desde temprana edad favorece al estudiante en el desarrollo de otras competencias, ya que esta permite el desarrollo del pensamiento analítico lo que es fundamental para desarrollar el aprendizaje significativo, que según David Ausubel *“El aprendizaje significativo se encarga de buscar vinculaciones entre los contenidos que aprendemos y los que ya habíamos aprendido”* (AUSUBEL, 1983).

1.2. JUSTIFICACIÓN

Para poder combatir la problemática presentada se realizó el desarrollo de una herramienta que estimule al estudiante de manera positiva y al mismo tiempo mejore su rendimiento académico. Por este motivo se realizó el diseño de un Juego Serio que tenga como principal objetivo desarrollar las habilidades para la resolución de problemas, la toma de decisiones, mediante el reforzamiento matemático. *“El videojuego aumenta la motivación en el aprendizaje, ayuda al alumno a adquirir conocimientos de una manera atractiva y contribuye al desarrollo de competencias.”*(Bourne & Salgado M, 2016)

A diferencia de la aplicación clásica de metodologías de educación la adición de los videojuegos (Gamificación) ofrece otra perspectiva de educación basada en función al entretenimiento:

- El desarrollo de pensamiento estratégico para soluciones complejas en distintos escenarios presentados por el software.
- El compromiso con la actividad.
- La retroalimentación inmediata.

- La simulación de roles y creación de narrativas atrapantes que nos invitan a jugar.
- El juego permite al jugador aprender a su ritmo.
- La satisfacción de completar un objetivo y obtener una recompensa.

Son elementos que al combinarlos con la currícula escolar permiten al estudiante interactuar con un escenario educativo de manera entretenida. De este modo se pretende dar un aporte a los estudiantes para que puedan rendir académicamente de manera positiva. Según David Ausubel psicólogo y pedagogo, *“el desarrollo y construcción de conocimientos a partir de información ya adquirida permite al estudiante crear conocimientos que encajen con la información vieja y la refuercen. Este tipo de construcción de conocimientos es importante para que el estudiante comprenda la lógica y secuencia por medio de su propia intuición. La práctica para la comprensión y búsqueda de soluciones de distintos problemas permiten por medio de un buen estímulo lograr captar este tipo de aprendizaje para el estudiante y esto se puede lograr de manera eficaz por medio de un videojuego”* (AUSUBEL, 1983).

Como también por medio de este software brindar a la población estudiantil un medio de entretenimiento educativo que no solo divierta y entretenga sino también estimule de manera positiva el desarrollo de habilidades y pensamiento lógico matemático.

1.3. PRESUNCIÓN FILOSÓFICA

La educación es una herramienta destinada a desarrollar la capacidad intelectual de las personas y es fundamental su práctica y estudio, por esta razón es esencial su aplicación en los estudiantes desde temprana edad. En los últimos años hemos visto una evolución de la tecnología que está siendo orientada como herramienta con distintos propósitos, viendo la tecnología como herramienta y aplicándola al panorama de la educación, podemos utilizarla como herramienta educativa. Anímense y edifíquense unos a otros, tal como lo vienen haciendo. (Filipenses 4:9)

1.4. OBJETIVOS

1.4.1. Objetivo General

“Desarrollar un Serious Gaming orientado a fortalecer las matemáticas en estudiantes de primer año de primaria aplicando la metodología SUM”

1.4.2. Objetivos Específicos

- Analizar y diseñar el concepto de videojuego en base a las competencias de matemática del primer grado de primaria según el plan curricular peruano.
- Aplicación de la metodología SUM para el desarrollo del videojuego.
- Desarrollar el Serious Gaming.
- Liberar el Serious Gaming como prueba en el Colegio Adventista Titicaca.

CAPÍTULO II. BASES TEÓRICAS

2.1. REVISIÓN LITERARIA

En el proyecto de tesis de la carrera de Ingeniería de Sistemas e Informática realizada por (REYES ROSENDO DAVID ESTEBAN, 2015) con el título de estudio, Desarrollo de un videojuego educativo 2D para plataformas Web con el motor de videojuegos HTML5 Turbulenz utilizando la metodología OOHDM. Desarrollo unos videojuegos educativos orientados para niños de 7 a 11 años de edad. Para su desarrollo utilizo el motor de videojuegos Turbulenz basado en HTML5 y JavaScript, para el análisis y desarrollo del software utilizo la metodología Huddle (metodología que sigue la analogía de SCRUM), se basa en un proceso específico para el desarrollo de videojuegos y se divide en tres fases: Preproducción, producción y Postmodern. Para la fase de diseño utilizo la metodología OOHDM. El videojuego tiene como propósito ayudar a desarrollar la coordinación óculo-manual y concentración de los niños.

En el proyecto de tesis de la carrera de Ingeniería de Sistemas realizada por (Palomino Taype, 2016) con el título de estudio, Desarrollo de un Videojuego Casual Multiplataforma para reducir el estrés. Desarrollo un videojuego usando la metodología SUM con una adaptación de la metodología GUP con el objetivo de abarcar el desarrollo usando la metodología SUM y abarcar el diseño de las interfaces usando la metodología GUM para desarrollar un videojuego causal para que reduzca el estrés de sus jugadores.

De acuerdo al Plan Curricular Peruano entregado por el ministerio de educación (Minedu, 2014) en el área competitiva de Matemáticas contempla 4 competencias que debe desarrollar el estudiante: Resuelve problemas de cantidad, Resuelve problemas de regularidad equivalencia y cambio, Resuelve problemas de gestión de datos e incertidumbre y Resuelve problemas de forma, movimiento y localización. Nos enfocaremos en la competencia de “Resuelve problemas de cantidad” basándonos en sus desempeños para el desarrollo de los requerimientos del videojuego.

Para el desarrollo de un Juego Serio es necesario contemplar todos los elementos correspondientes al desarrollo de un videojuego, ya que un Juego Serio no deja de ser un videojuego. Para el desarrollo de esta etapa del software nos basamos en el libro “Desarrollo de Videojuegos, Un enfoque Práctico” específicamente el volumen 4 que titula “Desarrollo de Componentes”.

2.2. MARCO TEÓRICO

2.2.1. Juegos Serios

Juegos serios o Serious Gaming tienen como objetivo educar o dar a entender un mensaje a sus usuarios o jugadores mediante un videojuego sin perder la esencia del mismo, en otras palabras contemplando todas las características de un videojuego pero con un propósito específico que depende del objetivo al que está orientado. Según Belén Gómez en su ponencia “Gamificación y Juegos Serios: ¿qué son y por qué funcionan?” (Gómez, 2018) la clasificación de videojuegos que nos presenta es:

Clasificación de Juegos Serios

- Motivación, juegos serios que permiten realizar actividades cotidianas de manera entretenida por medio de la motivación.
- Concienciación, por medio del videojuego brindar al usuario una experiencia directa sobre algún problema o perspectiva; los juegos serios concientizan a la población de manera no-instructiva. Permitiendo llevar el mensaje del juego de una forma distinta y eficaz.
- Práctica, los juegos serios permiten simular situaciones reales con el fin de permitir al usuario desenvolverse en un entorno seguro antes de realizar la actividad en concreto y sin ninguna experiencia práctica.
- Persuasión, permite involucrar distintos escenarios al usuario con el fin de persuadir en su toma de decisiones.
- Leasing, involucran al jugador y convierten el estudiar en algo interesante. Permitiendo al estudiante abrirse en la educación de otra manera.
- Evaluación, permite evaluar a sus usuarios mediante un juego, evitando problemas como puede ser la presión de realizar una evaluación en el usuario como también posibles errores.

Cada uno de los tipos de Serious Gaming presentados contemplan características relevantes que los diferencian entre sí con propósitos específicos, este tipo de metodología no fue creada recién en nuestro tiempo, sino que fue aplicada en distintos periodos del tiempo como por ejemplo en 1770 en Francia, una matrona diseñó un muñeco que simule un parto, con la finalidad de enseñar a nuevas matronas (Mendoza, 2015), este tipo de aplicaciones no son nuevas, sino que en nuestra actualidad su aplicación se puede realizar mediante software.

Elementos de un Serious Gaming

Un Serious Gaming independientemente de la clasificación que tenga, contempla de forma general los siguientes elementos:

Una Historia

En un videojuego serio una historia es casi necesaria, ya que le da sentido de porque se está realizando tal actividad y esto a su vez lo estimula a adentrarse en el personaje que representa en el juego e incluso en su interacción. Cuanto más sofisticado sea el argumento (y los personajes), más fácil será para los jugadores sumergirse y mayor será su motivación.

Gamificación

Un elemento importante en un Serious Gaming son las dinámicas, incluso clasificaciones, las recompensas, el distintivo o los sistemas de punto. Esta Gamificación estimula a los jugadores. Las clasificaciones también son muy importantes en los juegos serios: competencia que permite una interacción sana entre jugadores y los estimula para completar el objetivo del Serious Gaming.

Feedback

En los videojuegos el Feedback es casi inmediato, el jugador interactúa con el videojuego y al culminar recibe una recompensa o un castigo, hasta incluso en algunos. “*En los videojuegos más sofisticados, ese feedback es además detallado y argumentado: los usuarios pueden saber por qué se han equivocado e intentar hacerlo mejor la segunda vez.*”(Gamelearn, 2017). Si el feedback es bien aplicado puede ser una herramienta muy útil para la estimulación del usuario.

Simulación

Los Serious Gaming contemplan en su mayoría de casos situaciones o escenarios muy parecidos a la vida real, estos simuladores son los que permiten al usuario interactuar y practicar habilidades y conceptos dentro del juego.

El objetivo

El elemento clave de un Serious Gaming es el objetivo claro y específico para el que fue desarrollado, la finalidad principal es el objetivo con el que se ha planteado y su segunda finalidad es entretener.

2.2.2. Documento de Diseño de Videojuegos (GGD)

El Documento de Diseño (GGD) es un documento donde está especificado a detalle cada uno de los elementos que pertenecen al juego, desde detalles del guion del juego, hasta el fondo

musical correspondiente a cada etapa del mismo sin olvidarnos de las mecánicas, escenarios y elementos de interacción del jugador con el software. Este documento es creado y editado por el equipo de la mano de los diseñadores, artistas y programadores, luego es presentado al editor y al equipo de desarrollo para su aprobación. El objetivo general de este documento es servir como guía durante toda la etapa de desarrollo del software a todo el equipo de desarrollo del software.

Ciclo de Vida

El ciclo de vida de un GGD dura todo el desarrollo del software. Este se va actualizando de acuerdo a los avances del proyecto y acontecimientos que surgen entorno a este. Esto sucede debido al entorno dinámico en el desarrollo de un videojuego, por esta razón es considerado como documento viviente. Un documento de diseño puede empezar como un simple boceto general y terminar como una lista detallada de cada aspecto del videojuego.

Contenido

La estructura del Documento de Diseño varía de acuerdo al tipo de videojuego que se quiera desarrollar, pero existen formatos generales de la estructura principal que debe tener el documento. Los elementos que puede contener un Documento de Diseño son: texto, imágenes, diagramas, arte conceptual y cualquier otro elemento multimedia que pueda ilustrar de mejor manera las decisiones de diseño.

Estructura

La estructura de un videojuego varía de acuerdo al tipo de juego que se desea realizar y a los elementos que contemplara. *“Los elementos generales que se consideran en cualquier tipo de juego son: Historia, Personajes, Niveles/desarrollo del entorno, Jugabilidad, Arte, Sonidos y Música, Interfaz del Usuario y Accesibilidad”*(Tim Ryan, n.d.). Los otros elementos se van agregando por el diseñador de acuerdo a las necesidades del software pero con suficiente detalle para que el equipo de desarrolladores pueda implementarlo sin ninguna dificultad. Cabe considerar que cada elemento general no es absolutamente necesario definir en el documento de diseño; algunas de esas secciones podrían no ser parte del GGD, todo de acuerdo a las necesidades para el desarrollo del software.

2.2.3. Metodología de Desarrollo SUM

Una metodología de desarrollo es un marco de trabajo que permite estructurar, planificar y controlar el desarrollo de un software. No existe una metodología de desarrollo que se remarque sobre otras para el desarrollo de un software, todos pueden utilizar cualquier metodología si es

útil para lograr su proyecto e incluso utilizar distintas metodologías para contemplar distintos escenarios del proyecto.

La metodología SUM es una metodología basada en SCRUM, que utiliza la tendencia de metodologías ágiles debido al contexto con el que se trabaja: programación rápida, precisa y optimizada. Esta metodología orientada para el desarrollo de videojuegos tiene como objetivo desarrollar un software de calidad en tiempo y costo, así como la mejora continua del proceso para aumentar su eficiencia y eficacia. El objetivo principal es:

- Obtener resultados predecibles.
- Administrar eficazmente los recursos y riesgos del proyecto.
- Lograr una alta productividad del equipo de desarrollo

SUM fue creada con el fin de que se adapte a proyectos de pequeña envergadura y de corto tiempo con alto grado de participación del cliente.

El proceso de desarrollo se divide en cinco fases iterativas e incrementales que se ejecutan de forma secuencial con la excepción de la fase de gestión de riesgos que se lleva a cabo a lo largo del proyecto. Las cinco fases secuenciales son:

Concepto

En esta fase se considera todas las ideas o propuestas para el videojuego, como también se crea el Documento de diseño (High Concept) donde se explica a detalle cada una de las características del videojuego con la finalidad de contemplar desde una perspectiva panorámica todo lo que se plantea desarrollar. Otro documento que también se contempla en esta etapa es El documento de Diseño donde se considera el arte conceptual, storyboard, sonido, programación y mecánicas del juego

Planificación

En esta etapa se considera los alcances que tendrá nuestro videojuego de acuerdo al presupuesto que se tiene o herramientas con las que se cuenta para el desarrollo. Se evalúan los costos, tiempos, etc (planificación administrativa) como también cantidad de personal, herramientas necesarias para el desarrollo y contemplaciones extras. Y de acuerdo a lo anteriormente acordado se realiza un cronograma de actividades por cada fase que contempla todo el desarrollo del software.

Elaboración

En esta fase se empieza con el desarrollo del software en base a todas las fases anteriores mencionadas. Pero también considerando la parte esencial de la metodología SUM que es el desarrollo de entregables por etapas, comunicación constante en el equipo y adaptación a distintos cambios que existen en el proyecto.

Beta

Esta etapa empieza con la finalización del software versión beta, se realiza un testeo de todas las funcionalidades que ofrece el juego y se verifica su buen rendimiento en todos sus aspectos, también se realiza pruebas de Jugabilidad para evaluar e interactividad. Si existe algún problema el juego retorna a la fase 3 de elaboración y se realizan las correcciones necesarias.

Cierre

Culminando la fase beta el videojuego pasa a su liberación donde se presenta al público al que va orientado y se realiza una evaluación de su interacción con el público para futuras consideraciones,

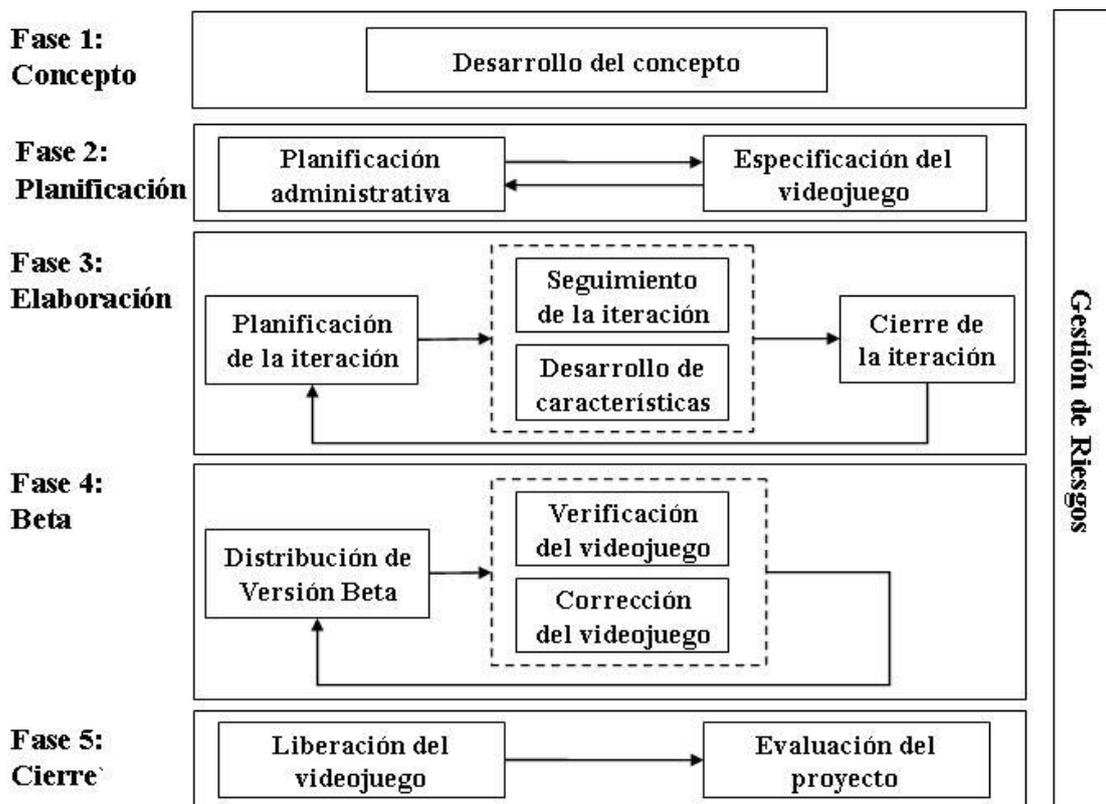


Figura 3 Estructura Metodología SUM

Fuente: (Gemserk)

2.2.4. ¿Porque la metodología SUM?

Para la metodología de desarrollo consideramos distintas metodologías que nos permitan desarrollar nuestro software ordenadamente como también flexible para su desarrollo, basándonos directamente en los métodos de trabajo, organización, seguimiento de desarrollo y tiempo de desarrollo para nuestro software.

Tabla 1

Metodologías de desarrollo Viabes para el desarrollo de Juegos

Proceso	Descripción	Valoración
Waterfall Process	El proceso de desarrollo de software secuencial especializado en el que el desarrollo se basa en el modelo Cascada a través de fases de diseño, iniciación, análisis, dibujo, construcción y prueba.	El producto final se demora más de lo esperado, ya que cualquier problema que surja tiene que representarse en uno de los pasos del anterior para corregirlo. Requiere hacer muchos cambios en los documentos y volver a los pasos anteriores promueve un proceso muy complicado.
Rational Unified Process	Es un proceso de desarrollo de software Iterativo, es adaptable y apto para satisfacer las necesidades del equipo del proyecto. Es comúnmente seguido por una metodología pesada.	Un proceso pero no esta enfocado al desarrollo de videojuegos. Demasiada documentación permite realizar un buen producto con lo que ello conlleva (tiempo, dinero y personas involucradas)
Essential Unified Process	Consiste en componer las buenas prácticas que recurring a los tres campos principales del proceso: el campo del proceso	EssUp genera un buen trabajo en equipo y está preparado para separar el trabajo creativo del mecánico. Separa los

	<p>unificado, de los métodos ágiles y el campo de la madurez del proceso. Apoya a cada uno de ellos con diferentes habilidades. EssUP se enfoca en prácticas esenciales que creen que deben tener todos los proyectos de desarrollo de software</p>	<p>engranajes en un alfa y un trozo de cuerda, siendo el alfa más importante en el proyecto. Es posible separar las prácticas del proceso y adaptarlas al desarrollo de los videojuegos.</p>
SUM	<p>SUM es una metodología basada en la metodología Scrum pero adaptada al desarrollo de videojuegos que contemplan la calidad en tiempo y costo y la mejora continua. Busca lograr resultados predecibles, administrar los recursos y riesgos del proyecto con competencia y obtener una alta productividad del equipo de desarrollo.</p>	<p>Es un software es muy adecuado para el desarrollo de un videojuego, pero cada paso de la documentación es específicamente muy genérico para el desarrollo de un videojuego educativo.</p>
OpenUp	<p>OpenUP es un proceso mínimo y suficiente, lo que significa que solo se incluye el contenido básico y necesario. Se informa que la mayoría de los elementos de OpenUP promueven el intercambio de información entre los equipos de desarrollo y mantienen un</p>	<p>La descripción del proceso es un gran enfoque en la colaboración del equipo y la inclusión de partes interesadas como parte del proceso. De alguna manera es similar al scrum, a diferencia de los participantes del scrum, no se comunican a las reuniones</p>

acuerdo compartido del proyecto, sus objetivos, alcance y avances.

diarias para comentar sobre el estado del proyecto. Sin embargo, hay muy poca información sobre el proceso para separarlo de otros cambios en el proceso unificado, pero la filosofía del trabajo duro en un equipo es esencial para el desarrollo de los videojuegos.

Fuente Basada en: (Abraham et al., 2010)

Las metodologías presentadas en la tabla anterior presentan ventajas y desventajas con respecto a su aplicación en el desarrollo de un software. Para el desarrollo de este videojuego se aplicara la metodología SUM que es una metodología flexible contempla aspectos como la calidad en costo, tiempo, mejora continua y está desarrollado para proyectos que contemplan un desarrollo máximo de un año. La metodología SUM contempla las etapas de desarrollo de un videojuego pero no contempla el aspecto educativo de un Serious Gaming, que en este proyecto está orientado a la educación. Por esta razón es necesaria una metodología que pueda abarcar la idea pedagógica que tiene que contemplar nuestro videojuego educativo, para esto aplicaremos una metodología de desarrollo educativo con el fin de integrarla a la metodología SUM.

2.2.5. Metodología de Desarrollo de Videojuegos Educativos de Pérez Marques

En esta metodología, el carácter didáctico es fundamental a la hora de dibujar o producir un software para fines educativos, ya que el mismo flujo se basa o se basa en una serie de pasos que permiten orientar y alcanzar los objetivos. Enseñar y aprender procesos propios sobre el hecho educativo. . Estamos buscando el software para facilitar los procesos de enseñanza y aprendizaje. En este sentido, Pere Maques realiza un ranking de software educativo en tutores, programa una base de datos y programa simuladores. Un marqués plantea un ciclo de desarrollo para un software educativo, una descripción detallada de las actividades y los recursos necesarios para cada uno de ellos. Esta metodología consiste en que señala el eje de la construcción de programas educativos en el equipo pedagógico, otorgándole el papel de líder.

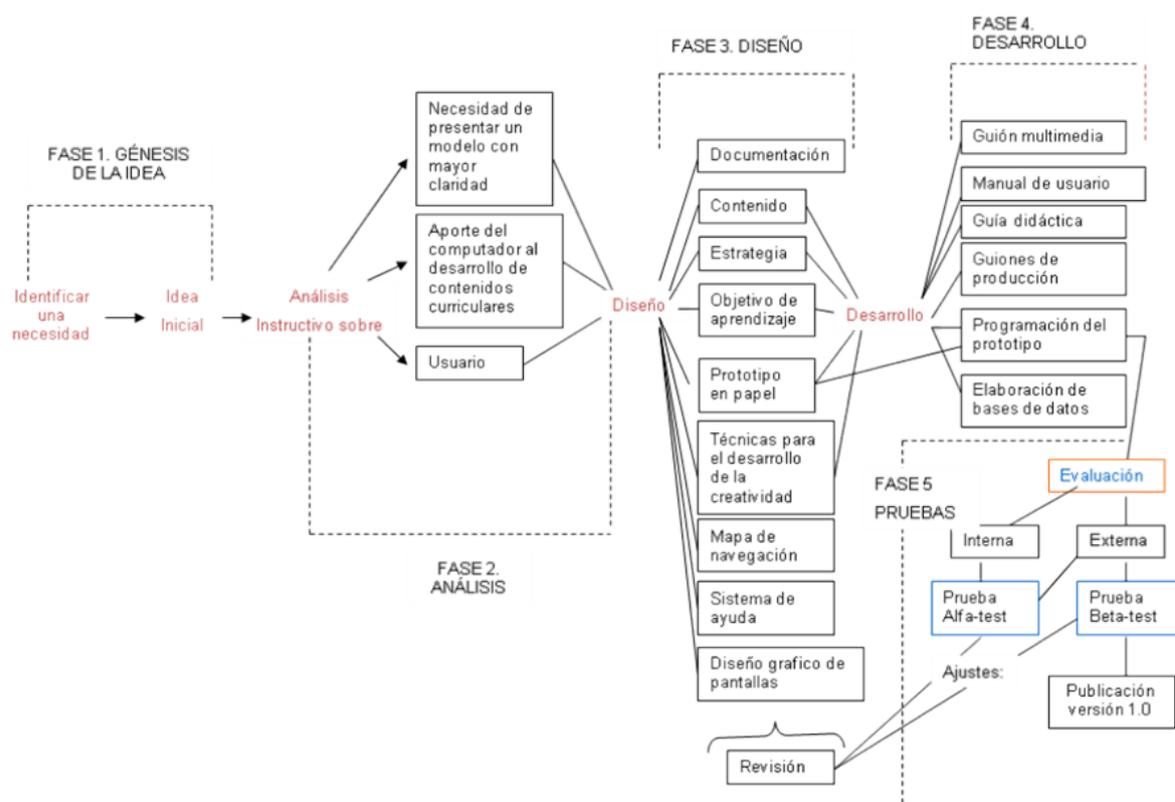


Figura 4 Metodología de Pérez Marqués para el desarrollo de software Educativo

Fuente: (Nuñez Wagner, 2014)

Etapas

- Definición del problema y análisis de necesidades: Antes de dibujar un material de aprendizaje multimedia, es necesario definir los objetivos a los que se apunta, un tipo de público para el que está orientado su desarrollo, sus características de infraestructura.
- Génesis de la idea: Es una fase de inicio, de factibilidad o de preproducción, que busca detectar de manera abrumadora una situación problemática que necesita ser resuelta con ayuda de computadora, para lo cual utiliza diferentes mecanismos que permiten argumentar el desarrollo de un software o utilizar uno ya existente.
- Diseño Instructivo: Incluye el primer documento de trabajo que parte de la idea original, es el primer guion en el que se centrará en los fundamentos pedagógicos del proyecto a desarrollar (contenidos, objetivos, estrategias, entre otros)
- Estudio de Viabilidad y Marco General del Proyecto: En esta fase se determina la factibilidad del proyecto a realizar, aquí se consideran los siguientes aspectos: pedagógico, funcional, técnico, económico y anuncios.

- **Guion Multimedia:** En esta fase se determina la viabilidad del proyecto a realizar, en esta etapa se toman la consideración los siguientes puntos: pedagógico, funcional, técnico, económico y comercial, etc.
- **Creación de los contenidos:** Se subdivide en: contenidos y documentación. La misión del grupo de especialistas en materias especifica el programa y los técnicos en un MULTIMEDIA. Además de eso, se considerará como la base de datos principal y complementaria del Programa. La documentación, será en nombre de los profesores, el especialista en temas del programa y los especialistas en dibujo informativo y materiales de enseñanza.
- **Elaboración del prototipo ALFA-Test:** Los profesionales, programadores y especialistas del Grupo de Especialistas en Multimedia desarrollan el primer prototipo interactivo del equipo.
- **Evaluación Interna:** Esta etapa es desarrollada por los miembros del equipo de dibujo y desarrollo del Material. Se desarrollará aplicando la metodología definida para estos tipos de materiales considerando los criterios de calidad establecidos.
- **Elaboración de la versión BETA-Test:** Después de las conclusiones internas e hizo los ajustes apropiados en el dibujo, una base de datos y un programa interactivo, el material se somete a una prueba técnica severa para purificar los posibles problemas de operación debido a errores de programación.
- **Evaluación Externa:** La evaluación externa de la versión beta-test del programa, se dará cuenta, las personas ajenas al equipo que participaron en su dibujo y desarrollo.
- **Versión final 1.0:** A partir de los resultados de la evaluación externa, se realizan los últimos ajustes en el hardware y la versión 1.0 del programa.

2.2.6. ¿Porque la metodología de Pérez Marques?

Existen varios tipos de metodologías orientadas a la educación de las cuales las 4 más usadas son: la metodología de Pérez Márquez, Metodología de Álvaro Galvis, Metodología de Brian Blum y Metodología de Thales. Andrea Elizabeth Carranza Infante Y Marisol Claudia Nuñez Wagner en su artículo titulado “Metodologías de desarrollo de software educativo” (Nuñez Wagner, 2014) Realiza una comparación de fortalezas y debilidades de estas metodologías en la siguiente tabla.

Tabla 2
Metodologías para el desarrollo de Software Educativo

Autor	Fortalezas	Debilidades
Pérez Márquez	<ul style="list-style-type: none"> • Ofrece pautas para elaboración de Evaluaciones, pruebas piloto y pruebas de campo. • El análisis es profundo en cada etapa de la metodología: curricular, instrucciones, necesidades de usuarios e Instituciones, entre otros. • Prevé documentación del proceso en cada una de sus fases. Contempla la ficha resumen que recoge las principales características del programa, el manual de usuario y una guía didáctica. 	<ul style="list-style-type: none"> • Las fases de su metodología de desarrollo de software educativo se entremezclan con las fases de diseño instruccional del material. • Al tener un nivel de detalle muy alto, se tiende a confundir las actividades presentes en cada fase, ocasionando dificultades para su aplicación. • Peso muy alto en el aspecto instruccional y de contenido, disminuyendo la importancia de los aspectos computacionales, lo cual podría generar productos de baja calidad de acuerdo con estándares internacionales. • No prevé evaluaciones al final de cada etapa del desarrollo.
Álvaro Galvis	<ul style="list-style-type: none"> • Posee un fuerte basamento empírico (experiencias previas). • Propone la incorporación de grupos de trabajo multidisciplinario para el desarrollo del proyecto 	<ul style="list-style-type: none"> • Desde la perspectiva computacional no ha avanzado, por lo cual es necesario enriquecerla tomando en cuenta los avances tecnológicos en el diseño y desarrollo

Brian Blum	<ul style="list-style-type: none"> • Fácil aplicación y desarrollo de la Metodología de trabajo. • Ofrece pautas precisas para elaboración de evaluaciones por expertos, pruebas piloto y pruebas de campo. • Las actividades de aprendizaje constituyen una relación conexas entre el diseño educativo y el diseño interactivo. • El análisis es profundo en cada uno de los aspectos del producto. • Realización de prototipos en papel. • Emplea herramientas de documentación, control y organización en cada una de las fases del proceso. • Proporciona apoyo al docente. • Involucra un equipo multidisciplinario. 	<p>computacional que se han logrado en los últimos años.</p> <ul style="list-style-type: none"> • No se contempla una secuencia estricta para la realización de las actividades que constituye cada una de las fases de análisis y diseño. • No prevé evaluaciones al final de la fase de análisis. • Se limita al uso del modelo de aprendizaje cognoscitivo para el diseño de los materiales. • No se especifica el modelo de prueba final (Usuarios, aspectos a evaluar, etc.) • No existen revisiones continuas al final de cada una de las etapas con el objetivo de verificar que se cumpla con todos los elementos que el cliente requiera.
Thales	<ul style="list-style-type: none"> • Hay una Integración de la tecnología al currículo. • Orientan aprendizajes. • Propicia actividades de diversa índole. 	

Fuente Basada en: (Nuñez Wagner, 2014)

Considerando las fortalezas y debilidades de cada metodología y utilizando una serie de criterios podemos hacer la elección de la más óptima para el desarrollo del software educativo. En este caso la metodología más apta escogida para el desarrollo de este software fue la

metodología de Pérez Marques porque permite estructurar todo el diseño del software en base a un enfoque pedagógico, además propone el trabajo multidisciplinario de todo el equipo de desarrollo del software incluyendo al cliente. Claro que es importante considerar sus desventajas que contempla esta metodología como el no proveer evaluaciones al final de cada etapa de desarrollo, pero estas se compensan con la metodología SUM.

Por esta razón se hizo uso de la Metodología SUM para el desarrollo de este videojuego educativo contemplando algunas etapas de la Metodología de Desarrollo de Videojuegos Educativos de Pérez Marques. Con el fin de realizar el diseño del software desde un enfoque pedagógico considerando todos los elementos relacionados a la educación que requiere nuestro software y al mismo tiempo desarrollar un software flexible y de calidad en tiempo y costo.

2.2.7. Godot Engine como motor de Videojuego

Que es un Motor de Videojuego

Un motor de videojuegos, es un concepto referido a una serie de rutinas que se pueden diseñar y representar en un videojuego. Su funcionalidad básica es proveer al videojuego de un motor de renderizado para los gráficos 2D - 3D, escenario, motor físico o detector de colisiones, animación, inteligencia artificial, redes, gestión de almacenes de drenaje y escenario gráfico. El proceso de desarrollo de un videojuego puede variar de acuerdo al tipo de videojuego que se quiera realizar y los elementos que lo componen, y se adapta a la creación del mismo motor de videojuegos o a otros juegos reutilizados.

En la actualidad tenemos motores de gráficos completos, el Kit de desarrollo de software (SDK) es un gran desarrollador de motores o es capaz de aprender sobre videojuegos para los usuarios interesados en este desarrollo y, por lo tanto, tiene un gran desarrollo de Introducción y ejemplos. En definitiva los motores de videojuegos son herramientas que permiten al desarrollador centrarse única y directamente en el desarrollo de su juego y su concepto, evitando la producción código genérico básico que ya la brinda el motor de desarrollo.

Godot Engine

Existen distintos tipos de motores de desarrollo y cada uno contempla una estructura distinta, Godot Engine es un motor de videojuegos multiplataforma con varias funciones que le permite crear juegos 2D y 3D desde una interfaz unificada. Ofrece una extensa fila de herramientas flexibles que permiten al usuario adaptarlas de acuerdo a sus necesidades de desarrollo, diseño y apariencia del videojuego, tanto en la creación del juego como en la exportación a distintas plataformas con un solo clic. Las plataformas más importantes que soporta Godot Engine son:

- Escritorio: Linux, macOS, Windows.
- Móviles: Android, Ios.
- Web: HTML5.

“Godot es completamente gratuito y de código abierto bajo la licencia rápida con licencia MIT. Sin condiciones, sin royalties, nada. Los juegos de los usuarios están contigo, hasta la última línea de los códigos de motor”(Liniesky & Manzur, n.d.). El desarrollo de Godot Engine es completamente independiente y está impulsado por la Comunidad de Godot, de modo que los usuarios pueden ayudar a su motor a adaptarse a sus expectativas.

“Godot adopta un diseño orientado a objetos desde el núcleo con su sistema flexible de escenas y jerarquía de nodos. Esto intenta mantenerlo lejos de patrones de programación estrictos para ofrecer un modo intuitivo de estructurar su juego.” (Linietsky, Manzur, & Godot, n.d.) Godot Engine trabaja con escenas que puedes crear, reutilizar y condicionar de acuerdo a las necesidades del desarrollador. Un ejemplo claro sería la figura N, existe una escena que se llama personaje y tiene las características de: mover y saltar; también existen otras dos escenas con los nombres: jugador y NPC, que contemplan características diferentes entre sí, pero heredan de la escena personaje sus características: mover y saltar.

Cada proyecto de Godot contempla una estructura de árbol que está formado por nodos que se pueden dividir en otros nodos. Un nodo es un tipo de objeto con un propósito específico y estas a su vez pueden heredar de sus nodos padre.

El lenguaje de programación que maneja Godot es GDScript un lenguaje dinámico que tiene la sintaxis de Python, como la mayoría de lenguajes dinámicos es más fácil de aprender, escribir y compilar. Pero Godot también soporta el lenguaje de programación C++, permitiendo codificar con ambos lenguajes sin alterar el resto del juego.

2.2.8. ¿Porque Godot Engine?

Para el desarrollo de este software educativo se tomó en consideración dos distintos motores de desarrollo de videojuegos; los cuales son Godot Engine y Unity, se escogieron estos dos motores de desarrollo basándonos en la gran comunidad activa que brinda tutoriales guías y modelos de desarrollo que facilitan su uso y aceleran el aprendizaje en estas herramientas.

La siguiente tabla presenta distintas características que contempla cada motor de desarrollo, se escogerá uno de estos motores de acuerdo a las necesidades que tendremos para el desarrollo de nuestro software.

Tabla 3
Motores de Desarrollo

	Godot Engine	Unity Engine
Licencia	OpenSource (Licencia MIT)	- Versión Gratuita Limitaciones - Versión de Pago
OS (editor)	C#, GDScript.	C# y una forma de JS
Nivel de Lenguaje	Alto	Regular
OS (exportar)	- Escritorio: Windows, macOS, X11 - Móvil: Android, iOS - Web: WebAssembly - Consolas: Ver Console support in Godot - VR: Oculus Rift, SteamVR	- PC: Windows, Linux, macOS - Movil: Android, iOS, Windows Phone, Tizen - Web: WebAssembly o asm.js - Consolas: PS4, PS Vita, Xbox One, Xbox 360, Wii U, Nintendo 3DS - VR: Oculus Rift, SteamVR, Google Cardboard, Playstation VR, Gear VR, HoloLens - TV: Android TV, Samsung SMART TV, tvOS
2D – 3D	- El motor de desarrollo en 2D es independiente al de 3D - Sistema de Escenas - Canalización de Animaciones	- Unity es un motor de desarrollo para juegos 3D adaptado para el desarrollo de juegos 2D - Componente/Escena (GameObject > Componente)
Sistema de Escenas	- Shaders fáciles de escribir - Depuración en el dispositivo	- Prefabs
Funciones destacadas	- Árbol de escenas y nodos	- Gran comunidad - Gran almacén de recursos

Fuente Basada en: (Liniesky & Manzur, n.d.)

El software a desarrollar contemplara las siguientes características con respecto a la interacción con el motor de desarrollo, estas servirán para escoger el motor de desarrollo adecuado para este proyecto.

Tabla 4
Características de Software para el Motor de Videojuego

Genero	Árcade
Sistema Operativo	Móvil Android
Lenguaje del Motor	Alto Nivel (muy intuitivo)
Estilo	2D
Motor de desarrollo	Libre

Fuente: Elaboración Propia

Las características presentadas en la tabla anterior que tiene que contemplar el Motor de Videojuego se enfoca simplemente en el sistema operativo al que ira dirigida y si el motor nos brinda todos sus servicios de manera libre. El motor que será utilizado para el desarrollo del software será Godot Engine basándonos en su licencia Open Source, motor de videojuegos enfocado para el desarrollo de juegos 2D, lenguaje de programación de alto nivel y a diferencia de Unity, Godot Engine consume menos recursos de la pc y ocupa menos espacio.

CAPÍTULO III. MATERIALES Y MÉTODOS

3.1. DESCRIPCIÓN DEL LUGAR DE EJECUCIÓN

Este videojuego educativo está orientado a estudiantes de escuelas de primero de primaria que se ubiquen en una zona urbana. Ya que el desarrollo del concepto del videojuego tiene como base el Programa curricular Educación Peruana. Pero como caso de estudio será aplicada en el Colegio Adventista Titicaca.

3.2. MATERIALES E INSUMOS

Tabla 5
Materiales e Insumos para el desarrollo del software

Preproducción		
Herramientas	Descripción	Costo en S/
Balsamiq	Desarrollo de prototipos (Licencia académica)	-
Lucidchart	Herramienta de diagramación basada en la web	Libre
Godot Engine	Motor de Videojuego para el desarrollo del software	Licencia MIT
Gimp	Herramienta para edición de imágenes digitales en forma de mapa de bits.	Licencia GNU
Microsoft Paint	Programa editor de imágenes de windows	-
MagicaVoxel	Editor de arte voxel, renderizador interactivo de ocho bits	Libre
Trello	Software de administración de proyectos con interfaz web.	Para uso privado gratuito

Fuente: Elaboración Propia

3.2.1. Lenguaje de Programación

El lenguaje de programación utilizado para el desarrollo del videojuego es GDScript, este es un lenguaje nativo contempla la sintaxis de Python por lo que es dinámico.

Sus grandes ventajas son:

- El idioma es sencillo y fácil de entender.
- El código escrito puede ser fácilmente reemplazado sin complicaciones.

- Su sintaxis permite que las líneas de código sean cortas lo que reduce los posibles errores.
- No es necesario compilar para probarlo.

Fundamentos físicos Teóricos del Motor de Videojuego

Godot Engine hace uso del motor de físicas Bullet que simula detección de colisiones y la dinámica de cuerpos blandos y rígidos dentro del mismo motor, estos son algunos de sus algoritmos de detección de colisión.

- Efecto del impulso sobre la velocidad: El calculo de un impulso en un solo cuerpo se puede calcular con la siguiente formula.

$$Impulse = F \Delta t = m\Delta v$$

$$Impulse_{torque} = \tau\Delta t = I\Delta\omega$$

$$\Delta v = \frac{Impulse}{m}$$

$$\Delta\omega = \frac{Impulse_{torque}}{I} = \frac{r \times Impulse}{I}$$

Hay un componente lineal y un componente angular, que implica calcular un producto cruzado entre el vector de impulso y el vector de posición relativa del punto de contacto con el centro de masa, r. (Coumans, 2015)

Resolver una sola colisión: Aquí está la ecuación completa para calcular el impulso de contacto.

$$P_{correction} = M_{effective} \Delta v_{desired} \cdot n$$

$$P_{correction} = \frac{-\Delta_{ab} \cdot n}{n \cdot n \left(\frac{1}{M_a} + \frac{1}{M_b} \right) + \left(\frac{r_a \times n}{I_a} \right) \times r_a + \left(\frac{r_b \times n}{I_b} \right) \times r_b}$$

```

1. b3Scalar resolveCollision(LWRigidBody& bodyA,
2.                          LWRigidBody& bodyB,
3.                          LWContactPoint& contactPoint) {
4.     b3Assert(contactPoint.m_distance<=0);
5.     btScalar appliedImpulse = 0.f;
6.     b3Vector3 rel_pos1 = contactPoint.m_ptOnAWorld -
bodyA.m_worldPose.m_position;
7.     b3Vector3 rel_pos2 = contactPoint.m_ptOnBWorld -
bodyB.getPosition();
8.     btScalar rel_vel =
contactPoint.m_normalOnB.dot(bodyA.getVelocity(rel_pos1) -
bodyB.getVelocity(rel_pos2));
9.     if (rel_vel < -B3_EPSILON) {
10.         b3Vector3 temp1 = bodyA.m_invInertiaTensorWorld *
rel_pos1.cross(contactPoint.m_normalOnB);
11.         b3Vector3 temp2 = bodyB.m_invInertiaTensorWorld *
rel_pos2.cross(contactPoint.m_normalOnB);
12.         btScalar impulse = -(1.0f + gRestitution) * rel_vel
/
13.         (bodyA.m_invMass + bodyB.m_invMass +
contactPoint.m_normalOnB.dot(temp1.cross(rel_pos1) +
temp2.cross(rel_pos2)));
14.         b3Vector3 impulse_vector = contactPoint.m_normalOnB
* impulse;
15.         appliedImpulse = impulse;
16.         bodyA.applyImpulse(impulse_vector, rel_pos1);
17.         bodyB.applyImpulse(-impulse_vector, rel_pos2);
18.     }
19.     return appliedImpulse;
20. }
21.
22. if (contactPoint.m_distance<0) {
23.     m_bodies[0]->computeInvInertiaTensorWorld();
24.     m_bodies[1]->computeInvInertiaTensorWorld();
25.     b3Scalar appliedImpulse =
resolveCollision(*m_bodies[0], *m_bodies[1], contactPoint);
26. }

```

En el código fuente de C ++ se ve así. (Coumans, 2015)

3.3. METODOLOGÍA

3.3.1. Tipo de Investigación

El tipo de investigación es una investigación propositiva y aplicada, porque propone la aplicación de un Serious Gaming como herramienta para reforzar el desarrollo académico de los estudiantes de manera lúdica y a su vez esta propuesta es aplicada para considerar si la aplicación de esta estrategia permita mejorar el desarrollo matemático de los estudiantes.

Propositiva: Porque la investigación propone la aplicación de una herramienta de solución que aporte al desarrollo de los estudiantes en el área de matemática.

Aplicativa: Porque la investigación da respuesta a objetivos específicos.

3.3.2. Arquitectura Solución

La arquitectura solución tiene el siguiente flujo, todo el software está desarrollado en base a la metodología SUM, tomando en cuenta esto desarrollaremos la primera fase de la metodología “Concepto” en base a las tres primeras etapas de la metodología de Pérez Marques: Definición del Problema y Análisis de las Necesidades, Génesis de la Idea y Diseño Instructivo. Ya que estas tres primeras etapas tienen como propósito desarrollar el concepto del software en base a nuestros objetivos educativos.

Para el desarrollo del software usaremos el motor de videojuegos Godot Engine en base a la metodología SUM, posteriormente aplicaremos el software en los estudiantes de primer grado de primaria y por último se realizara una encuesta a los estudiantes para evaluar su nivel de satisfacción con el software.

3.3.3. Diseño de la Investigación

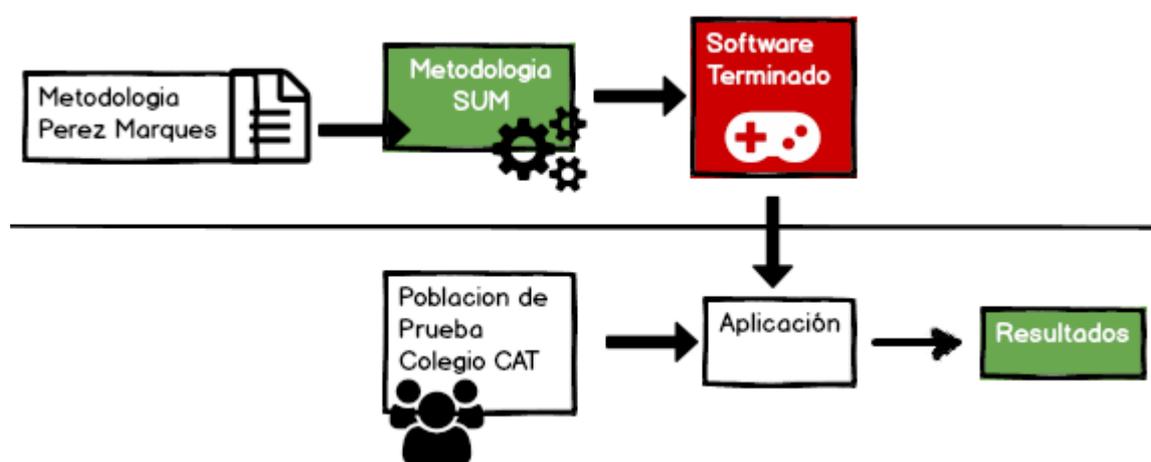


Figura 5 Diseño de la Investigación para Serious Gaming

Fuente: Elaboración Propia

3.4. DESARROLLO

Para el desarrollo del videojuego educativo se realizó la adaptación de dos metodologías: la metodología SUM y la metodología de Pérez Marques. Con el propósito de contemplar las características educativas que tiene que tener el software se aplicó dentro de la fase “Concepto” de la metodología SUM las tres primeras etapas de la metodología de Pérez Marques: Definición del problema y análisis de resultados, Génesis de la idea y Diseño instructivo. Estas tres primeras etapas definirán los elementos educativos que contemplara el videojuego y de los

que nos basaremos para el desarrollo de la mecánica del juego. La siguiente figura presenta la adaptación de ambas metodologías.

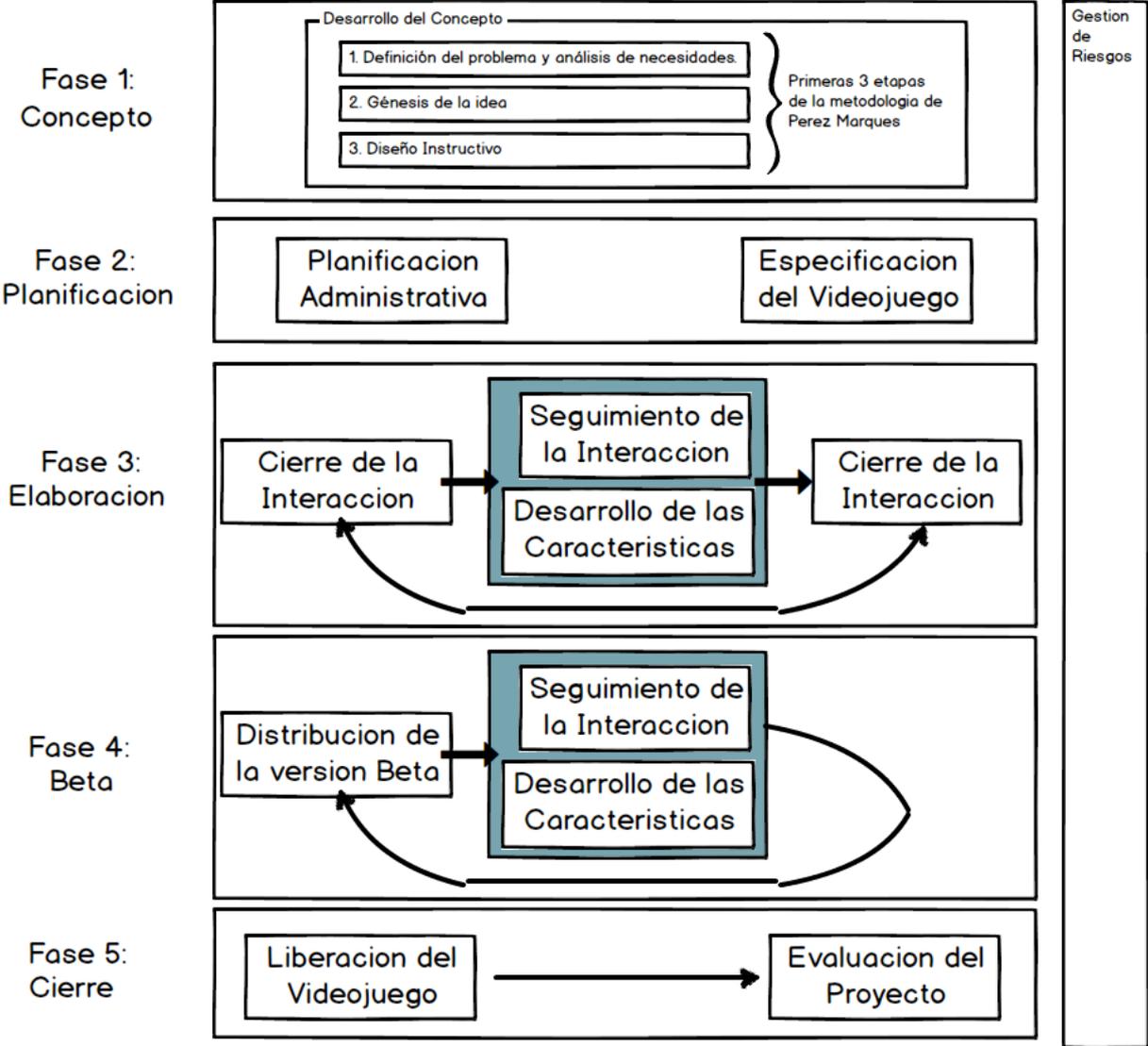


Figura 6 Metodología de desarrollo para el software

Fuente: Elaboración Propia

3.4.1. CONCEPCIÓN

Haciendo uso de las tres primeras etapas de la metodología de Pérez Márquez identificaremos los objetivos educativos del Juego Serio, estas competencias que serán elementos en los que se basaran todo el diseño del concepto del videojuego.

Definición del Problema

Para poder reforzar a los estudiantes del grado de primero de primaria es necesario buscar un apoyo que permita un desarrollo más desenvuelto en la resolución de problemas del área de

matemáticas. Para lograr esto es necesario que el estudiante cuente con un estímulo que le permita ver el aprendizaje desde un punto de vista positivo, este estímulo tiene que brindar al estudiante un escenario en el que no se sienta presionado y pueda permitirle experimentar a través de la lógica y conceptos matemáticos (aprendidos en clase) posibles soluciones a problemas con el objetivo de que el estudiante aprenda a través de la práctica.

Génesis de la Idea

Para identificar de manera más precisa las necesidades de los estudiantes desarrollamos las preguntas de la siguiente tabla con un docente de matemática. Para de esta manera abordar el pre diseño del Juego Serio.

Tabla 6
Identificación de Necesidades para el Juego Serio

Identificación de Necesidades			
Preguntas Orientadas	Respuestas	Viable usar videojuegos	Ventajas / Desventajas
¿Qué problemas de aprendizaje ha observado?	Los estudiantes tienen dificultad en matemática por la falta de práctica en la materia.	Si se puede aplicar	El poco conocimiento o práctica en los primeros años, desencadena un desinterés por la materia en futuros niveles educativos.
¿Cuáles son las competencias que más cuesta desarrollar?	Identificación de números ascendentes y descendentes con valores no continuos. Resolución de Operaciones matemáticas	Si se puede aplicar	Si se aplica de manera correcta el estudiante podrá jugar y aprender al mismo tiempo.
¿Qué temáticas son las más difíciles de apropiar?	Resolución de Operaciones matemáticas (Suma, Resta, etc.)	Si se puede aplicar	De esto depende que el juego tenga éxito o no.
¿Cuáles son las habilidades que más cuesta desarrollar?	Buscar soluciones a problemas por medio del planteamiento mental y lógico.	Contempla dificultad para su viabilidad.	Permite comprender como dar solución a los problemas matemáticos
¿Qué aspectos del proceso de aprendizaje desearía cambiar?	Resolución de Ejercicios matemáticos	Si se puede aplicar	Permite desarrollar la lógica y habilidad para resolución de problemas.

¿Qué actividades son las menos llamativas para los estudiantes?	Resolución de operaciones matemáticas suma y resta	Si se puede aplicar	Resulta tedioso y monótono para los estudiantes resolverlo en sus casas.
---	--	---------------------	--

Fuente Basada en: (Figueredo & Sosa, 2016)

A partir de la tabla anterior se determinó con el equipo de desarrollo las necesidades educativas que debe contemplar el Juego Serio.

Diseño Instructivo

De acuerdo al plan curricular de nivel primario, el área de matemática del primero de primaria contemplan las siguientes competencias:

- Resuelve problemas de Cantidad.
- Resuelve problemas de regularidad y cambio.
- Resuelve problemas de gestión de la incertidumbre.
- Resuelve problemas de forma, movimiento y localización.

De estas competencias este Juego Serio se orientara a “Resuelve problemas de Cantidad” esta competencia según el programa curricular peruano “*Consiste en que el estudiante solucione problemas o plantee nuevos problemas que le demanden construir y comprender las nociones de número, de sistemas numéricos, sus operaciones y propiedades*”. (Minedu, 2016)

Esta competencia realiza los siguientes desempeños:

Tabla 7

Desempeños de la competencia Resuelve Problemas de Cantidad

Desempeños de la competencia Resuelve Problemas de Cantidad
- Establece relaciones entre datos y acciones de agregar, quitar y juntar cantidades, y las transforma en expresiones numéricas (modelo) de adición o sustracción con números naturales hasta 20.
- Expresa con diversas representaciones y lenguaje numérico (números, signos y expresiones verbales) su comprensión de la decena como grupo de diez unidades y de las operaciones de adición y sustracción con números hasta 20.
- Expresa con diversas representaciones y lenguaje numérico (números, signos y expresiones verbales) su comprensión del número como ordinal al ordenar objetos hasta el décimo lugar, del número como cardinal al determinar una cantidad de hasta 50 objetos y de la comparación y el orden entre dos cantidades.

-
- Estrategias heurísticas.
 - Estrategias de cálculo mental, como la suma de cifras iguales, el conteo y las descomposiciones del 10.
 - Emplea las siguientes estrategias y procedimientos:
 - Procedimientos de cálculo, como las sumas y restas sin canjes.
 - Estrategias de comparación, como la correspondencia uno a uno.
 - Compara en forma vivencial y concreta la masa de los objetos usando otros objetos como referentes, y estima el tiempo usando unidades convencionales y referentes de actividades cotidianas (días de la semana, meses del año).
 - Realiza afirmaciones sobre las diferentes formas de representar el número y las explica con ejemplos concretos.
 - Realiza afirmaciones sobre los resultados que podría obtener al sumar o restar y las explica con apoyo de material concreto. Asimismo, explica los pasos que siguió en la resolución de un problema.

Fuente: (Minedu, 2016)

Con el propósito de realizar el videojuego con una mecánica fácil de entender para el jugador no se realizaran todos los desempeños de la tabla anterior, ya que el uso de todos estos desempeños exige la creación de una mecánica compleja y extensa para solo un videojuego y muy difícil y aburrido de entender para estudiantes de 6 a 7 años. Comprometiendo el valor de entretenimiento que este tipo de software debe ofrecer a los jugadores. Por esta razón solo se tomaran los siguientes desempeños.

Tabla 8
Competencias del Programa Curricular

Competencia	Descripción
Cardinal hasta 20	Expresa el número como cardinal al determinar una cantidad de hasta 50 objetos. Ejem (1,2,3,4,...,20)
Suma y resta hasta 20 Ejercicios Combinados (Suma y Resta)	Establece relaciones entre datos y acciones de agregar, quitar y juntar cantidades, y las transforma en expresiones numéricas (modelo) de adición o sustracción con números naturales hasta 20
Estrategias Heurísticas	El estudiante puede plantear soluciones mentales por medio de su creatividad en base a los conocimientos adquiridos.
Multiplicación Básica	El estudiante puede realizar multiplicaciones del dos de distintos números hasta 10

Fuente: Elaboración Propia

Estas son las tres etapas de la metodología de Pérez Marques que utilizaremos para el desarrollo del software. De acuerdo a estas tres etapas empezaremos a trabajar el “Concepto” del Videojuego Educativo basándonos en la metodología SUM.

El siguiente paso en el proceso es pensar para cada una de estas competencias, posibles escenarios que tendrá el juego donde desarrollara cada una de estos temas. Desde este punto el desarrollo del concepto se hará usando la metodología SUM y en base a las competencias identificadas en las etapas de la metodología de Pérez Marques. La adaptación de estas dos metodologías tendría el aspecto presentado en la siguiente ilustración.

En este fase los roles: equipo, cliente y verificador serán realizadas por la misma persona, es decir quien ejecuto el proyecto de tesis. El flujo de las tareas realizadas se muestra a continuación.

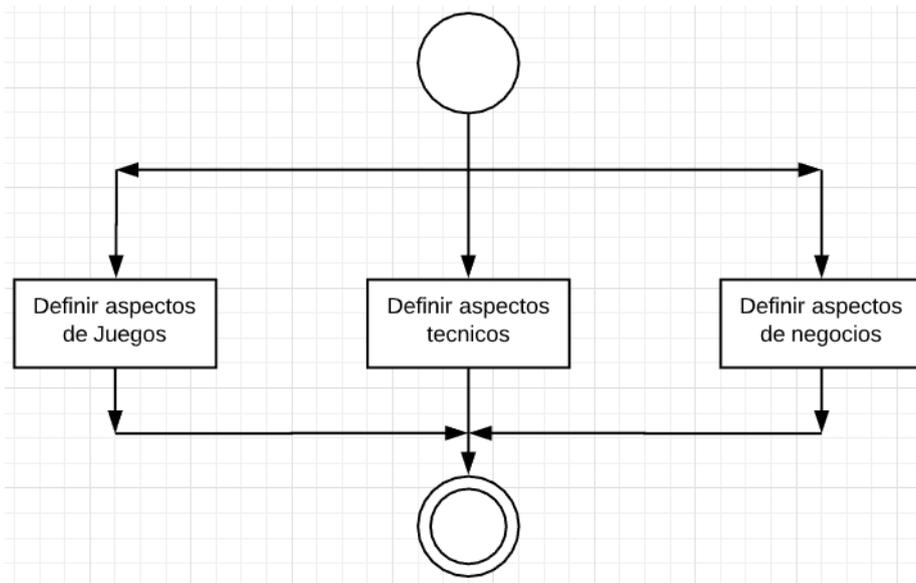


Figura 7 Flujo del desarrollo del concepto

Fuente: <http://www.gemserk.com/sum/>

Para el desarrollo ordenado del “concepto” del videojuego haremos uso del documento de diseño GGD (este documento se encuentra en el anexo B) que servirá como esqueleto de este. Todo el desarrollo del concepto estará definido a detalle en este documento que se irá actualizando en todo el desarrollo del software y servirá como guía para todo el equipo de trabajo, teniendo en cuenta esto empezaremos trabajando con los tres elementos en los que se divide esta primera fase:

Definición de aspectos del juego

En Anexo B contiene el documento de diseño GGD donde se encuentra esta información a más detalle, con el fin de brindar una idea más específica del diseño del concepto del juego en este documento solo se presentara las ideas más estructuradas.

De acuerdo a las capacidades del programa curricular que estamos considerando como objetivos, nos vemos en la necesidad de que el género del videojuego sea de lógica (puzzle), también el jugador deberá poder interactuar con el escenario permitiendo una acción y reacción con los elementos del escenario para poder solucionar puzzles presentados por nivel.

- Genero del Juego: Educativo
- Subcategoría del Juego: RPG, puzzle
- Lógica del Juego: “Duke” es un juego educativo desarrollado para niños, el objetivo del jugador encontrar la casa de un perro perdido; con los consejos y ayuda de otros personajes tendrá que completar distintos puzzles lógico matemáticos que se encontrara

en el camino, como también se enfrentara a distintos enemigos del bosque hasta lograr llegar a su hogar. Todos los elementos propios de los niveles incitan al jugador a desarrollar su lógica y resolver problemas matemáticos.

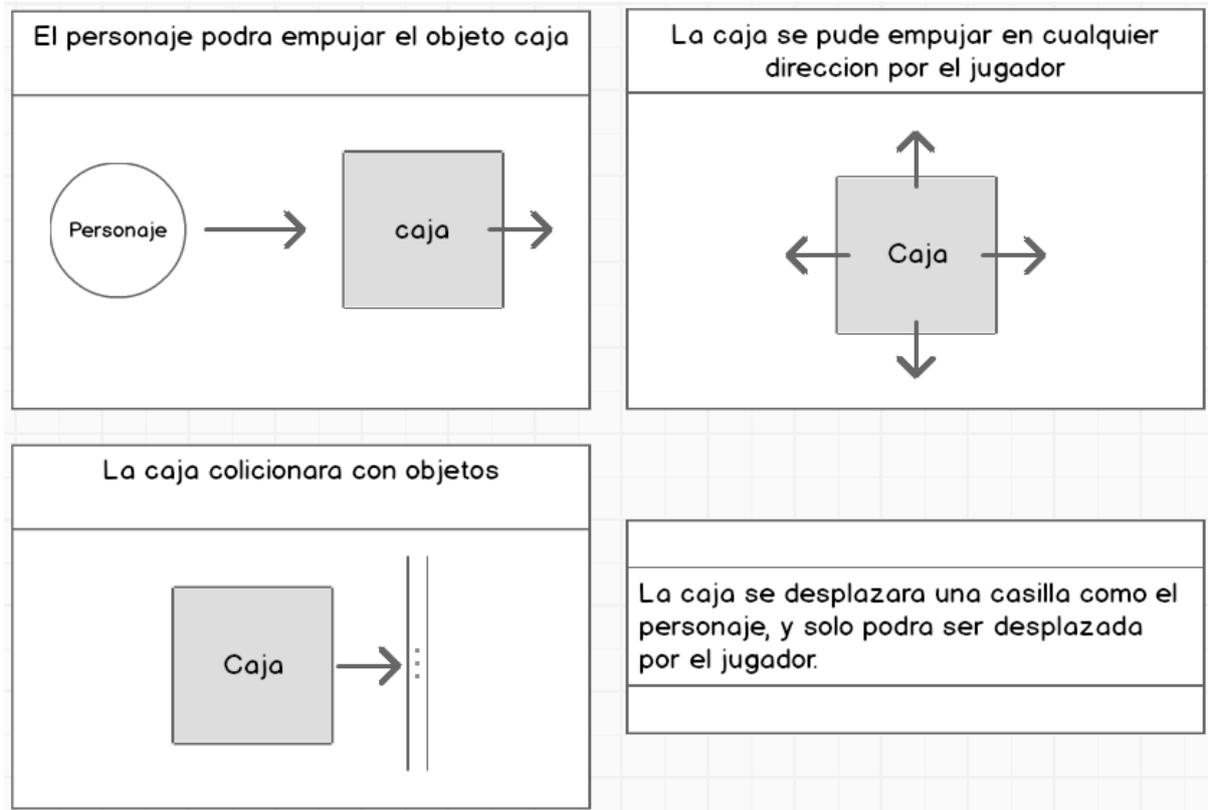


Figura 8 Boceto movimiento de cajas en el escenario

Fuente: Elaboración Propia

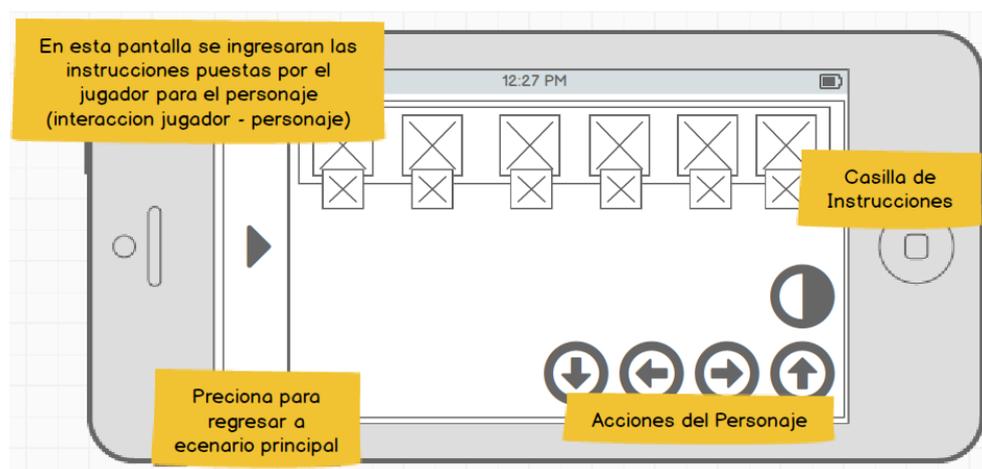


Figura 9 Boceto movimiento por instrucciones del jugador

Fuente: Elaboración Propia

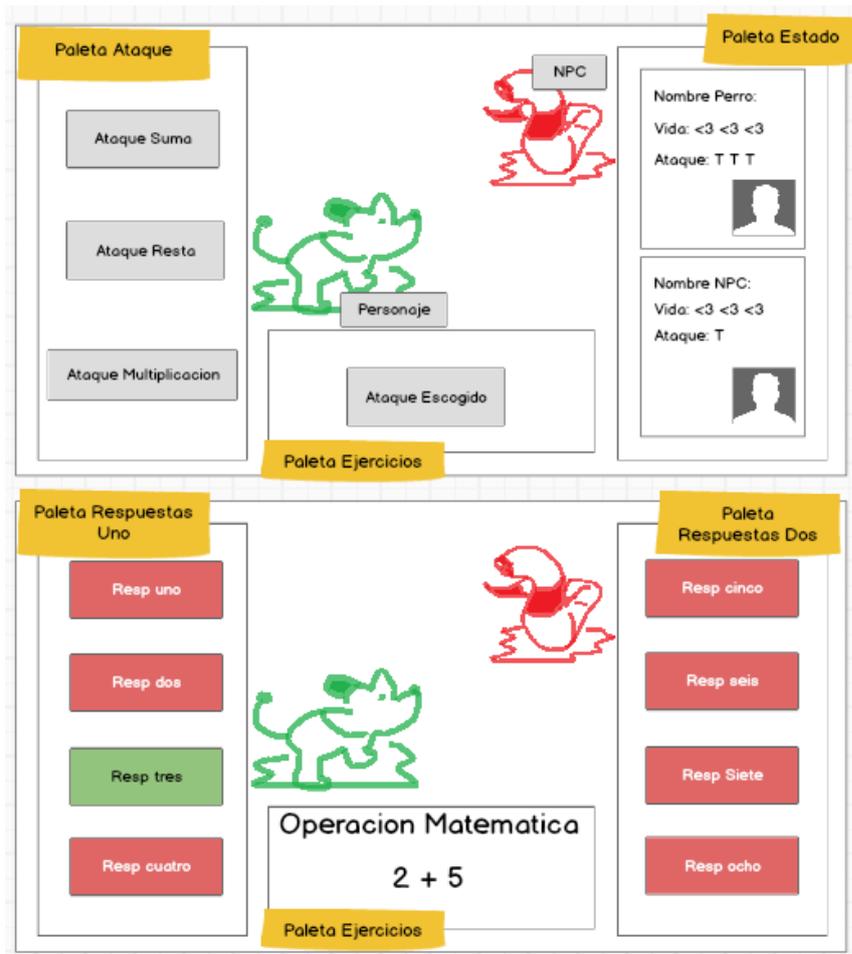


Figura 10 Boceto Modo Pelea contra npc

Fuente: Elaboración Propia

- Interfaces del videojuego, las interfaces que tendrá el videojuego son los siguientes: Menú Principal, Registro del Jugador, Menú Niveles, Menú Pausa, Escena Nivel Terminado, Fin del Juego.

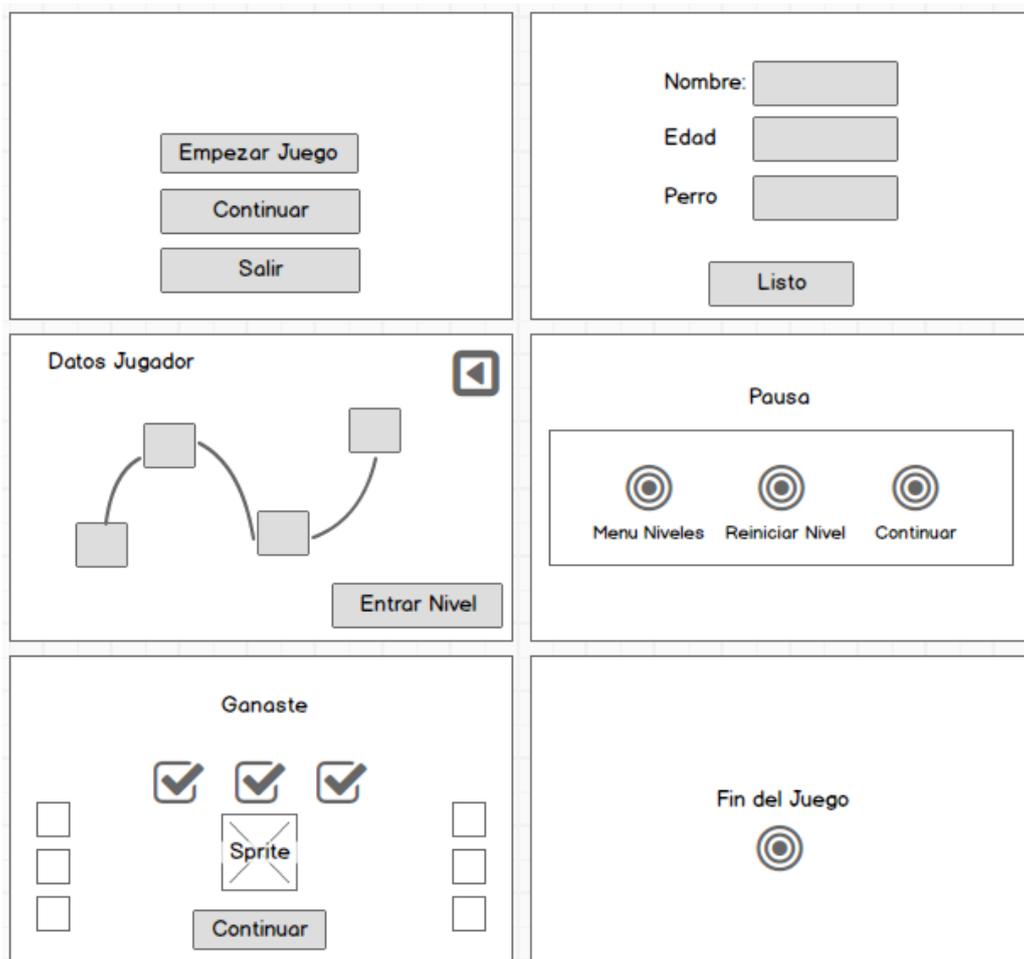


Figura 11 Boceto Interfaces del Videojuego

Fuente: Elaboración Propia

- Niveles del videojuego: El diseño de los niveles se realiza con la finalidad de que cada nivel tenga un propósito específico, por ejemplo nivel tutorial. El objetivo de los primeros niveles es explicar las mecánicas de juego con la interacción del jugador dentro del nivel, posteriormente los siguientes niveles se encargan de utilizar el aprendizaje adquirido en el anterior nivel y aplicarlo en otra situación, esto pasa de nivel a nivel procurando crear una escala de aprendizaje para el jugador hasta adaptar una nueva mecánica de juego a la mecánica ya aprendida en los anteriores niveles. Un ejemplo es el diseño de “niveles tutorial” presentado en la siguiente imagen. El propósito de estos niveles es explicar la mecánica de movimiento de cajas al jugador.

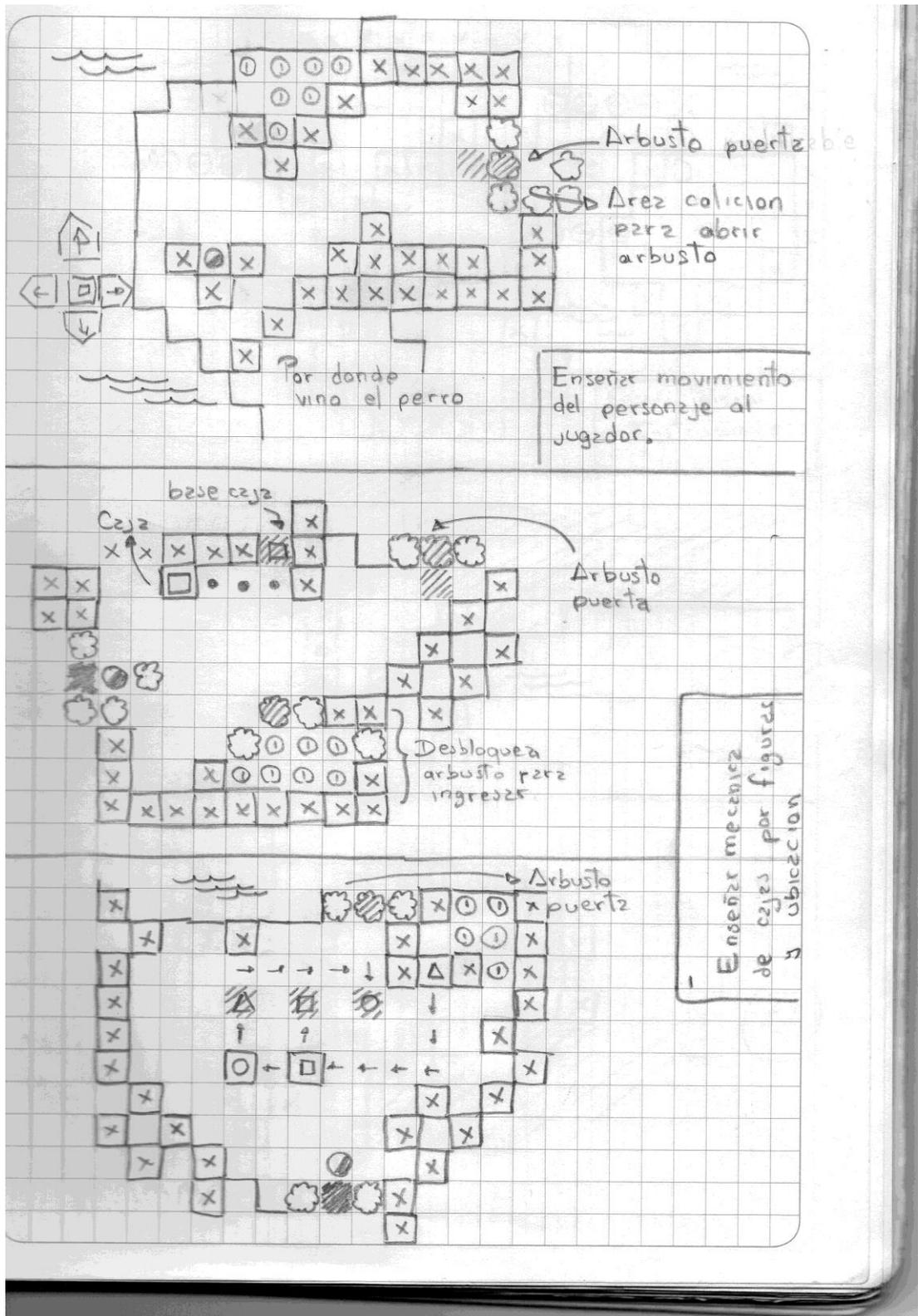


Figura 12 Boceto diseño de nivel tutorial

Fuente: Elaboración Propia

- **Objetivo del Videojuego:** el objetivo de este Serious Gaming es desarrollar las capacidades matemáticas de los estudiantes de primero de primaria mediante la práctica de estas mismas a través del videojuego.

Tabla 9
Objetivos del videojuego de acuerdo a Competencias

Competencias Matemáticas	Mecánica del Juego	Descripción
El juego debe permitir al jugador practicar la competencia "Cardinal hasta 20"	Movimiento por Instrucciones	- El personaje podrá cruzar los estanques de agua nadando pero por medio de instrucciones definidas por el jugador. Para lograra esto el jugador tendrá que contar la cantidad de cubos de agua que tiene que nadar, como también la cantidad de veces que el jugador tiene que girar hasta llegar a la orilla del otro lado.
El juego debe permitir resolver operaciones de suma y resta al jugador de acuerdo a la competencia		- Cuando el jugador colisione con un NPC. El jugador tendrá que enfrentarse al NPC enemigo.
El juego debe permitir resolver operación de multiplicación al jugador de acuerdo a la competencia	Modo pelea con NPC	- El personaje cuenta con tres tipos de ataque, para que el ataque de en el blanco deberá realizar un tipo de operación matemática: Primer ataque Suma, Segundo ataque Resta y Tercer ataque Multiplicación. Si el jugador falla en su respuesta el ataque no tendrá efecto y tendrá que esperar su próximo turno.
El juego debe permitir al jugador desarrollar "Estrategias Heurísticas" de acuerdo a su competencia	Resolución de Puzles	- Cada nivel contara con elementos con los cuales el jugador podrá interactuar para poder superarlos. Para lograr esto el jugador tendrá que hacer uso de su imaginación tomando como limites las reglas de juego y encontrar la solución más practica al problema presentado.

Fuente: Elaboración Propia

- **Definición de características:**
El perro se encontrara como escenario un bosque, con énfasis en diseños coloridos, simples y divertidos, tanto para los elementos del escenario como para los objetos, enemigos, etc. El escenario y los objetos contarán con un diseño 2D isométrico que

contaran con un efecto de luz que dará una apariencia 3D y al mismo tiempo rete al jugador a solucionar cada puzle desde esa perspectiva.

El juego contara con pequeños diálogos entre personajes que serán parte de la historia y comentarios de algunos NPCs en los niveles, esto para profundizar el mundo del juego y contar la historia. Los diálogos pertenecientes a historia mostraran el sprite del personaje y su nombre en un entorno estático y en ellos podemos ver al protagonista y el resto de personajes con más grado de detalle.

En cuanto al apartado musical, se optó por el uso de efectos de sonido tranquilo y pegadizo para los niveles, efectos satisfactorios al completar el nivel y fondo musical inquieto y divertido en el enfrentamiento con personajes no juzgables (NPC), También considerando el cambio de música ambiental dependiendo de nivel ingresado. Todos estos elementos se irán agregando al juego en cuanto se termine su desarrollo.



Figura 13 Escenario Bosque Menú Niveles Boceto

Fuente: Elaboración Propia

Definición de aspectos técnicos

Este software fue desarrollado para la plataforma Android y para su desarrollo se requirió el uso de herramientas como: el motor de desarrollo Godot Engine, programa de edición de imágenes GIMP, editor de arte MagicaVoxel y el resto de herramientas ya mencionadas. Estas herramientas serán utilizadas para el diseño de los elementos y el desarrollo del software.

Plataforma del videojuego

El videojuego está desarrollado para dispositivos Android desde la versión 5.0 en adelante.

Tecnologías y Herramientas

Los materiales y herramientas ya han sido definidos en la tabla 5 con el nombre de “Materiales e Insumos para el desarrollo del software”.

Definición de aspectos del negocio

El Serious Gaming está orientado a estudiantes que cursen el primer grado de primaria cuyo plan educativo este basado al Plan Curricular de educación primaria del ministerio de educación Peruano. El juego será subido a la plataforma de distribución digital “Google Play”.

- Público Objetivo: niños de 5 a 7 años de edad.
- Alcance del Videojuego: El estará disponible para descargarlo en la tienda virtual de Android (Play Store) con el fin de brindar facilidad para su descarga en dispositivos móviles.

3.4.2. PLANIFICACIÓN

En Base al Concepto definido en la anterior fase, desarrollaremos la planificación administrativa y la especificación del videojuego.

Planificación Administrativa:

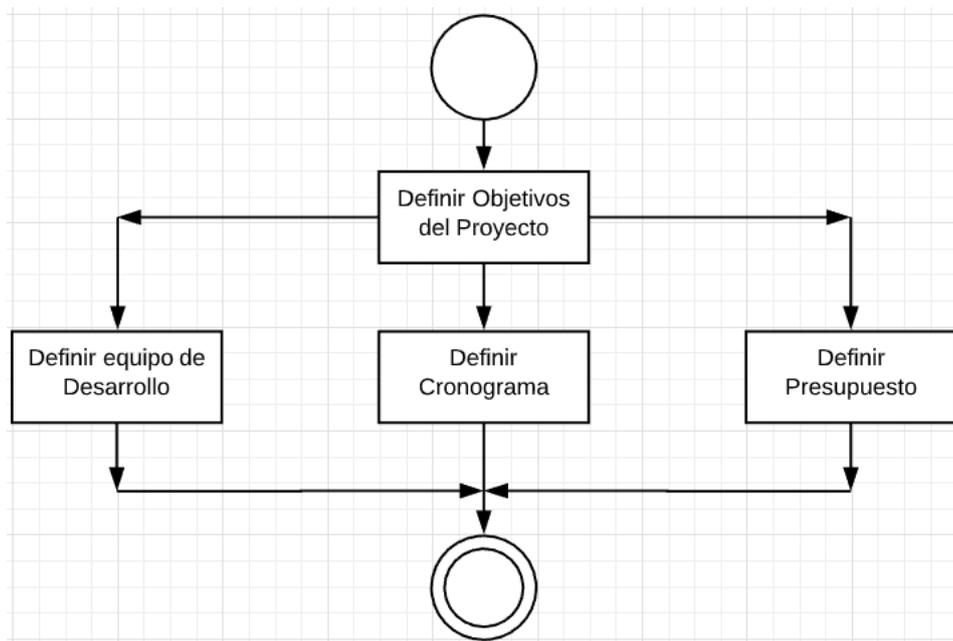


Figura 14 Flujo de tareas de planificación y administración

Fuente: <http://www.gemserk.com/sum/>

El equipo de trabajo consta de un director de proyecto, un diseñador y dos desarrolladores, de acuerdo al concepto del videojuego se realizaron toda la planificación administrativa, cronograma, presupuesto y objetivos del juego.

Definición del Equipo de Desarrollo

Para el desarrollo del videojuego Serios se armó el siguiente equipo de trabajo de:

- Grupo de Administración
 - Productor Interno: el encargado del desarrollo del proceso de desarrollo para el software
- Grupo de desarrollo
 - Diseñador
 - Desarrollador

Definición del Presupuesto

Detallamos el presupuesto necesario para la ejecución del proyecto y fuente financiada para ello:

Tabla 10
Presupuesto para Recursos Humanos Mensual

Recursos Humanos				
Ítem	Descripción	Cantidad	Precio S/	Parcial S/
1.1	Director de Proyecto	1	1200.00	4800.00
1.2	Diseñador	1	1000.00	1000.00
1.3	Desarrollador	2	400.00	1600.00
1.5	Total			7400.00

Fuente: Elaboración Propia

Tabla 11
Presupuesto para Servicios Mensual

Servicios (Gastos Adicionales)				
Ítem	Descripción	Cantidad	Precio S/	Parcial S/
1.1	Servicio de Agua y Luz	-	120.00	480.00
1.2	Servicio de Internet	-	80.00	320.00
1.3	Gastos Extras	-	140.00	560.00
1.4	Total			1360.00

Fuente: Elaboración Propia

Tabla 12
Presupuesto para Materiales un solo pago

Materiales				
Ítem	Descripción	Cantidad	Precio S/	Parcial S/
1.1	Pizarra acrílica movible	-	240.00	240.00
1.2	Herramientas de Oficina	-	80.00	80.00
1.3	Otros Gastos	-	200.00	200.00
1.4	Total			520.00

Fuente: Elaboración Propia

Total				9280.00
-------	--	--	--	---------

El proyecto presentado será financiado con recursos propios.

Tabla 13
Autofinanciamiento

Autofinanciamiento	S/. 9280.00
Total	S/. 9280.00

Fuente: Elaboración Propia

Especificación del Videjuego:

Basándonos en nuestro GGD identificamos los siguientes elementos para desarrollar nuestras características del videojuego, es decir los requisitos funcionales y no funcionales, solo tomaremos las características seleccionadas y su orden de prioridad. El documento completo con el refinamiento de las características se encuentra en el Anexo C Etapas de la Arquitectura del Software.

Tabla 14
Características Funcionales y nivel de prioridad del Juego Serio

Características Funcionales	Nivel de Prioridad
Menú	Alta
Menú Niveles	Alta
Registro Jugador	Alta
Paleta Nivel	Muy Alta
Tablero de Instrucciones	Muy Alta
Personaje	Muy Alta
Tile Set	Muy Alta
Objetos de Interacción	Alta
Ítems	Alta

NPC	Alta
-----	------

Fuente: Elaboración Propia

Tabla 15

Características no Funcionales y nivel de prioridad del Juego Serio

Características no Funcionales	Nivel de Prioridad
Estética del Juego	Muy Alta
Dispositivos disponibles	Muy Alta
Registro	Alta
Paleta de colores Interfaz	Alta
Paleta de colores Niveles	Alta

Fuente: Elaboración Propia

3.4.3. ELABORACIÓN

Planificación de la Interacción

Se segmentó y refinó las distintas Características de tal forma que puedan ser desarrolladas por el equipo de trabajo. El documento segmentado se encuentra en el Anexo C Etapas de Arquitectura del Software.

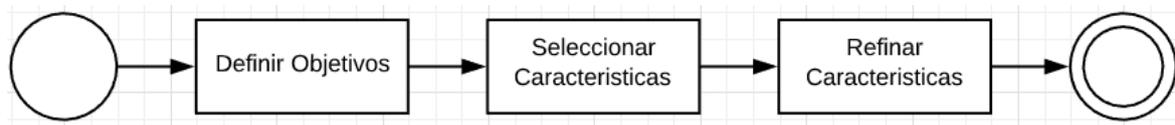


Figura 15 Flujo de la tarea desarrollo de características

Fuente: <http://www.gemserk.com/sum/>

Se realizaron quince interacciones, cada interacción contempla los requisitos funcionales a ser desarrollados.

Tabla 16

Relación de interacciones, requisitos e interfaces

Interacción	Elementos a Desarrollar (RF)	Refinamiento de Elementos	de Interfaces
Interacción 1	1, 2, 3, 4, 5, 6, 7, 8, 9, 11	Descomponer cada característica en tareas	Menú Principal, Menú Niveles
Interacción 2	13, 14		Registro del Jugador

Interacción 3	15, 16, 17, 19, 21, 22, 23, 24, 25	y desarrollarlo de manera secuencial.	Paleta Nivel
Interacción 4	18		Paleta Nivel
Interacción 5	20, 26, 27		Paleta Nivel
Interacción 6	28, 29, 30, 31, 32, 33		Tablero de Instrucciones
Interacción 7	34, 36, 37, 38, 39, 45, 46, 47		Personaje
Interacción 8	35		Personaje
Interacción 9	40, 65		Personaje, NPC
Interacción 10	41, 42		Personaje
Interacción 11	43, 44		Personaje
Interacción 12	49, 50, 51, 52, 53		Tile Set
Interacción 13	54, 55, 56, 57, 58		Objetos de Interacción
Interacción 14	59, 60		Ítems
Interacción 15	61, 62, 63, 64, 66, 67		NPC

Fuente: Elaboración Propia

Desarrollo de Características

En esta etapa se desarrolló cada característica del software de acuerdo a las interacciones planificadas. Para ver más a detalle el desarrollo con respecto a la estructura de nodos del software se puede consultar el anexo D Estructura de nodos Duke App.

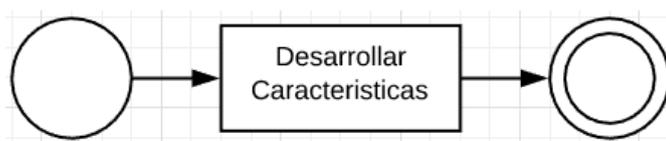


Figura 16 Flujo de la tarea desarrollo de características

Fuente: <http://www.gemserk.com/sum/>

Personaje

Para el desarrollo del personaje se realizaron las siguientes tareas utilizando el software de administración de proyectos Trello, como se muestra en la siguiente imagen.

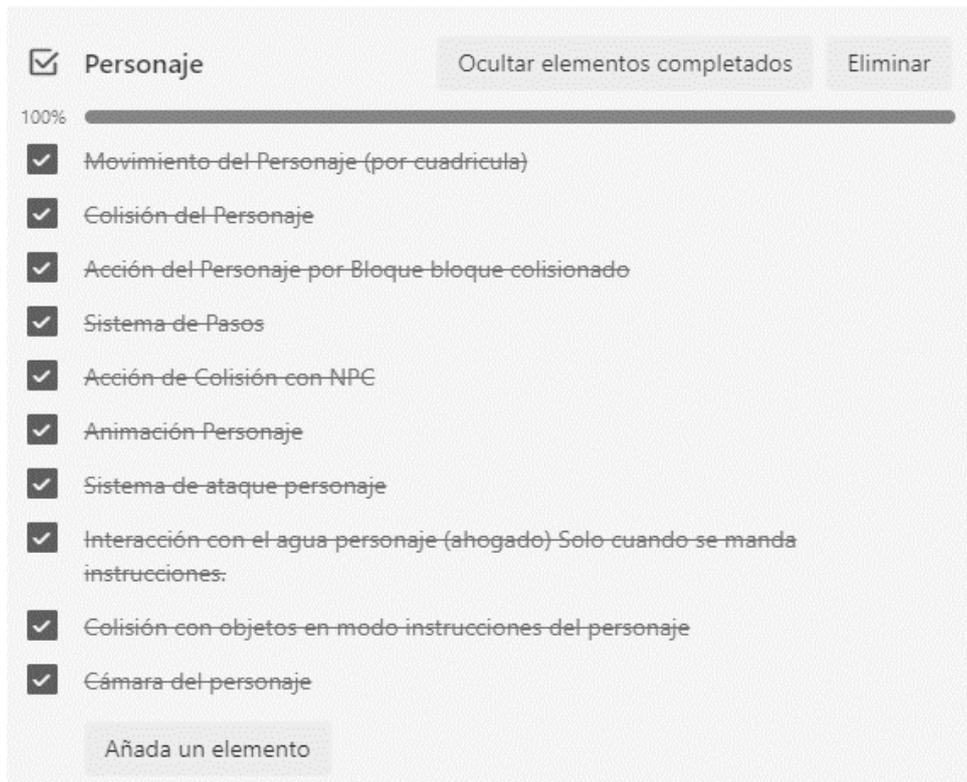


Figura 17 Tareas para el desarrollo del personaje

Fuente: Elaboración Propia

Los tipos de nodos utilizados para el desarrollo del personaje fueron los siguientes: Sprite, RayCast2D, Tween, Camara2D, Position2D, Animation Player, Area2D, CollisionShape2D, StaticBody2D y Label.

El diseño del personaje tiene que ser de tipo isométrico ya que el juego cuenta con esa característica, el sprite utilizado para el personaje fue diseñado usando “MagicaVoxel”.

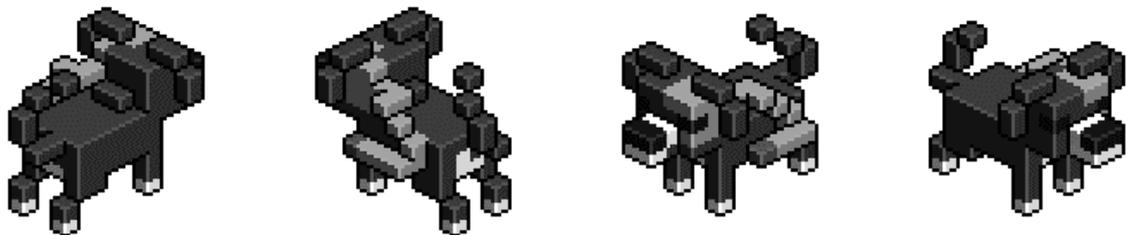


Figura 18 Sprite personaje principal

Fuente: Elaboración Propia

Como referencia se mostrara solo partes del código de programación del personaje:

```

7 var dir = {
8   "derecha": Vector2(40,20),
9   "izquierda": Vector2(-40,-20),
10  "arriba": Vector2(40,-20),
11  "abajo": Vector2(-40,20),
12 }

199 #FUNCIONAMIENTO CON TECLAS
200 if Input.is_action_pressed("up"):
201   frame = 0
202   bloqueo = false
203   if (!$arriba.is_colliding()):
204     mueve_a(dir.arriba)
205     bloqueo = true
206 if Input.is_action_pressed("down"):
207   frame = 2
208   bloqueo = false
209   if (!$abajo.is_colliding()):
210     mueve_a(dir.abajo)
211     bloqueo = true
212 if Input.is_action_pressed("left"):
213   frame = 1
214   bloqueo = false
215   if (!$izquierda.is_colliding()):
216     mueve_a(dir.izquierda)
217     bloqueo = true
218 if Input.is_action_pressed("right"):
219   frame = 3
220   bloqueo = false
221   if (!$derecha.is_colliding()):
222     mueve_a(dir.derecha)
223     bloqueo = true
224

323 func mueve_a(dir): # Movimiento por casilla del Personaje
324 if (modoDeJuego == modoJuego.instrucciones):
325   $anim_jugador.play("suspender")
326   #! get_tree().get_nodes_in_group("sfx")[0].get_node("1").pla
327 elif (modoDeJuego == modoJuego.plataforma):
328   $anim_jugador.play("salto")
329   #!
330   puede_mover = false
331   $Tween.interpolate_property(self,"global_position",
332     global_position, global_position + dir, 0.5,
333     Tween.TRANS_LINEAR, Tween.EASE_IN_OUT)
334   $Tween.start()
335   yield($Tween,"tween_completed")
336   puede_mover = true

```

Figura 19 Movimiento del Personaje por touch

Fuente: Elaboración Propia

```

420 func colicion_npc_jugador(valor):
421   puede_mover = false # Bloqueamos el movimiento del jugador
422   get_tree().get_nodes_in_group("estado_npc")[1].recetear_heart() #Receteamos los corazones antes de empezar
423   # Mandamos los valores de vida del jugador y tambien nombre
424   get_tree().get_nodes_in_group("estado_npc")[1].cont_heart(vidaJug, nombre_perro, gameHandler.fuerzaAtak)
425   #Deshabilitamos el movimiento del jugador y tambien grabamos la pocicion actual de la camara
426   posCamara = $Camera2D.global_position # Guardamos la posicion de la camara normal
427   #var paletaEstadoJugador = get_tree().get_nodes_in_group("estado_atk")[0] # Variable para nodo de paleta
428   match valor: # hacemos un zoom y acomodamos la camara para el modo pelea
429     1:
430       $Tween.interpolate_property($Camera2D, "global_position",|
431         $Camera2D.global_position, $abajo.global_position + Vector2(0,60),
432         1, Tween.TRANS_LINEAR, Tween.EASE_IN_OUT)
433       frame = 2 # orientamos al jugador mirando al npc
434     2:
435       $Tween.interpolate_property($Camera2D, "global_position",
436         $Camera2D.global_position, $derecha.global_position + Vector2(0,60),
437         1, Tween.TRANS_LINEAR, Tween.EASE_IN_OUT)
438       frame = 3
439     3:
440       $Tween.interpolate_property($Camera2D, "global_position",
441         $Camera2D.global_position, $arriba.global_position + Vector2(-10,60),#-10,60
442         1, Tween.TRANS_LINEAR, Tween.EASE_IN_OUT)
443       frame = 0
444     0:
445       $Tween.interpolate_property($Camera2D, "global_position",
446         $Camera2D.global_position, $izquierda.global_position + Vector2(15,10),#(60,0)
447         1, Tween.TRANS_LINEAR, Tween.EASE_IN_OUT)
448       frame = 1
449   $Tween.start()
450   yield($Tween, "tween_completed")
451   $Camera2D.zoom = Vector2(0.6,0.6) #Vector2(0.6,0.6)
452
453 # Habilitamos nodo paletaPelea de Main pelea
454 get_tree().get_nodes_in_group("mainPelea")[0].habilitarPaletaPelea()

```

Figura 20 Colisión del Personaje

Fuente: Elaboración Propia

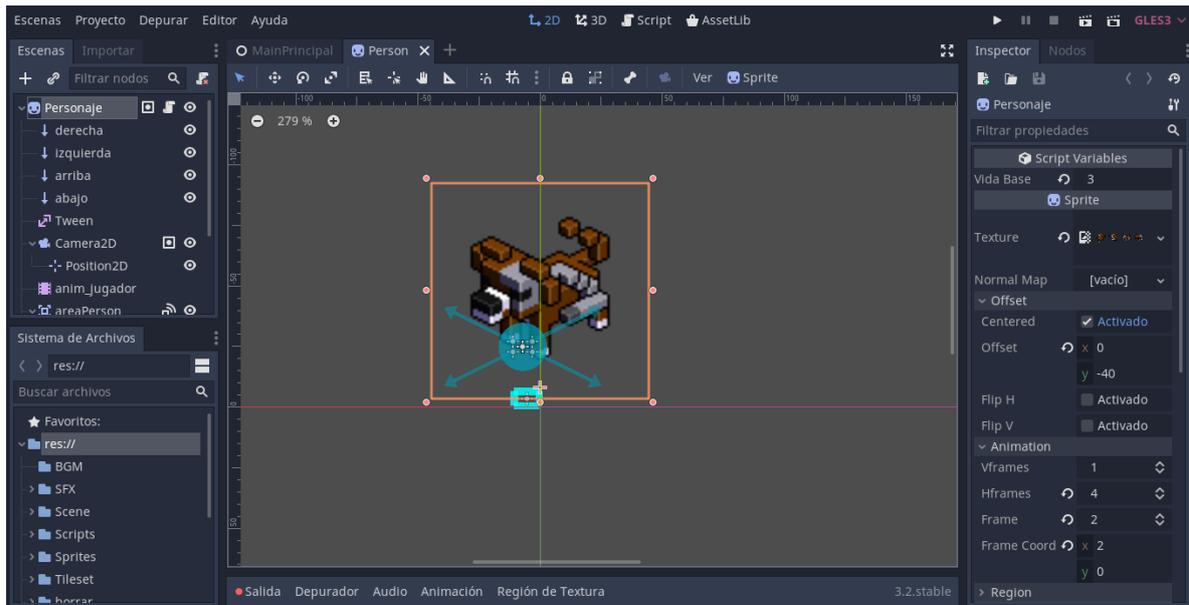


Figura 21 Escena Personaje en Godot Engine

Fuente: Elaboración Propia

Tablero Instrucciones

Para el desarrollo del tablero de instrucciones se realizaron las siguientes tareas

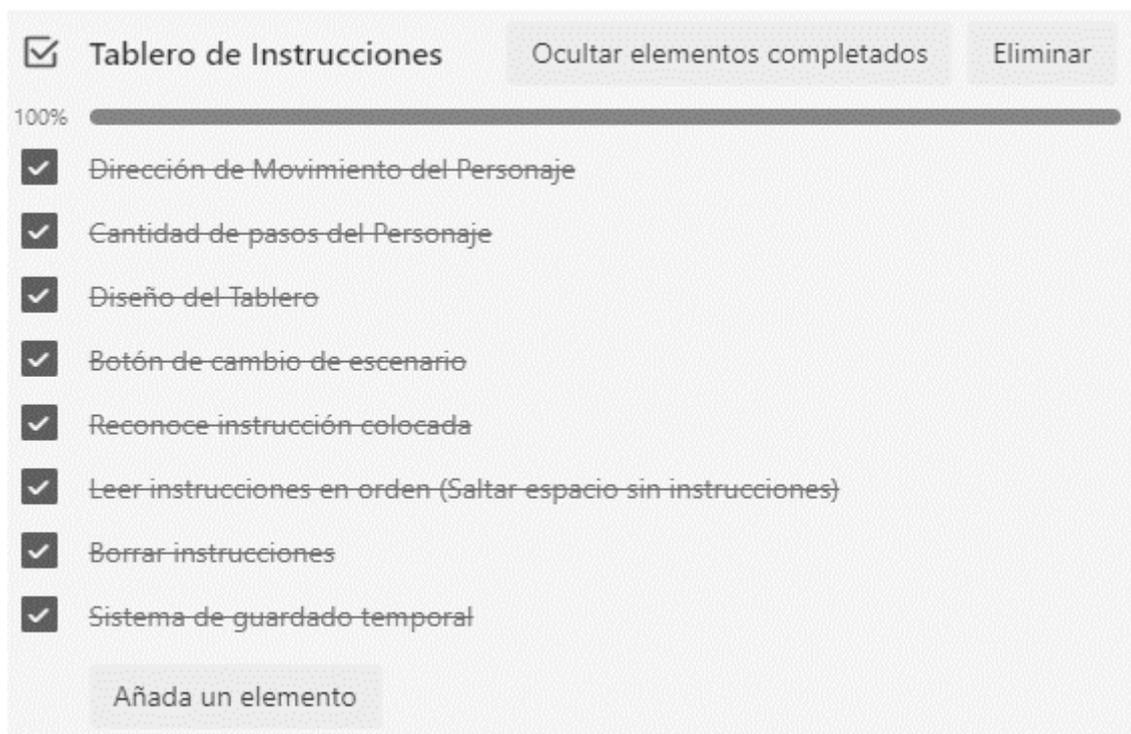


Figura 22 Tareas para el desarrollo del tablero de instrucciones

Fuente: Elaboración Propia

Los tipos de nodos utilizados para el desarrollo del tablero de instrucciones son los siguientes: Control, NinePatchRect, ColorRect, HBoxContainer, TextureButton, Label, TextureRect y AnimationPlayer.

Para el diseño de los sprites del tablero de instrucciones se usó la herramienta Microsoft Paint, ya que estos elementos en su mayoría eran botones.

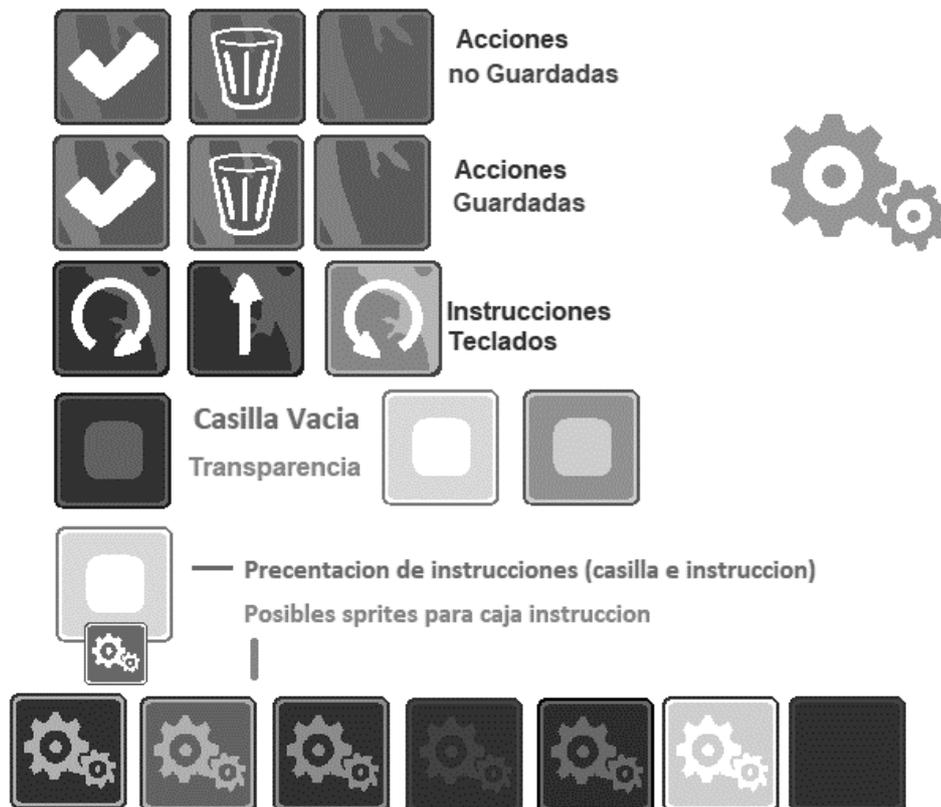


Figura 23 Sprites de la escena Tablero de Instrucciones

Fuente: Elaboración Propia

Como referencia se mostrara solo partes del código de programación del Tablero de Instrucciones.

```

44 > func _on_cassUno_pressed(): #Casilla Uno
45 > | guardado_instrucc()
46 > | $cajas/anim_cass.stop()
47 > | cass_select = true # Se selecciono una casilla
48 > | num_cass = 1 # Casilla seleccionada

```

Figura 24 Selección de Casilla Escena Tablero de Instrucciones

Fuente: Elaboración Propia

```

82 v func _on_bot_adelante_pressed(): # Boton avanzar
83 v |> if (cass_select): # preguntamos si se selecciono alguna casilla
84 |> |> tareaUno = true
85 |> |> tarea = 1 # Tipo de tarea que realiza el boton
86 v |> |> if (tareaUno and !tareaDos and !tareaTres): # Verificamos si
87 |> |> |> asigna_sprite() # Asignamos sprite a la casilla y agrega
88 v |> |> else:
89 |> |> |> cantidad = 0
90 |> |> |> tareaDos = false
91 |> |> |> tareaTres = false
92 |> |> |> asigna_sprite()
93 |> |> cantidad += 1
94 |> |> printCantidad()
95 v |> else:
96 |> |> print ("no selecciono ni una casilla")

```

Figura 25 Dirección de movimiento del Personaje

Fuente: Elaboración Propia

```

54 v func presBot_accion():
55 |> modoDeJuego = modoJuego.instrucciones
56 |> # Atrapamos en la variable el nodo tablaInstrucciones
57 |> var tablaInstrucciones = get_tree().get_nodes_in_group("tablero")[0]
58 |> #|> invulnerabilidad = true # activamos invulnerabilidad
59
60 v |> for i in range (6): # Recorremos cada array de las instrucciones para ejecutarlas
61 |> |> casilla = tablaInstrucciones.instrucciones[i][0] # Numero de Casilla
62 |> |> accion = tablaInstrucciones.instrucciones[i][1] # Tipo de accion
63 |> |> repeticion = tablaInstrucciones.instrucciones[i][2] # Cantidad de veces que se
64 v |> |> for rep in (repeticion): # Reproduce accion de acuerdo a la cantidad (repeticion)
65 |> |> |> # Mandamos el numero de casilla al nodo degui para graficar animacion de instr
66 |> |> |> get_tree().get_nodes_in_group("gui")[0].reproducir_tablaMini(casilla)
67 |> |> |> movPersonaje()
68 |> |> |> yield(get_tree().create_timer(0.5), "timeout")
69 v |> |> |> if (colision_bloque): # Identificar
70 |> |> |> |> yield()
71 v |> |> if (!piso_jugador): # Si el jugador esta el vacio
72 |> |> |> muerto = true # Activamos la variable muerto
73 |> |> |> $anim_jugador.play("morir") # Reproducimos animacion de morir
74 |> |> |> yield($anim_jugador,"animation_finished") # Hcemos una pausa
75 |> |> |> visible = false # Escondemos el nodo personaje
76 |> |> |> morirJugador()
77
78 |> |> obstaculo = false
79 |> |> modoDeJuego = modoJuego.plataforma
80 |> |> tablaInstrucciones.tabla_reset() # Despues de reproducir la instruccion receteamos

```

Figura 26 Leer las instrucciones del Personaje

Fuente: Elaboración Propia

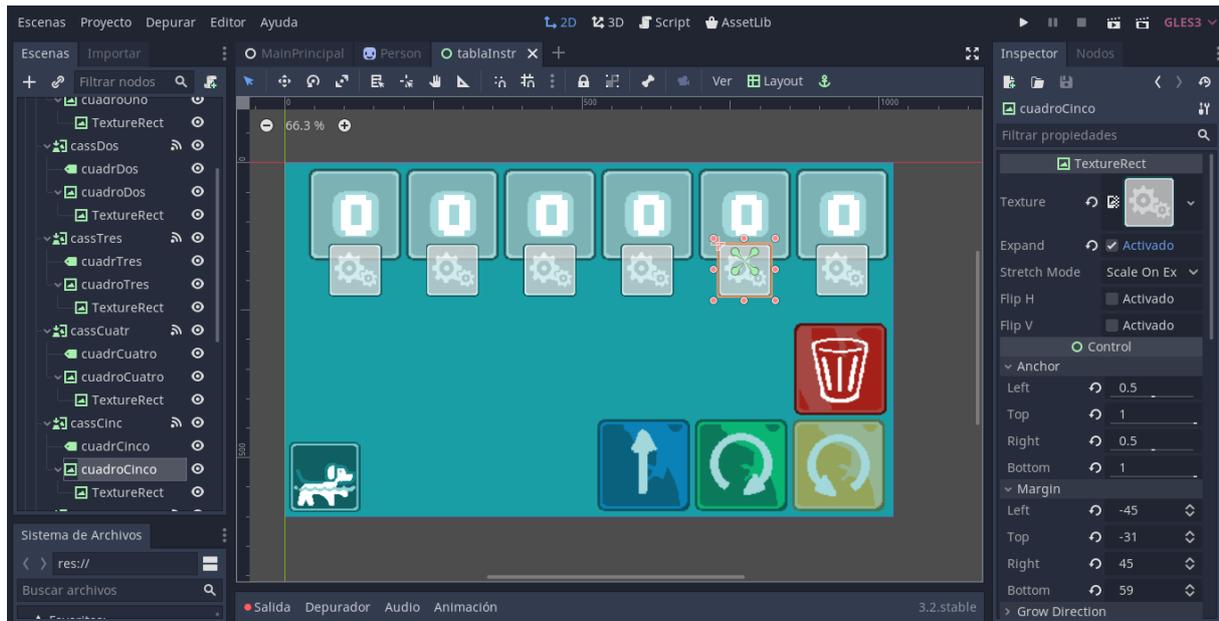


Figura 27 Escena Tabla de Instrucciones en Godot Engine

Fuente: Elaboración Propia

Tile Set

Tile Set se creó para el diseño de los niveles ya que las escenas creadas y agregadas al Tile Set se reusaran para cada nivel. Las tareas para su desarrollo son las siguientes.

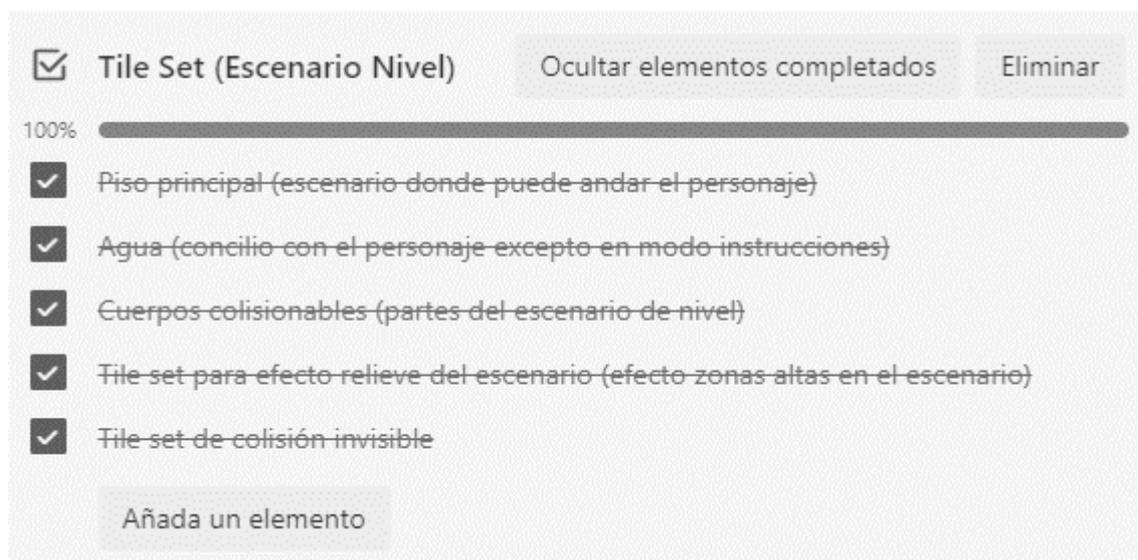


Figura 28 Tareas para el desarrollo de Tile Set

Fuente: Elaboración Propia

Los tipos de nodos utilizados para la escena Tile Set fueron los siguientes: Node, TileMap, Sprite, StaticBody2D, Area2D y Position2D.

Todos los sprites diseñados fueron realizados con la herramienta “MagicaVoxel” como se muestra en la siguiente imagen.



Figura 29 Sprites diseñados para Tile Set (Niveles)

Fuente: Elaboración Propia

Para el desarrollo de la escena TileSet (Nivel) solo se realizó el armado de elementos como sprites dentro del TileMap instanciando nodos ya desarrollados como son:

- Personaje.
- Objetos de interacción.
- Ítems.
- NPC.
- Etc.

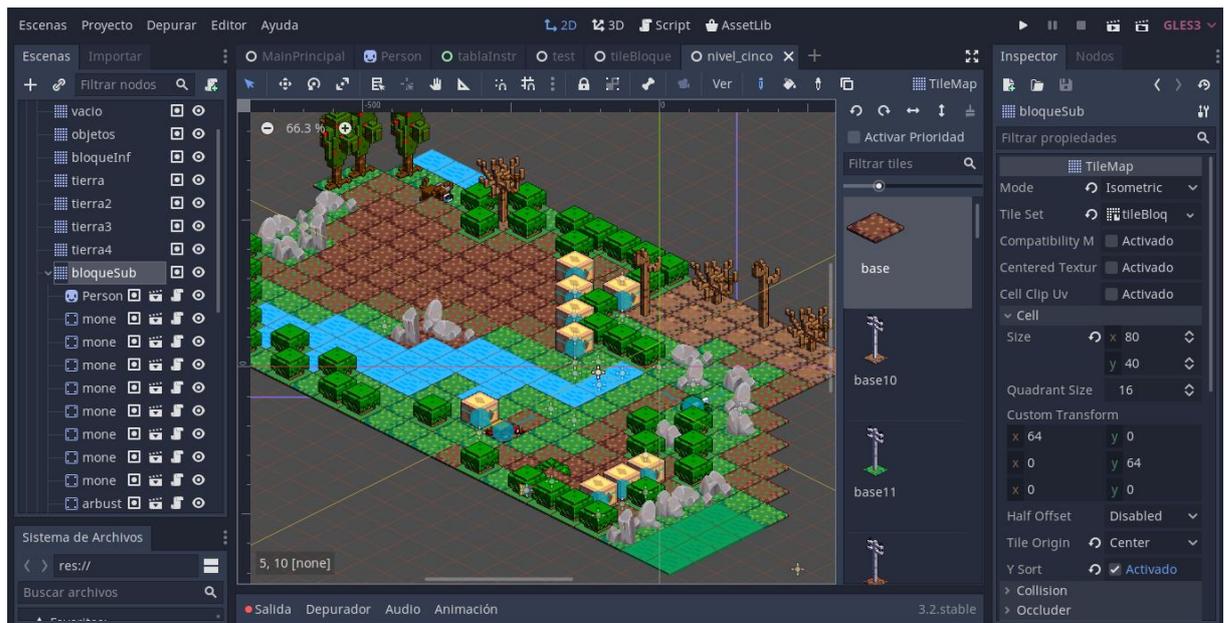


Figura 30 Escena Tile Set (Niveles) en Godot Engine

Fuente: Elaboración Propia

Seguimiento de la Interacción

En esta tarea se verificó que las características desarrolladas cumplan con el marco de objetivos y requisitos planeados y se aseguró que no haya percances en el desarrollo por interacción.

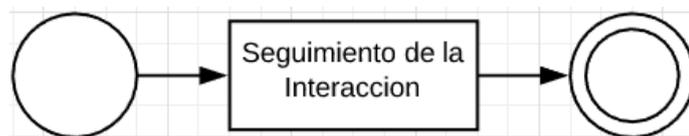


Figura 31 Flujo de monitoreo de la Interacción

Fuente: <http://www.gemserk.com/sum/>

El desarrollo de esta se muestra a continuación:

Tabla 17
Seguimiento de las interacciones

Interacción	Problemas Técnicos durante el desarrollo.	Realizado conforme a los resultados.
Interacción 1	Ninguno	Si
Interacción 2	Ninguno	Si
Interacción 3	Ninguno	Si

Interacción 4	Ninguno	Si
Interacción 5	Ninguno	Si
Interacción 6	Ninguno	Si
Interacción 7	Ninguno	Si
Interacción 8	Ninguno	Si
Interacción 9	Ninguno	Si
Interacción 10	Ninguno	Si
Interacción 11	Ninguno	Si
Interacción 12	Ninguno	Si
Interacción 13	Ninguno	Si
Interacción 14	Ninguno	Si
Interacción 15	Ninguno	Si

Fuente: Elaboración Propia

Cierre de la Interacción

En esta actividad se evaluó el estado del videojuego con esta implementación, también se evaluó el cumplimiento de los objetivos de la interacción y en el caso de existir algún problema se actualizo el plan de proyecto.

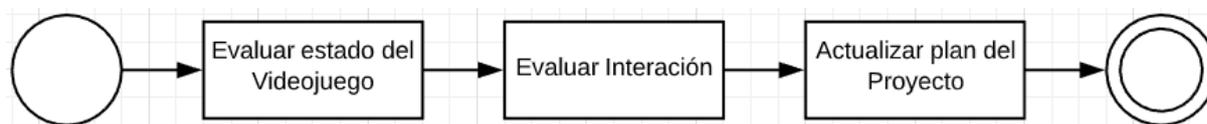


Figura 32 Flujo del Cierre de la Interacción

Fuente: <http://www.gemserk.com/sum/>

El videojuego conto con 15 versiones internas de desarrollo, cada versión creada fue ejecutada sin ningún percance. De versión en versión se ha ido implementando las características y evaluando su interacción unas con otras hasta implementar todas las características en cada interacción, luego de la cual fue actualizado el proyecto. En la tabla presentada a continuación se detalla la información de las versiones:

Tabla 18
Versiones del videojuego

Versión	Características	Porcentaje de avance del software versiones	de Interacciones por Realizadas	Versión Jugable
0.1	Desarrollo del Menú principal, Menú Niveles.	6.6 de 100	1	Si
0.2	Guardado de partida, registro jugador y registro de datos de partida.	13.2 de 100	2	Si
0.3	Panel de Control: fondo base, escenario dinámico, botones touch, punto visual, cambio de escena, reproducción de comandos, nivel.	19.8 de 100	3	Si
0.4	Menú Pausa.	26.4 de 100	4	Si
0.5	Escena nivel terminado, sistema de estrellas, sistema de hitos.	33 de 100	5	Si
0.6	Casillas de instrucciones, botones acción, cambio de escena, botón borrar casilla, fondo base y sistema guardado.	39.6 de 100	6	Si
0.7	Movimiento de botones, cámara personaje, animaciones del personaje, colisión escenario, tipos de ataque, colisiones ítems, colisiones NPC y colisiones temporales.	46.2 de 100	7	Si
0.8	Movimiento por instrucciones personaje.	52.8 de 100	8	Si
0.9	Paleta que muestra estado del personaje y NPC.	59.4 de 100	9	Si

1.0	Paleta de respuestas alternativas, paleta ejercicios.	66 de 100	10	Si
1.1	Sistema de puntos y sistema de corazones.	72.6 de 100	11	Si
1.2	Bloque colisión, agua, piso principal. Bloque relieve y bloque invisible.	79.2 de 100	12	Si
1.3	Caja movable, base cajas, arbusto móvil, bandera y rampas.	85.8 de 100	13	Si
1.4	Corazones, huesos (puntos). Movimiento NPC, colisión con jugador, derrota, posibilidad de	92.4 de 100	14	Si
1.5	ataque, sistema daño y características especiales.	100 de 100	14	Si

Fuente: Elaboración Propia

3.4.4. BETA

En esta fase se desarrolló una planificación de las interacciones dentro del juego. Con la finalidad de distribuir los elementos que se revisaran. Para luego realizar una verificación completa y pasar a la corrección de posibles errores o problemas.

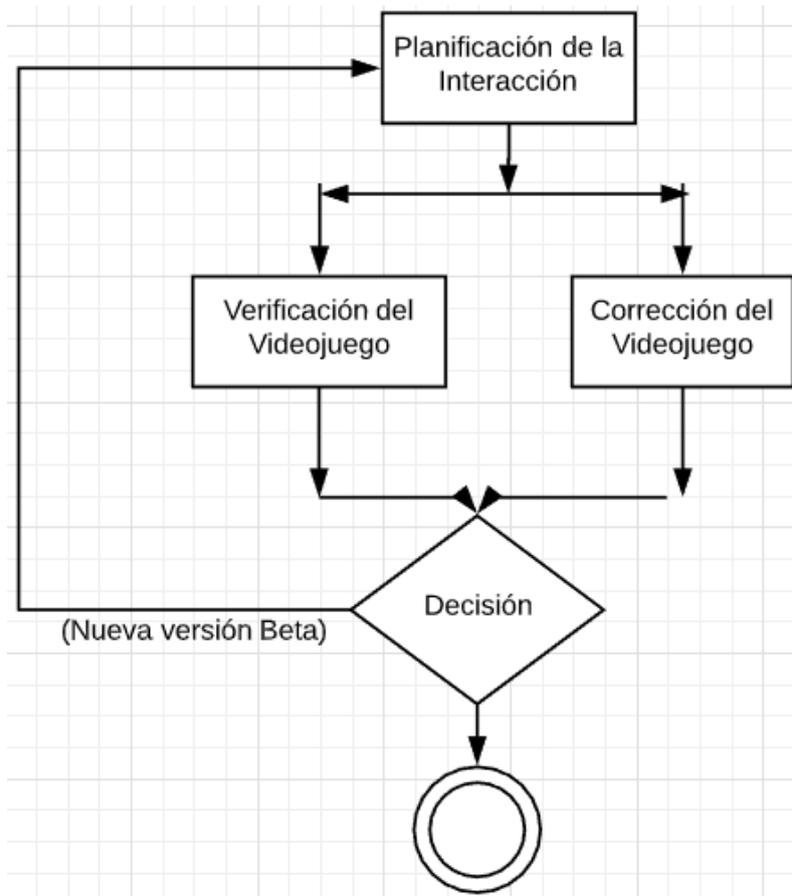


Figura 33 Flujo de Interacción Beta

Fuente: <http://www.gemserk.com/sum/>

Planificación de la Interacción

Se examinó que requisitos funcionales y no funcionales se van a revisar, luego se realizó la tarea de distribuir la versión beta que fuera verificada. El flujo de las tareas se muestra a continuación.

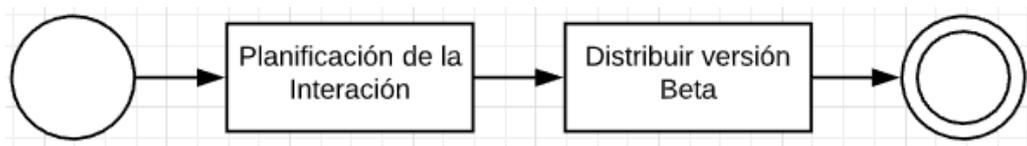


Figura 34 Flujo de tareas de planificación de la interacción

Fuente: <http://www.gemserk.com/sum/>

Verificación del Videojuego

En este punto se evaluó y verifico el videojuego.



Figura 35 Verificación del Videojuego

Fuente: <http://www.gemserk.com/sum/>

Tabla 19

Interfaces verificadas del videojuego

Nro.	Interfaz Probada	Condiciones
01	Menú Principal	<ul style="list-style-type: none"> - Transición de caratula. - Identificación automática de partida anterior en el juego. - Reiniciar partida y guardar datos jugador. - Ingresar a partida anterior con todos los datos guardados y salir de la aplicación. - Mostrar nombre del jugador. - Activación de niveles de acuerdo a niveles superados por el jugador.
02	Menú Niveles	<ul style="list-style-type: none"> - Redirección a nivel seleccionado por medio de un botón. - Redirección al menú principal por medio de un botón. - Animación aleatoria de niveles secretos.
03	Paleta Nivel	<ul style="list-style-type: none"> - Funcionamiento correcto de botones touch y teclado. - Impresión de nivel en fondo base de acuerdo a nivel escogido. - Cabecera con registro de puntos almacenados actualizable. - Cabecera con activación a menú pausa por medio de un botón. - Guardado automático de partida del jugador - Salto de nivel a nivel. - Salto de nivel elección - Mecánica de movimiento por touch o teclado.
04	Personaje	<ul style="list-style-type: none"> - Identificación de tipo de colisión con objetos, npc, ítems. - Cambio de sprite por botón de movimiento presionado - Animación de Movimiento.

		<ul style="list-style-type: none"> - Condiciones de movimiento sobre colisiones. - Identificación de ultima colisión atravesada - Identificación de daño del jugador. - Identificación de cantidad de daño por tipo de npc. - Animación de fin del juego (caer al agua, perder contra npc) - Movimiento de cámara en base al movimiento del personaje. - Modificación de acercamiento de la cámara en modo pelea. - Activación de sistema de estrellas de acuerdo a nivel terminando.
05	Nivel Terminado	<ul style="list-style-type: none"> - Activación de botón pantalla para salto de nivel - Animación de perro como reacción a botón presionado - Animación de texto para dar sensación de cambio de escena.
06	Menú Pausa	<ul style="list-style-type: none"> - Redireccionamiento a menú niveles, reinicio de partida y continuar partida, pausar partida. - Asignación de acción a casilla, cantidad de veces que se repite la instrucción.
07	Tabla de Instrucciones	<ul style="list-style-type: none"> - Proceso de asignación de instrucción, guardado y reseteo. - Reproducción de instrucciones en paleta nivel. - Reseteo de la tabla completa al reproducir instrucción.
08	Paleta estado (personaje y NPC)	<ul style="list-style-type: none"> - Muestra de características actualizadas jugador y npc. - Muestra visual de estado de vida personaje y npc. - Muestra visual de estado de ataque de personaje y npc. - Animación de sprites personaje y npc. - Impresión de Ejercicio de acuerdo al tipo de ataque y dificultad seleccionado
09	Paleta Ejercicios	<ul style="list-style-type: none"> - Resolución interna del ejercicio
10	Paleta respuestas	<ul style="list-style-type: none"> - Selección de casilla repuesta correcta - Impresión de respuestas incorrectas en base a respuesta correcta. - Comparación de respuesta seleccionada con correcta.
11	Generar Ejercicio	<ul style="list-style-type: none"> - Generación de ejercicio de acuerdo de dificultad y tipo de operación.

		- De acuerdo al tipo de operación estructurar posición de números en el ejercicio.
		- Elementos colisionables ajustados a dimensiones de sprite.
12	Tile Set	- Modo de tilemap isométrico.
		- Elementos no colisionables ajustados a tile set.
		- Caja se desplaza solo con la colisión del personaje, identificación de posición de caja, identificación de cuerpos colacionables.
		- Identificación de tipo de caja ingresada a su área.
13	Objetos	- Identificación de cantidad de puzles dentro del nivel.
	Interacción	- Desactivación de arbusto por cajas ajustadas.
		- Detección de cuerpos colisionables.
		- Bandera: Identificación de personaje para terminar nivel.
		- Rampa: Identificación de personaje para activar botón instrucciones.
14	Ítems	- Desactivación por colisión del personaje.
		- Suma de puntos o corazones a jugador.
		- Colisión con jugador activación de modo pelea.
		- Colisión con objetos del Tile set
15	NPC	- Movimiento automático por escenario presentado.
		- Desactivación del npc por pérdida con jugador.
		- Probabilidad de ataque exitoso.
		- Sistema de daño

Fuente: Elaboración Propia

Corrección del Videojuego

En esta punto se realizaron las correcciones a los errores encontrados en el juego de acuerdo a la prioridad de impacto, el flujo de estas tereas se muestra en la siguiente figura.

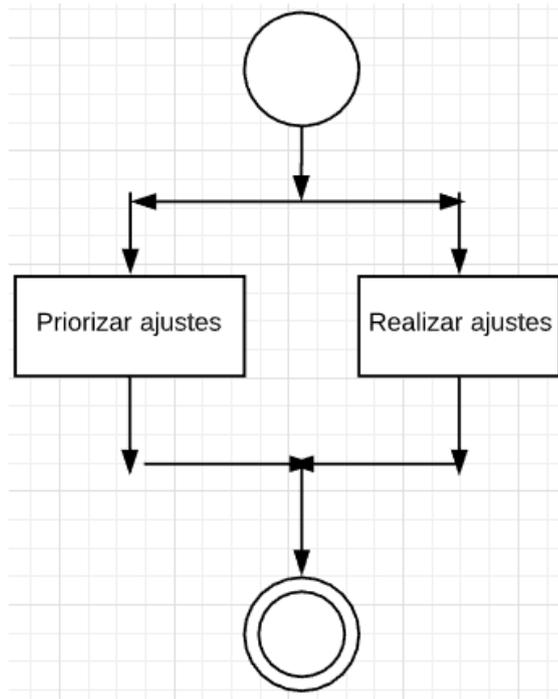


Figura 36 Flujo de Corrección del videojuego

Fuente: <http://www.gemserk.com/sum/>

El desarrollo de las siguientes tareas se muestra a continuación.

Tabla 20
Lista de Errores detectados y Solucionados

Nro.	Error	Prioridad (Alta, media, baja)	Solucionado
1	El personaje puede atravesar muros por medio de instrucciones. (tabla de instrucciones, personaje)	Alta	Si
2	Ajustado de Resolución para tipo de pantalla. (configuración motor)	Alta	Si
3	La interacción del personaje con nodo rampa causa error al reiniciar el nivel.	Alta	Si
4	Existe un bug al completar el puzle de cajas, solo funciona en un solo orden.	Alta	Si

5	Reconoce puntuación nivel anterior en nivel actual. En el sistema de estrellas	Alta	Si
6	Error en el modo pelea con varios NPCs del mismo grupo.	Alta	Si
7	Después de cada ataque activar de inmediato la paleta de estado del jugador y npc	Alta	Si
8	Mejoramiento de instrucciones para mejorar la mecánica de juego (tabla de instrucciones)	Alta	Si
9	Desactivar botón caratula al regresar al menú principal (menú principal)	Baja	Si
10	Error de colisión entre el jugador y el npc en la transición de cuadrícula a cuadrícula en los niveles. (personaje, npc)	Alta	Si
11	Cuando se inicia partida de cero, los niveles siguen desbloqueados. (menú niveles)	Alta	Si
12	Cuando salvas una partida el último nivel desbloqueado se esconde. (menú niveles)	Alta	Si
13	Modificación de tamaño y apariencia de números en las instrucciones (tabla instrucciones)	Alta	Si
14	Reemplazo de Tile set invisible para limitar escenario a Tile set bloque invisible	Media	Si

Fuente: Elaboración Propia

3.4.5. CIERRE

Esta fase contemplo tres etapas liberación de juego, evaluación del proyecto.

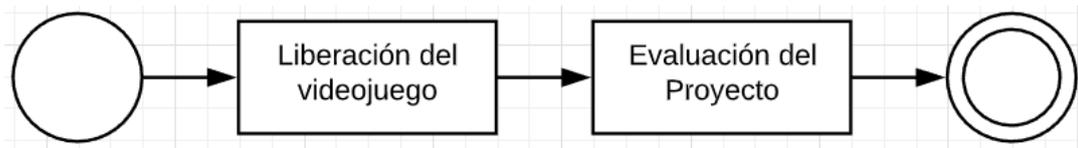


Figura 37 Flujo de Cierre del proyecto

Fuente: <http://www.gemserk.com/sum/>

Liberación del videojuego

En esta etapa se terminó con la versión final del videojuego. El flujo de este proceso se muestra a continuación.

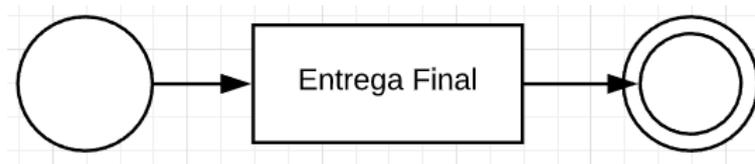


Figura 38 Flujo de liberación del videojuego

Fuente: <http://www.gemserk.com/sum/>

En este punto se liberó el aplicativo en su primera versión a la tienda virtual de Google “Play Store” para su posterior instalación en sus dispositivos móviles Android. Para lograr esto se realizó el siguiente proceso:

Generar KeyStore

Para poder subir nuestro apk al Play Store se nos exige que nuestra aplicación cuente con un KeySore único para futuras actualizaciones. Esta llave se puede generar por medio de comandos pero en este caso lo generaremos por medio del Android Studio.

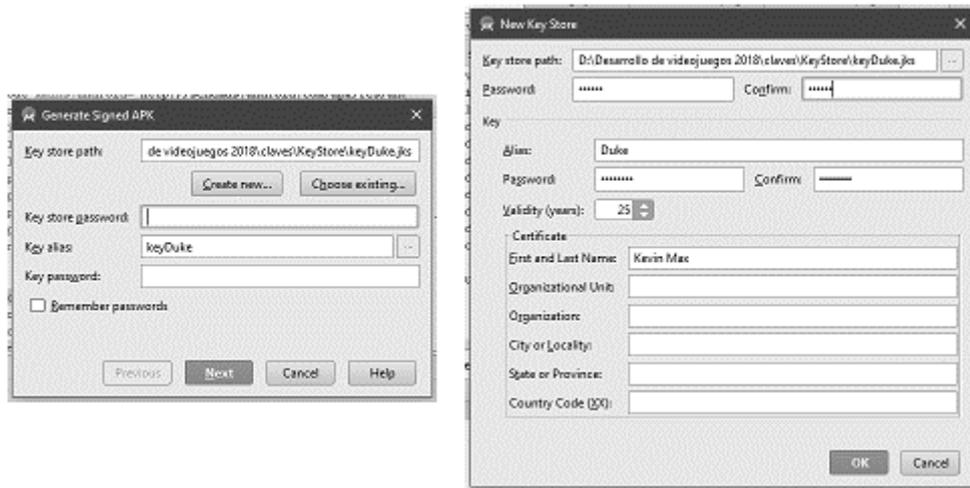


Figura 39 Interfaz creación de KeyStore

Fuente: Elaboración Propia

Exportación Android

Se especifica la ubicación del KeyStore generado, y se configura la exportación del proyecto dentro de Godot en el apartado de KeyStore ingresamos la ubicación, usuario y contraseña de nuestra KeyStore. Después se exporta el juego en formato apk desactivando la depuración.

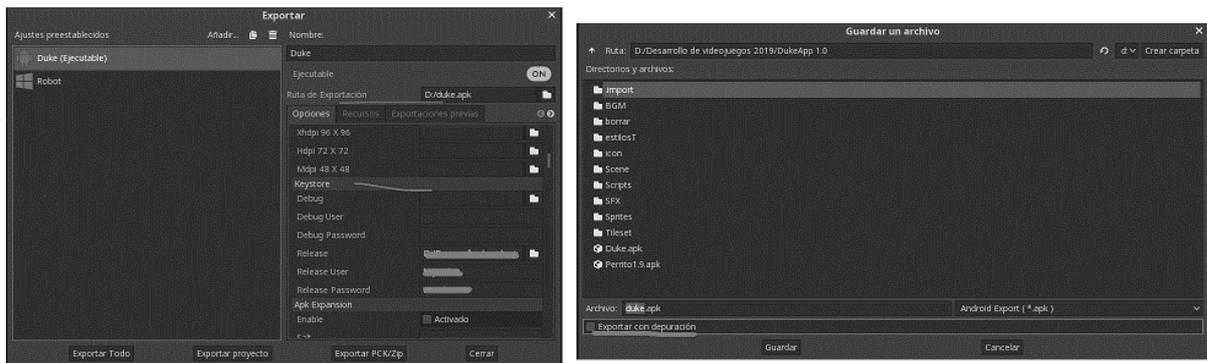


Figura 40 Interfaz de exportación para Android

Fuente: Elaboración Propia

Publicar en Google Play

Para uso del servicio de Google Play Consolé se tiene que hacer un único pago de 25 dólares, realizando este depósito ya puedes subir tus aplicaciones al Play Store. En el interfaz de “Google Play Console” creamos una nueva aplicación y llenamos todos los datos con respecto a la aplicación que nos solicitan. Luego llenamos: Ficha de PlayStore, Clasificación de Contenido, Contenido de la Aplicación y Precio y distribución. En este caso la aplicación contara con el ítem de anuncios activado ya que en posteriores versiones del videojuego se

implementaran nuevos elementos. Culminando todo esto ingresamos a versiones de la aplicación y subimos nuestra apk a “Segmento de Producción”.

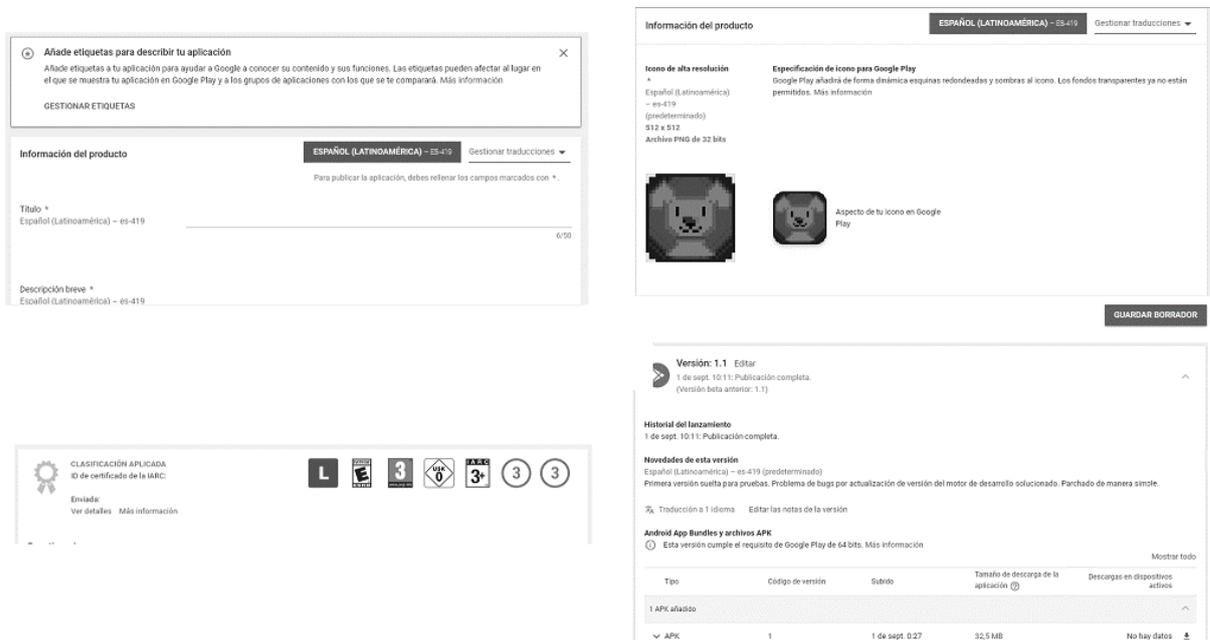


Figura 41 Interfaz de publicación de aplicación en Play Store

Fuente: Elaboración Propia

Pasando un par de días en los que Google Play revisa tu aplicación y dependiendo si cumple los parámetros la aplicación se publicara.

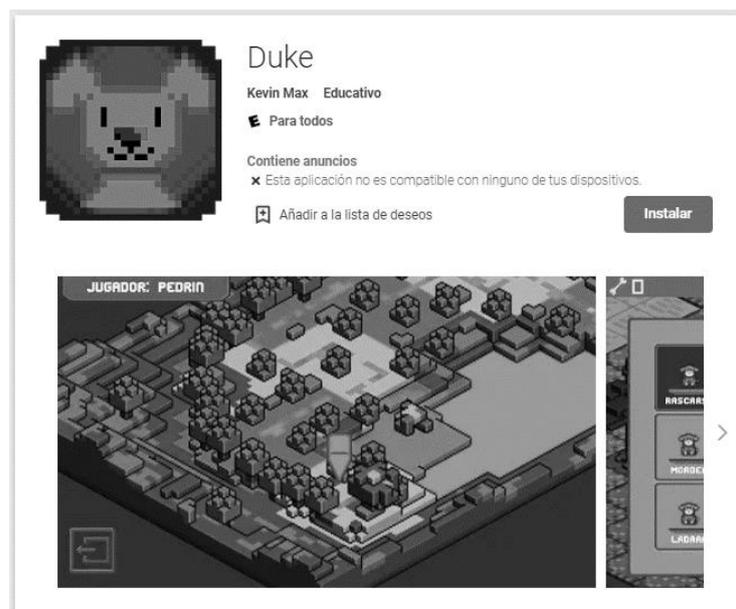


Figura 42 Interfaz aplicación publicada en Play Store

Fuente: Elaboración Propia

Con fin de que los estudiantes puedan probar el aplicativo en la institución educativa también se hizo la adaptación e instalación del software en el laboratorio de cómputo de primaria de la institución educativa. Esto gracias a las plataformas de exportación que soporta el motor de desarrollo y la adaptación de la aplicación a computadora.

La exportación para computadora es muy similar a la exportación Android. La aplicación con formato “.exe” generada es el juego ejecutable, esta versión del juego serio para pc por el momento se realizó solo para hacer las pruebas en la institución educativa.

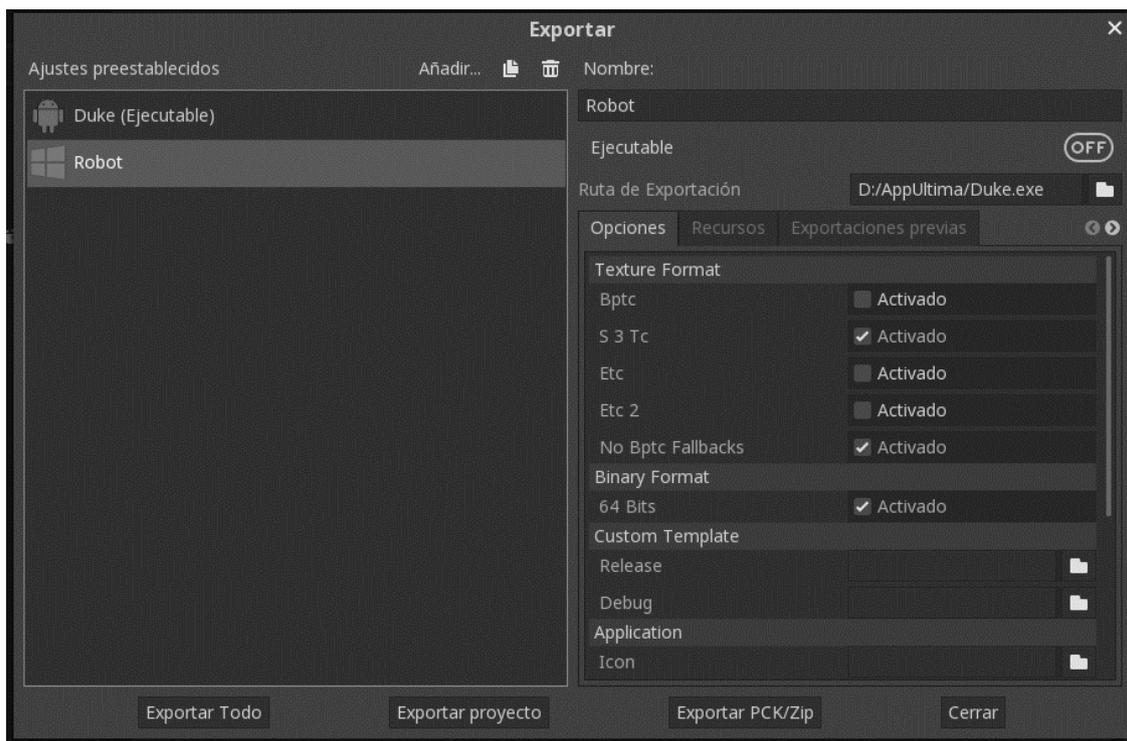


Figura 43 Interfaz de exportación a Sistema Operativo Windows

Fuente: Elaboración Propia

Evaluación del Proyecto

En esta tarea se registró los acontecimientos más importantes en el desarrollo del proyecto. Los cuales son lecciones aprendidas, recomendaciones, dificultades y logros realizados.

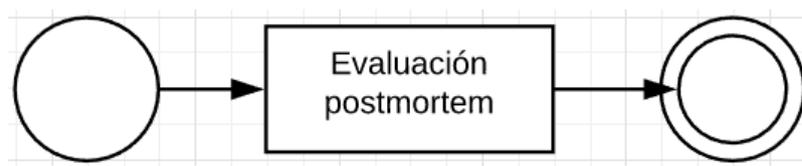


Figura 44 Flujo de Elaboración del Proyecto

Fuente: <http://www.gemserk.com/sum/>

Lecciones aprendidas, dificultades, logros y recomendaciones:

- El proyecto no siempre se realiza de manera correcta al cronograma debido a factores internos y externos.
- Hacer un presupuesto solo cuenta como referencia a los gastos que se pueden realizar en el proyecto, esto puede cambiar por el cambio de moneda.
- El desarrollo de un videojuego requiere de un equipo especializado para cada área. arte gráfico, diseñador de mecánicas, sonora,
- En el proceso de programación se presentan muchos errores ya que la interacción entre los elementos creados y el jugador puede ser muy variada y compleja de acuerdo a la mecánica del juego.
- A diferencia de un software normal el videojuego no puede presentar errores o por lo menos estos no deben afectar la experiencia de inmersión en el juego. Si esto se pierde el juego fracasa en su objetivo principal el entretener.
- Realizar testeos de cada versión del juego es fundamental para evitar posibles errores perjudiciales en el futuro.
- Es muy importante comprobar desde el principio de desarrollo los dispositivos en que funcionara el juego para de esta manera realizar los ajustes necesarios mucho antes de tener la versión final.
- Se recomienda el aprender el lenguaje ingles ya que gran parte de la documentación del lenguaje de programación y la comunidad que desarrolla profesionalmente este tipo de software hablan este idioma.
- La publicación de software educativo para niños en la plataforma “Play Store” toma mucho tiempo para evaluar el software y pasarlo a su publicación.

Gestión de Riesgos

La gestión de riesgos se llevó a cabo desde la fase uno con el fin de considerar todos los posibles riesgos en el desarrollo de cada fase y poder edificarlos antes de que sea un problema para su eliminación o su control

- Identificar Riesgos, identificamos todos los posibles riesgos que podamos encontrar y evaluamos posibles soluciones para mitigar los riesgos

Monitorear Riesgos, damos un seguimiento a los posibles riesgos de acuerdo a su nivel de impacto.

3.4.6. LIBERACIÓN EN LA INSTITUCIÓN EDUCATIVA

Prueba del Juego Serio

Con el propósito de evaluar el interés de los estudiantes sobre el juego serio. Se hizo la liberación del videojuego en el Colegio Adventista del Titicaca donde se trabajó con los estudiantes del primer grado de primaria. Esta actividad fue realizada dentro del horario de clases con la autorización de la dirección de nivel primaria de la institución. Tomando en cuenta que los estudiantes no tienen acceso a celulares o tabletas en el horario de clases se realizó la adaptación e instalación del videojuego educativo en las computadoras del laboratorio de cómputo de nivel primario. Como se muestra en la siguiente ilustración.



Ilustración 1 Ejecución de la aplicación en PC

Fuente: Elaboración Propia

El tipo de encuesta que se tomó es de “opinión pública”, ya que lo que se necesita saber es la opinión de los estudiantes sobre el videojuego con el que están interactuando y cuantitativa con el propósito de que sean fáciles de entender y responder, la encuesta fue realizada por diecisiete alumnos y adaptada para que los estudiantes de primero de primaria no tengan dificultades en entender y responder. Las preguntas realizadas fueron lo más breves posibles y con un lenguaje fácil de entender.

Los estudiantes contaron con la asistencia de un docente para brindarles ayuda si tenían algún problema con respecto a la encuesta. Las preguntas realizadas en la encuesta fueron las siguientes:

- ¿El juego es divertido?
- ¿Te gustaría jugar este tipo de juegos?
- ¿Has aprendido algo nuevo con el juego?
- ¿Entendiste cómo funcionaba el juego?
- ¿Crees que el juego es fácil de jugar?
- ¿Te gustan los colores del juego?
- ¿Te gustan los personajes del juego?
- ¿Qué te gusto más del juego?

Los pasos que se siguieron para realizar esta prueba fueron los siguientes:

- Paso 1: Se hizo jugar el videojuego por un periodo mínimo de 15 minutos a los estudiantes de primer grado de primaria.



Figura 45 Aplicación del juego serio en el Colegio Adventista Titicaca

Fuente: Elaboración Propia

- Paso 2: Se hizo llenar el cuestionario a los estudiantes.

Encuesta para estudiantes de Primero de Primaria del Colegio Adventista Titicaca

Mi nombre: _____

Mi Edad: _____

Encuesta de Para estudiantes de Primer grado de primaria

Marca la carita más feliz si quieres decir *sí* o la carita triste si quieres decir que *no*.

1. ¿El juego es divertido?

2. ¿Te gustaría jugar este tipo de juegos?

3. ¿Has aprendido algo nuevo con el juego?

4. ¿Entendiste cómo funcionaba el juego?

5. ¿Crees que el juego es fácil de jugar?

6. Te gustan los colores del juego?

7. Te gustan los personajes del juego?

8. Que te gusto más del juego.

|

Figura 46 Cuestionario para estudiantes

Fuente: Elaboración Propia

Las encuestas realizadas pueden verse en el anexo F Encuestas estudiantes Primero de Primaria.

Proceso de Encuestas

Por medio de la encuesta realizada a los estudiantes se pudo identificar el nivel de interés que se tiene a los juegos serios orientados a la educación. Para realizar el proceso de las encuestas daremos un valor a cada respuesta de la siguiente manera:

Tabla 21
Representación de respuestas de la encuesta a números

Respuesta	Representación por números
Muy Mala	1
Mala	2
Regular	3
Buena	4
Muy buena	5

Fuente: Elaboración Propia

En base a la tabla anterior realizamos la enumeración de respuestas por pregunta y sacamos el porcentaje que representa las preguntas del cuestionario, con el objetivo de identificar el interés de los estudiantes.

Tabla 22
Nivel de interés de los estudiantes para el juego serio

Pregunta	Muy malo	Malo	Regular	Bueno	Muy bueno
Pregunta 01	0	0	1	2	14
Pregunta 02	1	0	0	5	11
Pregunta 03	0	0	2	5	10
Pregunta 04	1	0	1	4	11
Pregunta 05	0	0	3	3	11
Pregunta 06	0	1	3	2	11
Pregunta 07	0	1	5	2	9
	2%	2%	13%	19%	65%

Fuente: Elaboración Propia

De manera gráfica los resultados sería los siguientes.

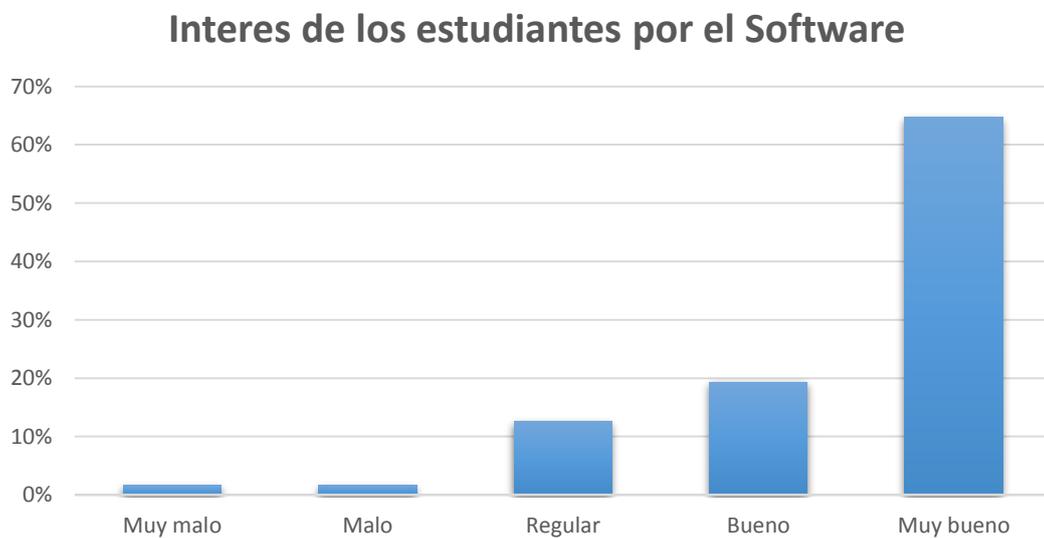


Figura 47 Nivel de interés de los estudiantes para el juego serio

Fuente: Elaboración Propia

De los estudiantes de primer grado de primaria que realizaron la encuesta, el 84% tienen interés en el juego serio de “bueno a muy bueno”, representando también a más del 50 % de estudiantes que participaron en la prueba. Demostrando que el videojuego educativo es de interés de los estudiantes de primero de primaria.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Resultado Uno

Basándonos en la metodología de Pérez Marques se identificó las competencias del área de matemática que usaremos para el desarrollo de un software educativo, pero en este caso se definió los pilares principales para el desarrollo del juego serio. Las competencias identificadas son las siguientes:

- Cardinal hasta 20
- Suma y resta hasta 20 (ejercicios combinados)
- Desarrollo de estrategias Heurísticas
- Multiplicación Básica

Estas competencias fueron tomadas del plan curricular peruano publicado en el año 2017 para nivel primario de las instituciones educativas ubicadas en zonas urbanas (minedu, 2017). De esta manera definimos el desarrollo del software a un propósito educativo.

4.1.2. Resultado Dos

Para el desarrollo del juego serio se hizo uso de la metodología SUM orientada al desarrollo de juegos. La primera fase de la metodología SUM se encarga del pre diseño del videojuego donde se indica la mecánica de juego y otros elementos que definen el tipo de juego y su estructura, todos estos elementos se consideran para el desarrollo del software, esta fase se llama “concepción” y para el desarrollo particular de este software todo el diseño de concepción se hace en base a las competencias de matemática identificadas por las etapas de la metodología de Pérez Marques. El resto de fases propias de la metodología SUM se realizaron sin ninguna alteración. Como se muestra en la siguiente imagen en el desarrollo de características.

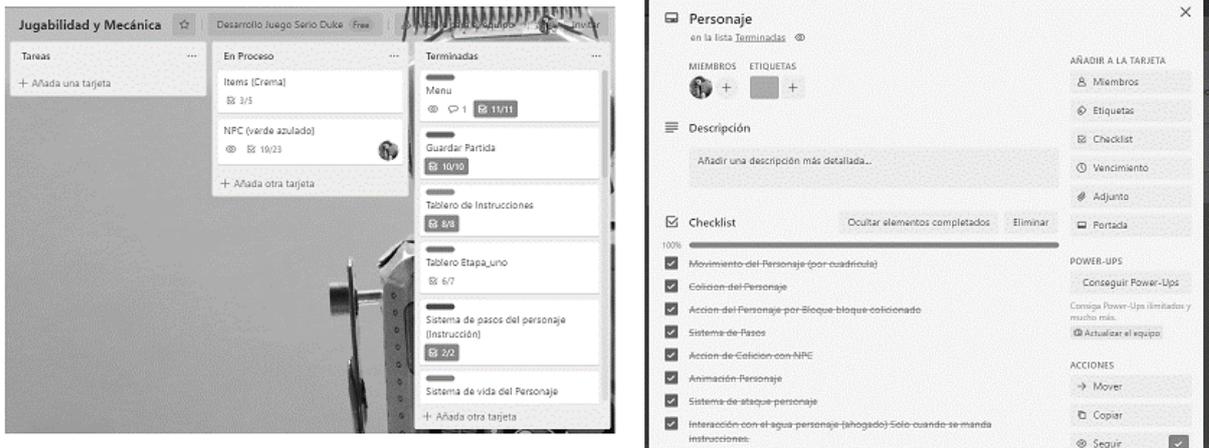


Figura 48 Administración del desarrollo de características

Fuente: Elaboración Propia

La aplicación de la metodología SUM en el desarrollo del videojuego fue fundamental para contemplar todo el desarrollo del software como también identificar posibles errores en cada interacción antes de su lanzamiento, otras de las características de la metodología permitió la adaptación y modificación del videojuego en pleno desarrollo de la característica para adaptar elementos o solucionar percances que no se identificaron en la planificación de la interacción dentro de la fase de elaboración, sin afectar el desarrollo de otras características.

Sin embargo los elementos desarrollados que no corresponden a la programación como son: el arte conceptual, el desarrollo del guion, etc. Son elementos que fueron desarrollados en base a la fase del concepto pero dentro del documento de diseño GGD.

4.1.3. Resultado Tres

El desarrollo del software fue terminado dando como resultado su primera versión 1.1 como se muestra en las siguientes imágenes.



Figura 49 Apariencia del Juego Educativo Duke

Fuente: Elaboración Propia

Cada una de las características del videojuego contemplo una construcción que no solo vario en el desarrollo sino en los tipos de nodos y herramientas propias del motor de desarrollo utilizados, dando como resultado los siguientes elementos.

Menú

Tabla 23

Cantidad de Líneas de cada script dentro de la escena Menú Principal

Nombre de Script	Cantidad de Líneas de Código
Menú	57
Ajustes Icons	12

Fuente: Elaboración Propia

El menú principal cuenta con tres nodos principales: control y caratula.

- Control: el nodo control que contiene el fondo del menú principal como también los botones del menú: empezar juego, continuar juego anterior y salir del juego. Y el registro del jugador.
- Caratula: contiene un botón que abarca toda la pantalla con el propósito de presentar el videojuego.

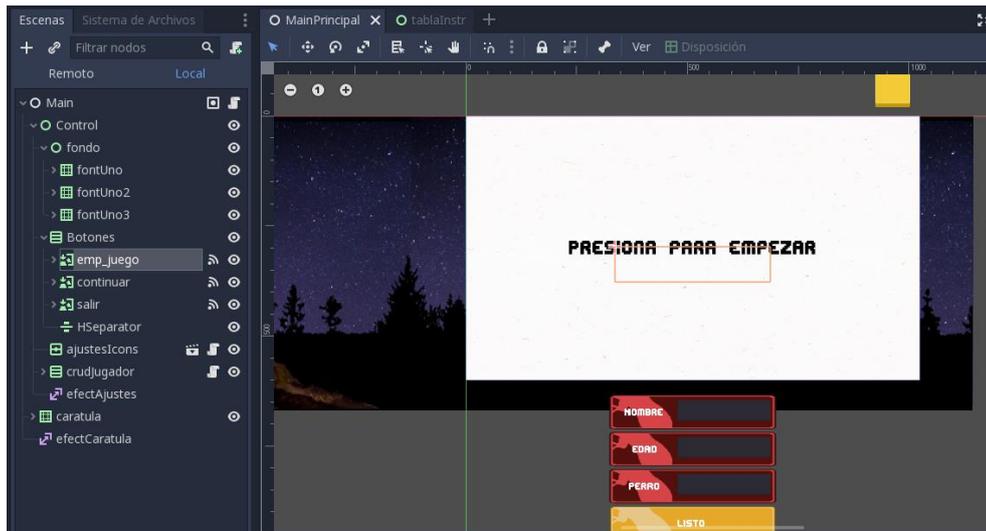


Figura 50 Nodos del menú principal

Fuente: Elaboración Propia

El botón Empezar Juego habilita el registro del jugador para empezar una nueva partida, para que el jugador pueda comenzar el juego o desee reiniciar el juego para empezar desde el principio; para continuar la partida anterior está el botón de “Continuar” que recrea el juego en el anterior estado guardado de forma automática. El botón “Salir” interrumpe el proceso del juego y lo cierra.

Menú Niveles

Tabla 24

Cantidad de líneas de cada script dentro de la escena Menú Niveles

Nombre del Script	Cantidad de líneas de Código
Main Nivel	134
Puntero	23

Fuente: Elaboración Propia

El menú niveles contiene todas las características establecidas para el versión 1.1 del videojuego, los elementos desarrollados para esta versión son los siguientes:

Sistema de desbloqueo de niveles por nivel superado

Al momento de ingresar a la escena “Menú niveles” se realiza la activación de niveles desde uno al resto de niveles de acuerdo al valor numérico del contador de niveles. Cuando el jugador logra terminar un nivel, se comprueba si el nivel superado es mayor al valor numérico del contador de niveles, si el nivel es mayor al contador de niveles entonces este se actualiza.

Mostrar datos del jugador en la parte superior de la pantalla

Por medio de los datos almacenados en el registro jugador, se muestra el nombre del jugador en el menú niveles.

Adaptación de botón “Ingreso de nivel” a nivel escogido

Cuando se selecciona el nivel al que se desea ingresar, el botón encargado de ingreso al nivel cambia la escena nivel al nivel escogido.

Movimiento aleatorio de elementos de fase

Como parte de la animación de la escena menú niveles, existen 3 elementos que tienen una animación la cual se reproduce cada ‘x’ botones presionados dentro de la escena, este valor numérico ‘x’ se escoge aleatoriamente después de cada animación reproducida.

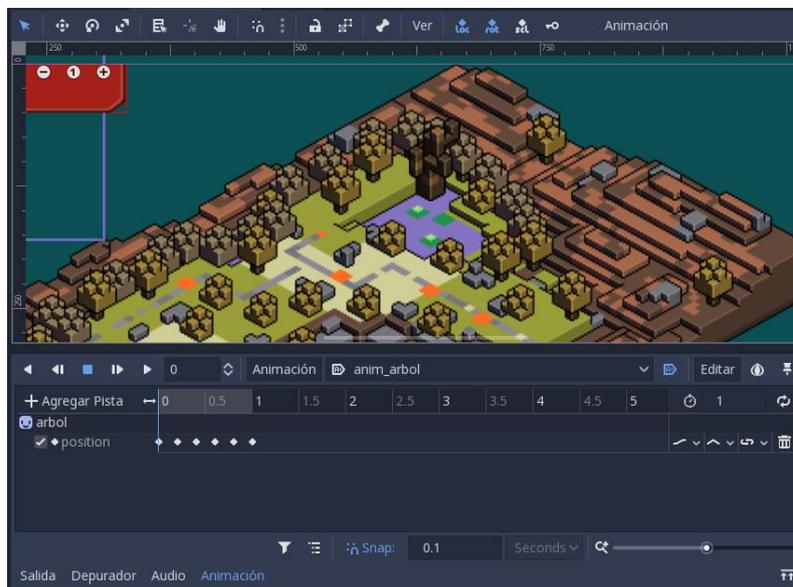


Figura 51 Animación de elementos del menú niveles

Fuente: Elaboración Propia

Registro Jugador

Tabla 25

Cantidad de líneas de cada script dentro de la escena Menú Principal

Nombre Script	Cantidad de Líneas de Código
Registro Jugador	17

Fuente: Elaboración Propia

Cuando se realiza el registro del jugador y se confirma los datos registrados se guardan los datos almacenados y se reinicia todos los datos almacenados dentro del videojuego.

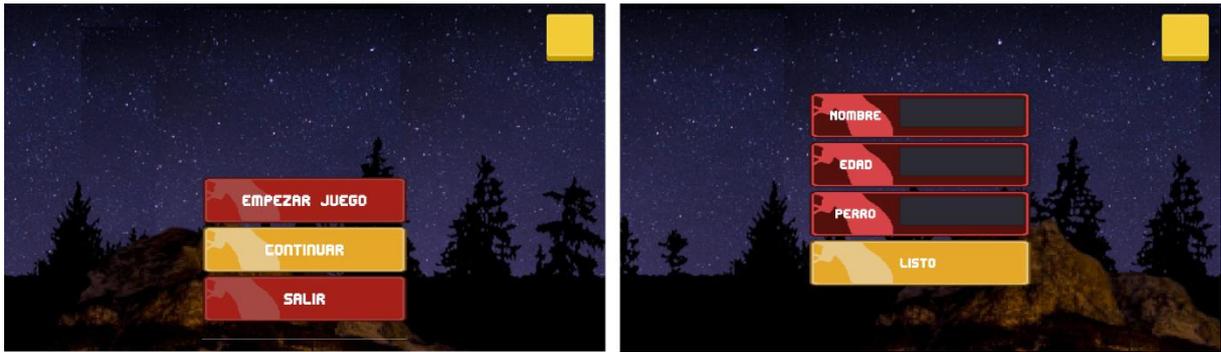


Ilustración 2 Menú Principal, Registro del jugador

Fuente: Elaboración Propia

Guardar Partida

Para guardar el avance del jugador dentro del juego se almacenan un grupo de variables las cuales son: nombre del jugador, edad, mascota, monedas y último nivel. Estas variables son almacenadas dentro de un json el cual se va actualizando cada vez que se supera un nuevo nivel y cuando el jugador termina el registro de datos. Si el jugador inicia una partida todos los datos almacenados se borran.

Si existe una partida guardada cuando se inicia el juego automáticamente el botón de continuar se habilitara, cuando existe una partida guardada y se ingresa al menú de niveles automáticamente de acuerdo al catador de niveles. Se mostrara los niveles ya desbloqueados.

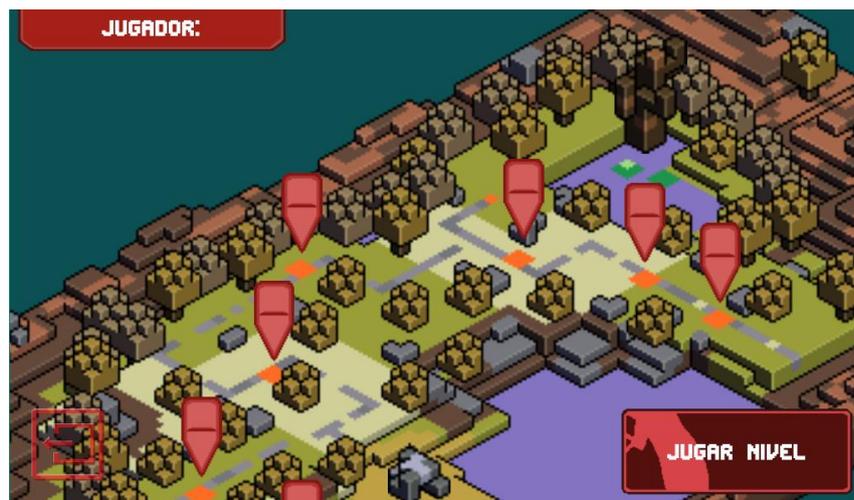


Figura 52 Identificación de niveles desbloqueados por partida guardada

Fuente: Elaboración Propia

Paleta Nivel

Tabla 26

Cantidad de líneas de cada script dentro de la escena Paleta Nivel

Nombre de Script	Cantidad de Líneas de Código
Paleta Nivel (main)	136
HUD	123
Segunda Pantalla	12
Sistema Pelea	159
Pausa	50
Cam_salto	51

Fuente: Elaboración Propia

La paleta nivel es una de las escenas más importantes del videojuego, esta escena es el marco de todos los niveles del videojuego y viene a ser la escena en la que son instanciadas la mayoría de escenas del juego como: tablero de instrucciones, personaje, nivel superado, menú pausa, etc. Contiene cuatro nodos y dos instancias:

Escena

En este nodo se carga como instancia el nivel escogido dentro del menú de niveles o en el caso de que el cambio de nivel sea por nivel superado cambia por salto de nivel.

GUI

Este nodo se encarga de los controles virtuales del juego para la interacción del jugador como se muestra en la siguiente imagen.

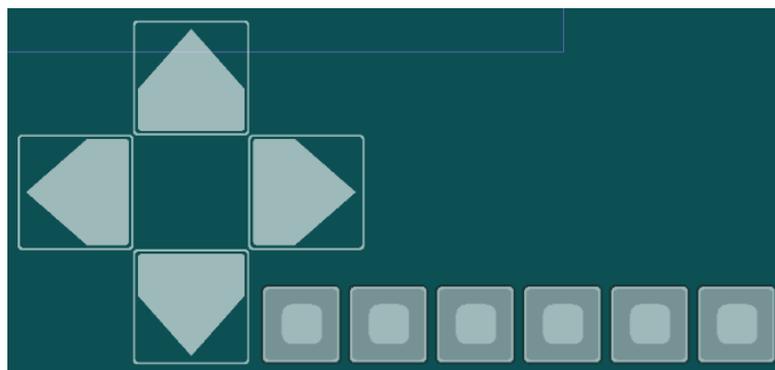


Ilustración 3 joystick virtual para el movimiento del personaje

Fuente: Elaboración Propia

Como también instancia la escena “Modo Pelea” y la escena “Menú Pausa”; estas dos escenas se encuentran instanciadas dentro del nodo GUI porque este tipo de nodo tiene la propiedad de sobreponer su contenido y nodos hijos dentro del nodo principal. Permitiendo mantener su contenido sobre los nodos hijos de “Paleta Nivel”.

SFX

Este nodo se encarga de almacenar los efectos de sonido y aplicarlos cuando sea necesario dentro del videojuego.

BGM

Este nodo se encarga de almacenar la música de fondo del videojuego y aplicarlo cuando sea necesario.

Tablero de Instrucciones (instancia)

Encargada de definir instrucciones al personaje dentro del nivel

Cámara Salto (instancia)

Este nodo almacena la ubicación de la cámara del jugador para la activación del menú instrucciones.

Tablero de instrucciones

Tabla 27

Cantidad de líneas de cada script dentro de la escena Tablero de Instrucciones

Nombre de Script	Cantidad de Líneas de Código
Tabla Instrucciones	257
Segunda Pantalla Dos	16

Fuente: Elaboración Propia

El tablero de instrucciones cuenta con tres nodos principales y un botón para regresar al escenario del nivel escogido:

- Cajas: en este nodo se encuentran las casillas donde se asignaran: tipo de sprite, tipo de acción y cantidad de veces que se repetirá la acción.
- Acciones: en este nodo se encuentran todas las acciones que se le puede asignar al personaje jugador.
- Contenido Extra: en este nodo por el momento se encuentra el botón para resetear la instrucción ya definida dentro de la casilla seleccionada.

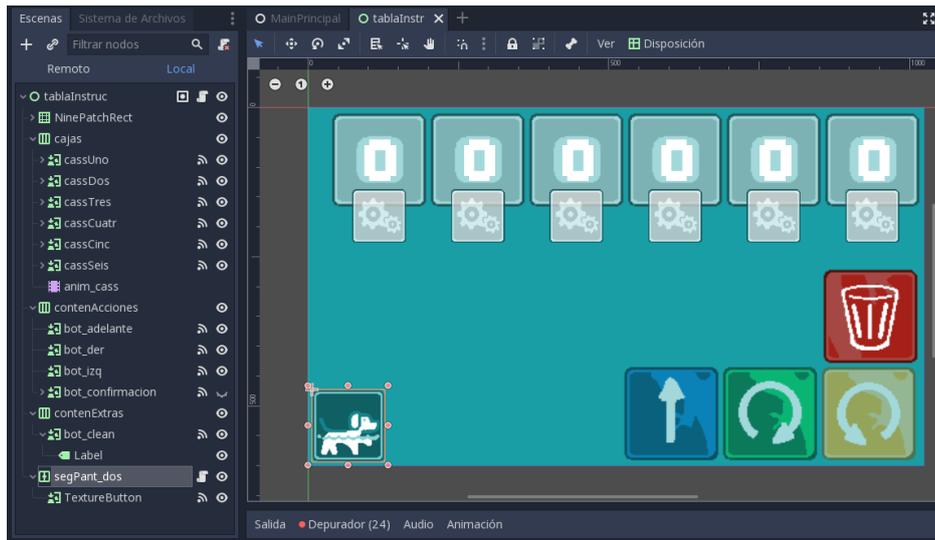


Figura 53 Nodos del tablero de instrucciones

Fuente: Elaboración Propia

Los datos que almacena el tablero de instrucciones son: casilla, acción y cantidad de veces que se repite la acción. Estos datos son almacenados dentro de un array y representan una instrucción, y esta instrucción se almacena dentro de otro array que almacena todas las instrucciones que realizara el personaje. Como se muestra en el siguiente ejemplo.

Alamacen Intrucciones = $[[InstruccionUno], [IntruccionDos], \dots, [InstruccionSeis]]$

InstruccionX = $[numCasilla, numAccion, numCantidad]$

Estos datos son introducidos por el jugador dentro del tablero de instrucciones, mientras que visualmente se muestra la modificación en la pantalla por medio de botones y sprites; el botón borrar permite eliminar los datos de la casilla seleccionada y el guardado automático del resto de instrucciones. También se guarda automáticamente las instrucciones al presionar el botón de regreso al nivel.

Personaje

Tabla 28

Cantidad de líneas de cada script dentro de la escena Personaje

Nombre del Script	Cantidad de Líneas de Código
Personaje	577

Fuente: Elaboración Propia

El nodo personaje contiene todo lo relacionado al personaje jugable del videojuego como movimientos, colisiones, estados, cámara, etc. Contemplando todos los elementos para la versión 1.1.

Movimiento del personaje

Se creó dos tipos de interacción con respecto a los controles de movimiento para el videojuego por medio de teclado y touch. El videojuego está orientado para ser liberado para dispositivos Android, pero con el propósito de realizar el testeo en pc y para realizar las pruebas con los estudiantes de primaria se realizó una adaptación para pc.

Todos los niveles tienen un diseño isométrico, para realizar el desplazamiento del personaje de manera que no pierda la estética del efecto isométrico su desplazamiento desde su punto de ubicación al punto destino en las direcciones de movimiento es la suma de las coordenadas en ambos ejes ‘x’ y ‘y’ a la ubicación del personaje.

Tabla 29
Coordenadas de movimiento para el personaje

Dirección	Coordenadas de movimiento a la dirección (x, y)
Arriba	40, -20
Abajo	-40, 20
Derecha	40, 20
Izquierda	-40, -20

Fuente: Elaboración Propia

Lectura de instrucciones

La escena “Tabla de Instrucciones” almacena temporalmente las instrucciones de movimiento y acción del personaje. En esta parte el jugador por medio de un bucle que recorre todo el array principal de instrucciones va ejecutando cada tarea que debe realizar el personaje dentro del nivel. Cuando el personaje se mueve por medio de instrucciones definidas en el “tablero de instrucciones” no identifica el tipo de colisión agua lo que le permite desplazarse sobre esta colisión. Si el personaje queda varado en el bloque de colisión agua por distintos motivos, el personaje reproduce la animación de nivel perdido y termina el juego.



Figura 54 Movimiento del personaje por instrucciones

Fuente: Elaboración Propia

Colisiones del Personaje

El videojuego cuenta con distintos tipos de objetos de colisión los cuales se identifican por el grupo al que pertenecen, la interacción del personaje en la colisión con estos objetos varía de acuerdo al grupo al que pertenecen. Para identificar los cuerpos que le colisionan la escena personaje utiliza el nodo de tipo Raycast2D el cual identifica la colisión de un cuerpo y bloquea el movimiento del personaje en esa dirección.

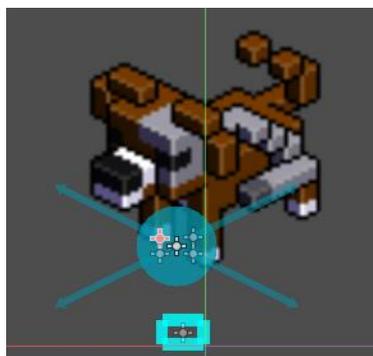


Figura 55 Nodos Raycast 2D de la escena personaje

Fuente: Elaboración Propia

Modo Pelea

Al colisionar con un personaje no jugable (npc) automáticamente se activa el modo pelea, este modo contempla 3 nodos importantes, los cuales son paleta ataques que muestra los tipos de ataque del personaje, paletas estado que muestra nombre y cantidad de vida del jugador como también un perfil. El último nodo es la paleta respuestas donde se muestra las alternativas de

respuesta matemáticas al estudiante. Todos estos nodos son instanciados dentro del nodo Paleta Nivel.

Interacción Personaje no Jugable NPC

La escena personaje no reconoce la colisión con un npcs directamente, sino la escena personaje no jugable (npc) es la que informa sobre la colisión a la escena personaje, también le informa los datos del npc (tipo, cantidad de vida y nombre del personaje)

Sistema de Corazones

El personaje en esta primera versión del videojuego contara con seis corazones para el modo pelea, estos corazones irán disminuyendo de acuerdo a la cantidad de daño sufra el perro por nivel.

Sistema de Ataques

El personaje contara con tres tipos de ataques: rascarse, morder y ladrar cada tipo de ataque es un tipo de operación matemática: suma, resta y multiplicación. El daño del ataque varia de cero a dos de acuerdo a personaje no jugador (npc) al que le es lanzado, como ejemplo se muestra la siguiente tabla.

Tabla 30
Sistema de ataque de acuerdo al tipo de enemigo

Tipo de NPC \ Tipo de Ataque	Rascarse	Morder	Ladrar
Garrapata	2	1	0
Rana	0	2	1
Conejo (Este tipo de npc no se implementó en la versión 1.1 del videojuego)	1	0	2

Fuente: Elaboración Propia

Fin del juego

Existen dos formas de perder dentro del videojuego, la primera es por medio de las instrucciones cuando el jugador comete un error y el perro después de reproducir todas las instrucciones se encuentra en el agua. Lo que causa el reinicio del nivel para que el jugador

comience otra vez y la segunda es cuando el jugador es derrotado por un personaje no jugable (npc) al quedarse sin corazones de vida en el modo pelea.

Tile Set (Desarrollo de niveles)

Para el desarrollo de los niveles se hizo uso de un nodo de tipo Tile Map que se modificó su modo a isométrico y se ajustó el tamaño de cada tile de manera que contemple una forma isométrica exacta.

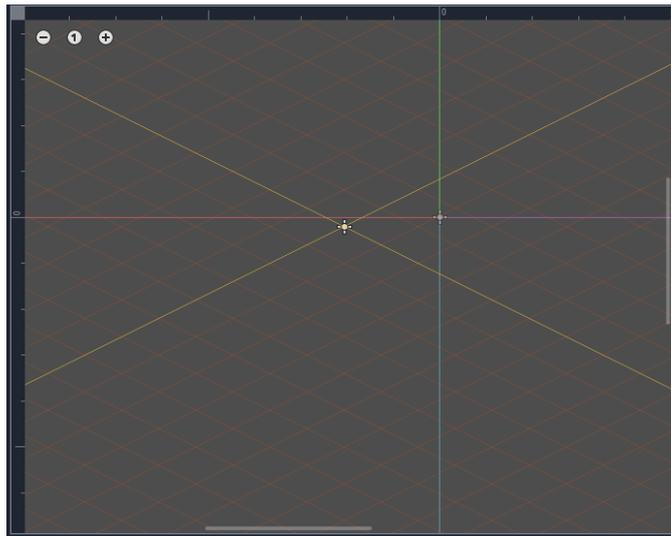


Figura 56 Ajuste de nodo Tile Map en modo isométrico

Fuente: Elaboración Propia

El desarrollo de los elementos que darán forma al nivel se realizó en una escena con el nombre “tileBloque” que contiene todos los sprites y sus colisiones: rocas, arboles, tipos de suelo, arbustos, postes y bloques.

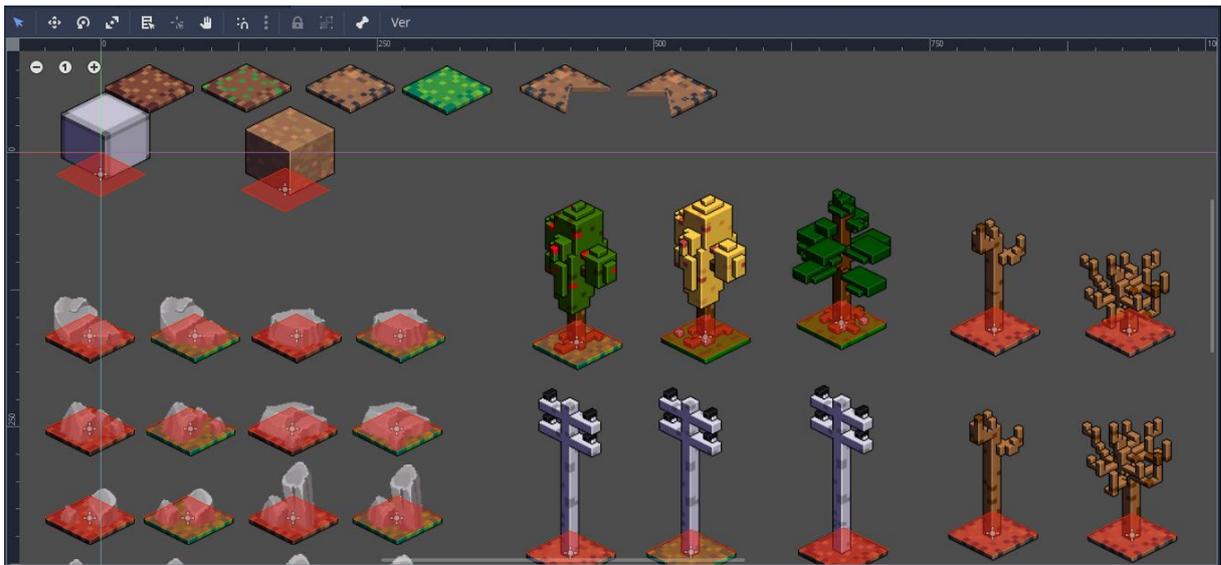


Figura 57 Escena tileBloque para la construcción del niveles

Fuente: Elaboración Propia

Esta escena “tileBloque” se convirtió en un tileSet y se integró dentro del nodo TileMap, de la misma forma se realizó con el nodo TileMap de otros elementos como bloque agua. Los elementos como: puntos, cajas, bandera, personaje, npc y otros fueron agregados por medio de instancias dentro de la misma escena del nivel. Para definir los límites de cada nivel se usó de dos nodos tipo “Position2D” para limitar el rango de visión de la cámara del jugador por medio de condiciones.

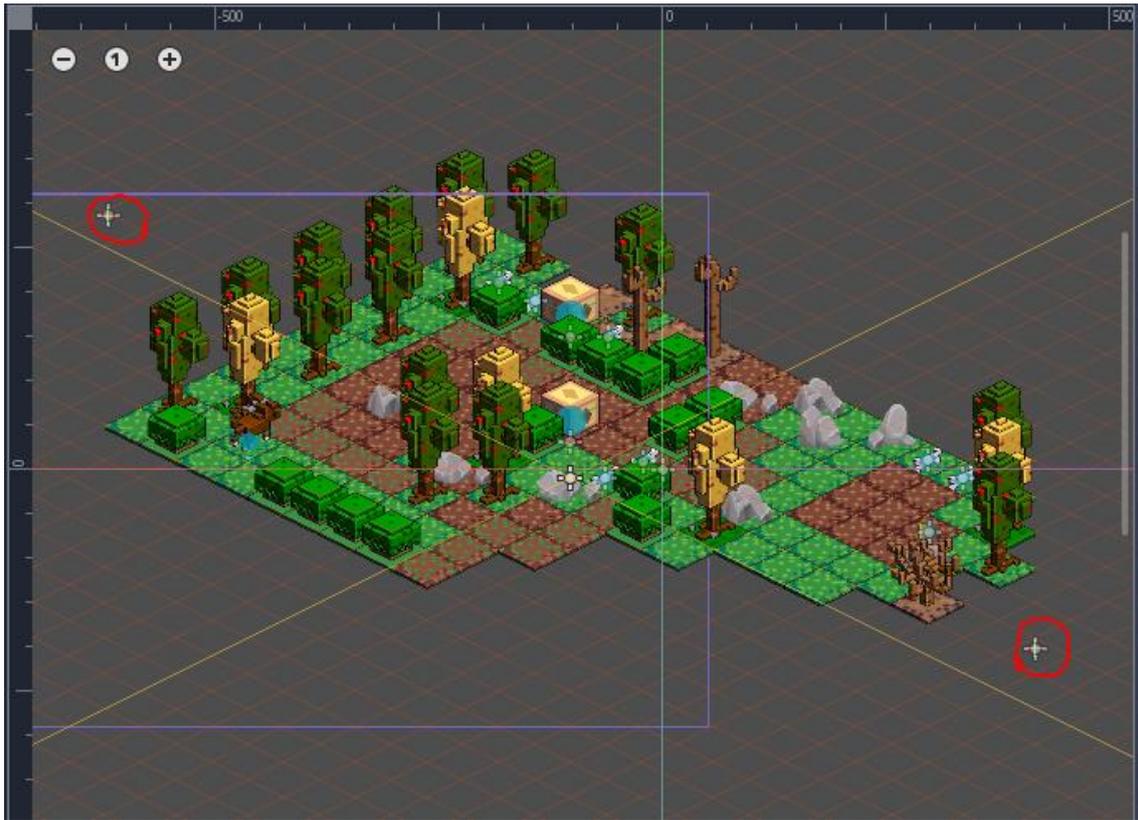


Figura 58 Limitación de desplazamiento de la cámara del jugador dentro del nivel

Fuente: Elaboración Propia

Estos son los elementos que se usó para construir cada nivel del videojuego. A partir de este punto se diseñó los niveles de acuerdo al borrador de diseño de niveles dentro del GGD. Como se muestra en la siguiente imagen los tres primeros niveles

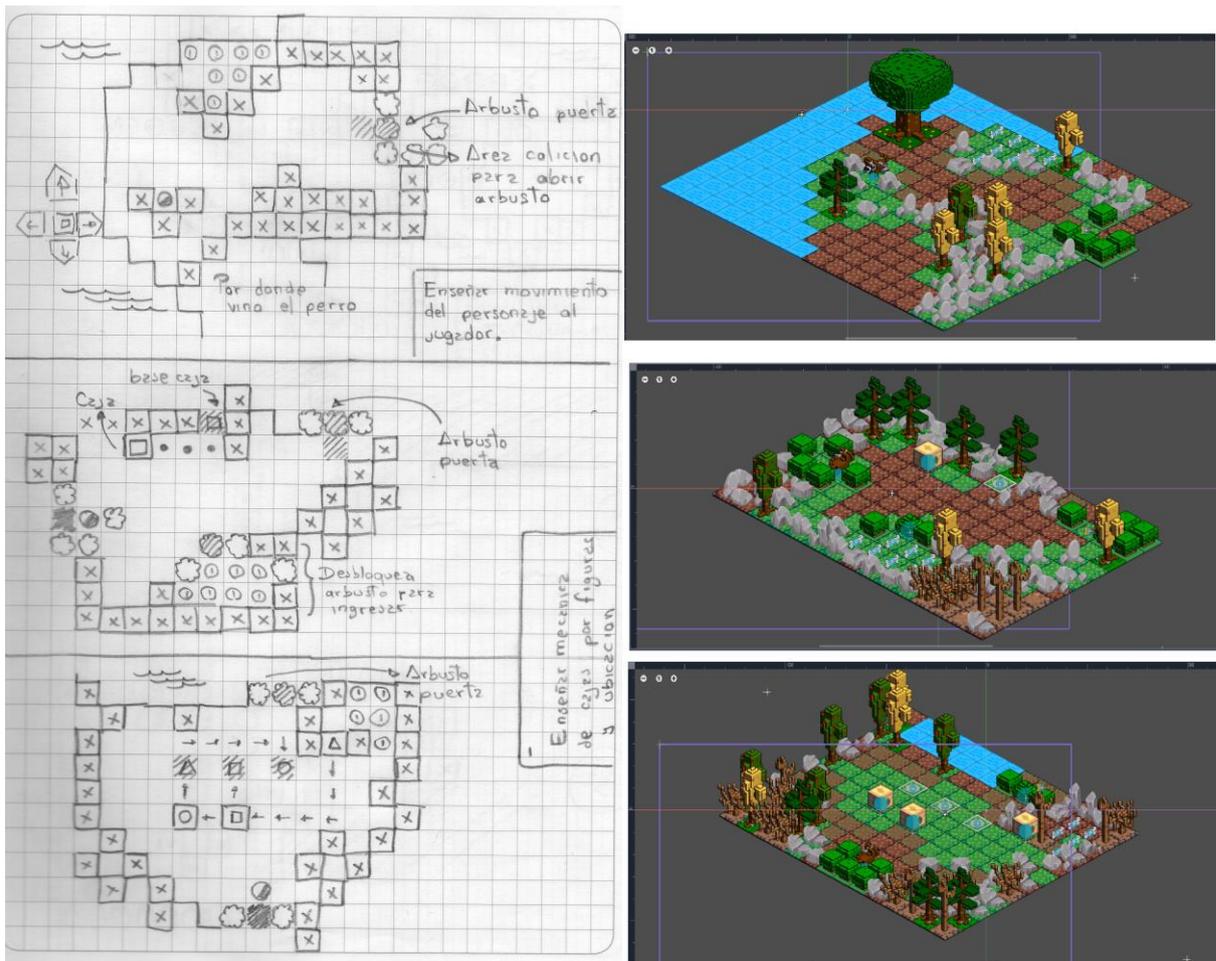


Figura 59 Construcción de niveles de acuerdo boceto de diseño

Fuente: Elaboración Propia

Objetos de Interacción

Tabla 31

Cantidad de líneas de cada script dentro de las escenas de objetos colisión

Nombre de Script	Cantidad de Líneas de Código
Caja	82
Base Caja	6
Puerta (Arbusto)	21
Bandera Fin	28

Fuente: Elaboración Propia

El videojuego cuenta con una variedad de elementos que forman parte de cada nivel del videojuego, cada uno de estos elementos son una escena diferente con sus propias características como sprites, tipos de nodos, sonidos e incluso cada una cuenta con su propio

sistema de funcionamiento e interacción con el jugador. Estos elementos dentro de la versión 1.1 son los siguientes:

Caja

Este elemento puede ser movido por el jugador en distintas direcciones y pueden ser ubicadas en bases que correspondan a la figura que pertenezcan. En el caso de que su ubicación sea incorrecta, la caja cambiara de color a rojo pero si es correcta su color sera verde y ya no se podrá mover más. Mayormente se usan para desbloquear arbustos para terminar el nivel.

Base Caja

Estas son las ubicaciones donde se acomodan las cajas, son reconocibles por la figura que representan.

Arbusto Móvil

Este elemento sirve para el bloqueo de caminos dentro del nivel. Si el jugador dentro del nivel ubica cada caja dentro de sus bases entonces este arbusto desaparece.

Bandera

Cuando el jugador tiene contacto con la bandera, automáticamente termina el nivel.

Rampas

Este elemento habilita la tabla de instrucciones del personaje para que el personaje se pueda desplazar de manera autónoma por el escenario sin respetar los bloques de agua.

Ítems

Tabla 32

Cantidad de líneas de cada script dentro de la escena puntos

Nombre de Script	Cantidad de Línea de Código
Moneda	25

Fuente: Elaboración Propia

Dentro del videojuego se cuenta con dos tipos de ítems para esta versión, las cuales son:

Huesos

Existen tres tipos de Huesos que se diferencian por el color, la cantidad de puntos que brinda son de: 1, 5 y 10. Si el jugador consigue todos los huesos de un nivel conseguirá una huella al terminar el nivel.

Personaje no Jugable (NPC)

Tabla 33

Cantidad de líneas de cada script dentro de las escenas npc

Nombre de Script	Cantidad de Líneas de Código
Enemigo	184
Garrapata	72
Rana	41

Fuente: Elaboración Propia

Esta primera versión 1.1 cuenta con dos tipos de personajes no jugables los cuales cuentan con características y diseños diferentes. Pero que a su vez comparten algunas similitudes con respecto a los nodos utilizados y otros elementos.

Garrapata

Este tipo de personaje no jugable (npc) cuenta con la característica especial de duplicar su ataque cada ataque certero lanzado en su turno y alterar su sprite. Como también si este tipo de npc es atacado su ataque se reduce menos uno y su sprite se vuelve a alterar. Si la garrapata se cambia de sprite por tercera vez desaparece y termina el enfrentamiento.

Rana

Este tipo de npc cuenta con un ataque más alto, como también una velocidad de movimiento mas alta, que permite la probabilidad de colisionar con el personaje jugable.

4.1.4. Resultado Cuatro

Por medio de la autorización brindada por la dirección del Colegio Adventista Titicaca el juego serio fue adaptado y liberado en el laboratorio de cómputo de nivel primario, para que los estudiantes de primero de primaria puedan interactuar con el software a través de una PC. Esta adaptación del software para PC solo se realizó para hacer las pruebas con los estudiantes, ya que estos no tienen acceso a un celular o tableta en el horario de clases. Como se muestra en las siguientes imágenes.



Figura 60 Liberación del Juego Serio en el Colegio Adventista Titicaca

Fuente: Elaboración Propia

Con la finalidad de conocer el interés de los estudiantes sobre el Juego Serio orientado a la educación se realizó una encuesta adaptada para estudiantes a 17 estudiantes de nivel primario que dio como resultado el grafico que se presenta a continuación.

Interes de los estudiantes por el Software

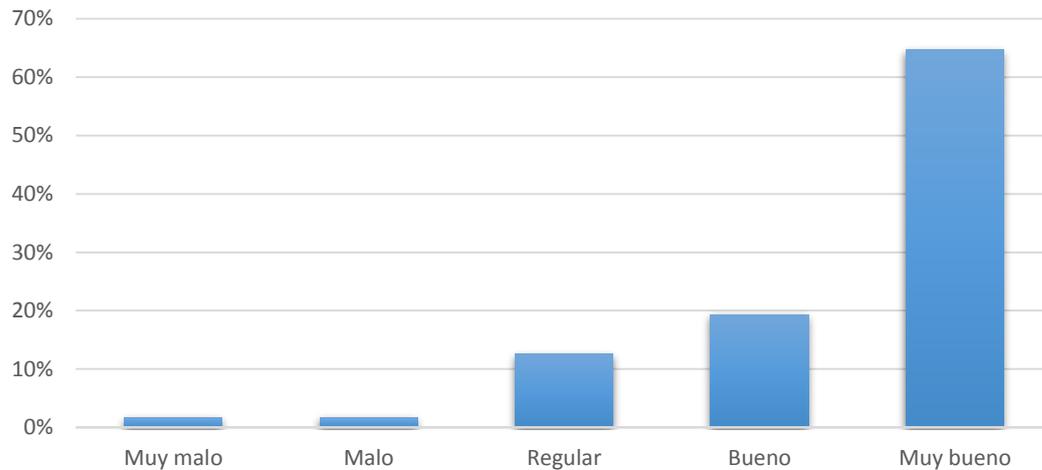


Figura 61 Nivel de interés de los estudiantes para el juego serio

Fuente: Elaboración Propia

Como se puede apreciar por medio de la medición de satisfacción: muy malo, malo, regular, bueno y muy bueno. Que el porcentaje de estudiad antes que considera que el juego era bueno es de 19% y el porcentaje de estudiantes que cree que el juego es muy bueno es de 65%. Demostrando que el juego serio es de interés para los estudiantes. El aplicativo desarrollado en esta investigación se liberó en la plataforma de distribución digital “Play Store” y es libre para su descarga. Se le puede localizar en la tienda escribiendo en el buscador “Duke educativo”.

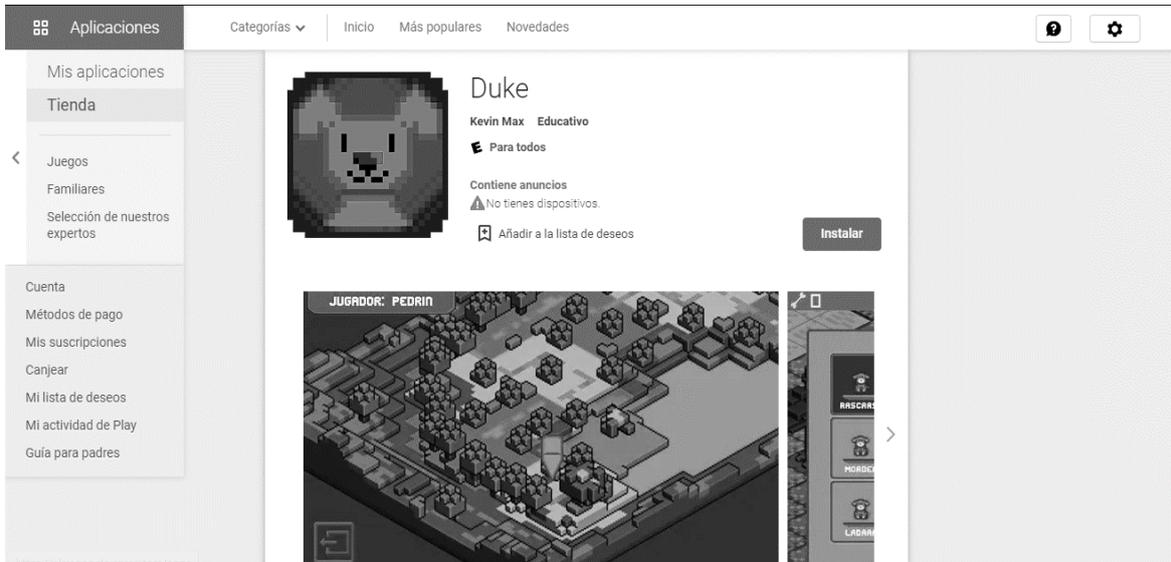
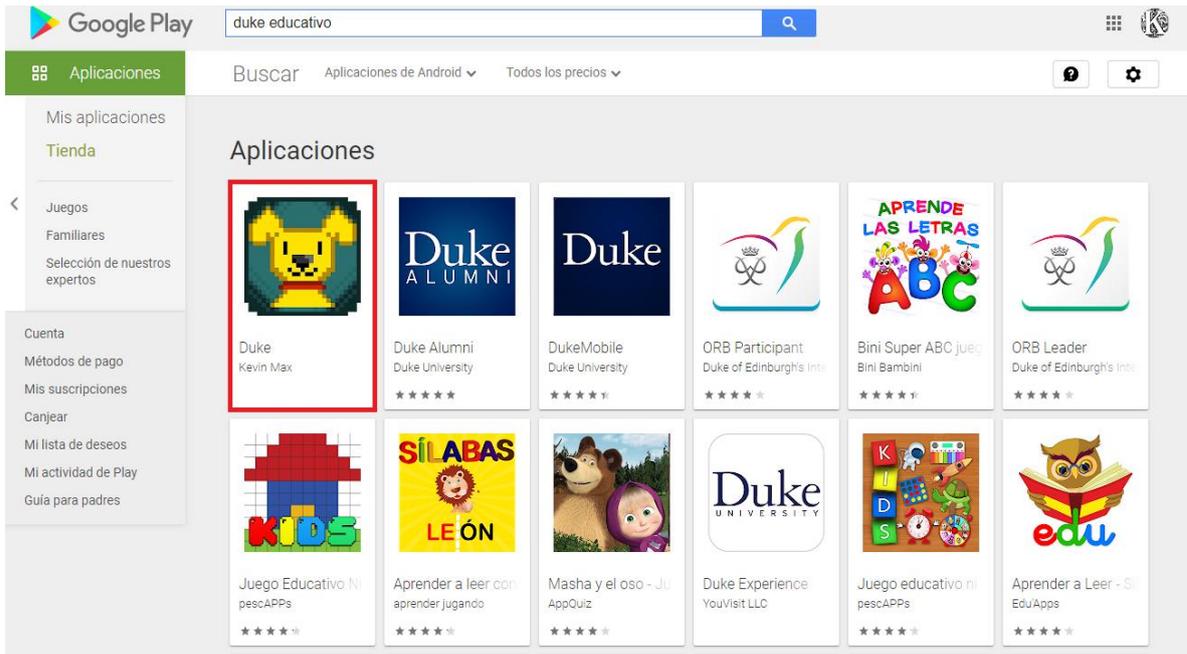


Figura 62 Juego Serio Duke en Play Store

Fuente: Elaboración Propia

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Luego de ejecutar el presente proyecto de investigación se concluye que:

- El uso de las tres primeras etapas de la metodología de Pérez Márquez orientado al desarrollo de software educativo permitió definir de manera más sólida el objetivo principal del videojuego. Permitiendo considerar estas competencias para el desarrollo del Concepto de la Metodología SUM.
- En el desarrollo de un videojuego es fundamental tener una idea clara del concepto del videojuego ya que en esta es donde se desarrolla la manera en la que el jugador interactúa con el juego, la historia que se contara al jugador, la razón del porque se realizan tales acciones, en simples palabras lo que define al juego como divertido o aburrido, para eso se hizo uso del documento de diseño GGD, donde se definió a detalle todas las características que tiene que cumplir el juego desde la mecánica, diseño de niveles, historia hasta los efectos de sonido del videojuego.
- Gracias al Motor de desarrollo Godot es posible exportar el videojuego a distintas plataformas con el mismo código fuente, tanto en tiendas virtuales como también directamente a los teléfonos o tabletas. Esta facilidad del motor de desarrollo permitió la exportación del videojuego en las computadoras de escritorio del laboratorio de la institución educativa.
- Usar los videojuegos como herramientas educativas permite al estudiante interactuar con el aprendizaje educativo de manera más divertida y agradable permitiéndole interactuar y buscar soluciones de problemas a su manera. Esto facilita la comprensión y asimilación de nuevos conceptos y contenidos aprendidos en clase.

5.2. RECOMENDACIONES

Con la finalidad de realizar trabajos de investigación similares al detallado en este documento, así como trabajos futuros, se recomienda lo siguiente:

- Desarrollar un videojuego requiere de un equipo especialista en cada área para que el proyecto sea de calidad. Ya que el desarrollo de un proyecto por cuenta propia o con solo un grupo de personas con la misma especialidad no permitirá un trabajo fluido y de paso un proyecto de calidad.

- La adaptación de dos metodologías es entendible de acuerdo al tipo de software o videojuego que se quiera realizar, ya que el propósito de las metodologías es poder facilitar el desarrollo del software del proyecto.
- Es importante plantear el alcance de tu proyecto de acuerdo a las herramientas y conocimientos que se tiene para evitar futuros problemas y complicaciones a futuro. En caso contrario es importante identificar cuáles son las herramientas o conocimientos que se adolecen para considerar su desarrollo y que este no sea un problema a futuro.
- Usar imágenes y elementos visuales con una calidad de gráficos necesaria, que no exceda su tamaño innecesariamente como también usar archivos de audio tipo OGG ya que estos ocupan menos espacio y además Godot soporta ese formato. Todo esto con la finalidad de que la aplicación cuando se ejecute consuma la menor cantidad de recursos del dispositivo móvil.
- Si se pretende desarrollar un videojuego en el Motor de Desarrollo Godot, se recomienda utilizar la versión 3.1.1 o superior que sea estable, esto para evitar problemas con el nivel del API que exige Google Play para publicar aplicaciones.
- Aprender el idioma inglés ya que gran parte de la documentación para cualquier lenguaje de programación, motor de videojuegos o proyectos profesionales relacionados al desarrollo de videojuegos de manera profesional se encuentran en ese idioma.
- El desarrollo de juegos serios orientados a desarrollar otras competencias.

CAPÍTULO VI. LISTA DE REFERENCIAS

- Abraham, G., Urrutia, M., Nava López, C. E., Felipe Fernández Martínez, L., Aarón, M., & Corral, R. (2010). PROCESOS DE DESARROLLO PARA VIDEOJUEGOS, 39. Retrieved from <https://www.mendeley.com/viewer/?fileId=4e9cf50a-1d13-018f-69df-bd3728e1b7cf&documentId=dfb0450a-8350-348f-9155-2b3c924a6a7b>
- AUSUBEL, D. (1983). TEORIA DEL APRENDIZAJE SIGNIFICATIVO, 10. Retrieved from https://s3.amazonaws.com/academia.edu.documents/38902537/Aprendizaje_significativo.pdf?response-content-disposition=inline%253B%2520filename%253DTEORIA_DEL_APRENDIZAJE_SIGNIFICATIVO_TEOR.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%25
- Bourne, C., & Salgado M, V. (2016). Los videojuegos pueden transformar el aula - Aika Educación. Retrieved November 20, 2018, from <http://www.aikaeducacion.com/tendencias/los-videojuegos-transforman-aula/>
- Cardoso Espinosa, E. O., & Cerecedo Mercado, M. T. (2009). El desarrollo de las competencias matemáticas en la primera infancia.
- Coumans, E. (2015). Bullet Physics Simulation Introduction to rigid body dynamics and collision detection. Retrieved from https://docs.google.com/presentation/d/1-UqEzGEHdskq8blwNWqdgnmUDwZDPjZUvg437z7XCM/edit#slide=id.ga4b37291a_0_0
- DINEIP - DINESST. (2005). Diseño Curricular Nacional de Educación Basica Regular. Lima. Retrieved from <http://www.minedu.gob.pe/normatividad/reglamentos/DisenoCurricularNacional.pdf>
- Figueredo, O. B., & Sosa, E. A. (2016). Juego serio: modelo teórico para su diseño y producción.
- Gamelearn. (2017). Lo que necesitas saber sobre serious games y game-based learning. Retrieved November 23, 2018, from <https://www.game-learn.com/lo-que-necesitas-saber-serious-games-game-based-learning-ejemplos/>
- Gemserk. (2008). SUM para Desarrollo de Videojuegos. Retrieved from <http://www.gemserk.com/sum/>
- Gómez, B. (2018). Gamificación y Juegos Serios: ¿qué son y por qué funcionan? - T3chFest 2018 - YouTube. Retrieved November 22, 2018, from <https://www.youtube.com/watch?v=GRmbHwVArUg>
- Liniesky, J., & Manzur, A. (n.d.). De Unity a Godot Engine — documentación de Godot Engine - latest. Retrieved November 23, 2018, from https://docs.godotengine.org/es/latest/getting_started/editor/unity_to_godot.html
- Linietsky, J., Manzur, A., & Godot, C. (n.d.). Filosofía del diseño de Godot — documentación de Godot Engine - latest. Retrieved June 30, 2019, from https://docs.godotengine.org/es/latest/getting_started/step_by_step/godot_design_philosophy.html
- Mendoza, V. (2015). La matrona que salvó miles de vidas con un útero de trapo | Yorokobu.

- Retrieved November 22, 2018, from <https://www.yorokobu.es/angelique-du-coudray/>
- minedu. (2017). Currículo Nacional. Retrieved from <http://www.minedu.gob.pe/curriculo/>
- Minedu. (2014). *Marco Curricular Nacional*. Lima.
- Minedu. (2016). Educación Básica Regular Peru.
- Ministerio de Educación Perú. (2016). *Informe de evaluación matemática sexto grado de primaria*. Lima, Perú. Retrieved from http://umc.minedu.gob.pe/wp-content/uploads/2016/07/EM_Matematica_baja-2.pdf
- Nuñez Wagner, M. C. (2014). Metodologías de desarrollo de software educativo. Retrieved from https://www.academia.edu/8654429/Metodologías_de_desarrollo_de_software_educativo
- Palomino Taype, I. M. (2016). *DESARROLLO DE UN VIDEOJUEGO CASUAL MULTIPLATAFORMA PARA REDUCIR EL ESTRÉS*. UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS. Retrieved from <http://181.176.178.114/browse?value=Palomino+Taype%2C+Iván+Marcial&type=author>
- PISA. (2015). *El Perú en PISA Informe nacional de resultados*. Retrieved from http://umc.minedu.gob.pe/wp-content/uploads/2017/04/Libro_PISA.pdf
- Rafael, A. (2016). Causas y consecuencias del desinterés de la educación en alumnos de preparatoria. Retrieved from https://www.academia.edu/27987186/Causas_y_consecuencias_del_desinterés_de_la_educación_en_alumnos_de_preparatoria
- Tim Ryan. (n.d.). Gamasutra - The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal. Retrieved June 28, 2019, from https://www.gamasutra.com/view/feature/3384/the_anatomy_of_a_design_document.php

CAPÍTULO VII. ANEXOS

Anexo A. Programa Curricular de Educación Primaria

<http://www.minedu.gob.pe/curriculo/pdf/programa-curricular-educacion-primaria.pdf>

Anexo B. Documento de Diseño del Videojuego GDD

<https://drive.google.com/open?id=1uIIHJiO90cjQ1G122PWB4lwcX5Gwx70Y>

Anexo C. Etapas de la Arquitectura del Software

<https://docs.google.com/document/d/1KeX75UCru6tnk-vJN86Z-INSJmuycfRauU1f9ymF2P8/edit?usp=sharing>

Anexo D. Diagrama de Gantt

<https://drive.google.com/file/d/1MjFznLKriLKuwY0mdR-OToMUxdOyIuHw/view?usp=sharing>

Anexo E. Encuestas estudiantes Primero de Primaria

https://docs.google.com/document/d/1q1qsKG0ViMD1UZgjrc83L_7x6EublSj-V11DQuRQOig/edit?usp=sharing



UNIVERSIDAD PERUANA UNIÓN COLEGIO ADVENTISTA DEL TITICACA

INICIAL - PRIMARIA - SECUNDARIA

“Año del Diálogo y la Reconciliación Nacional”

AUTORIZACIÓN

El Director de la Institución Educativa “Adventista del Titicaca” del distrito de Juliaca, provincia de San Román y departamento de Puno; de la UGEL – San Román de la Dirección Regional de Educación de Puno, quien suscribe;

HACE CONSTAR:

El estudiante Kevin Maximiliano Arenas Cancapa, identificado con DNI N° 72225446, egresado de la Escuela Profesional de Ingeniería de Sistemas de la “Universidad Peruana Unión FJ”, tiene la autorización para aplicar un software como caso de estudio para la investigación titulada: “DESARROLLO DE UN SERIOUS GAMING PARA ESTUDIANTES DEL PRIMER AÑO DE PRIMARIA APLICANDO LA METODOLOGÍA SUM”, durante los meses enero y febrero de 2019.

Se expide la presente constancia a petición del interesado para los fines que vea por conveniente.

Puno, Juliaca, 05 de diciembre de 2018



Dr. Queleon Elmer Mamani Quispe
Director del Colegio Adventista del Titicaca



UNIVERSIDAD PERUANA UNIÓN
**COLEGIO ADVENTISTA
DEL TITICACA**

INICIAL - PRIMARIA - SECUNDARIA

"AÑO DE LA LUCHA CONTRA LA CORRUPCIÓN E IMPUNIDAD"

CONSTANCIA

La que suscribe, **Directora del Colegio Adventista del Titicaca** de la ciudad de Juliaca – Puno.

HACE CONSTAR:

Que, **KEVIN MAXIMILIANO ARENAS CANCAPA** con código de estudiante N° 201310751 de la EP. Profesional de Ingeniería de Sistemas, realizó su prueba del proyecto de tesis “**Desarrollo de un Serious Game para los estudiantes de Primer grado de nivel Primaria**”.

Se expide la presente constancia a solicitud del interesado, para los usos y fines que considere convenientes.

Juliaca, Chullunquiani, 10 de octubre de 2019



[Handwritten Signature]
Mg. Mirisol Chambi Mamani
Directora General
Colegio Adventista del Titicaca