

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**FERONIA:  
VIDEOJUEGO DE ACCIÓN Y  
PLATAFORMA**

**ITALO GIOVANNI CARRASCO LUCERO  
ANDRÉS ALEXIS CORTÉS BARAHONA**

INFORME FINAL DEL PROYECTO PARA  
OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO EN EJECUCIÓN INFORMÁTICA

DICIEMBRE, 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**FERONIA:  
VIDEOJUEGO DE ACCIÓN Y  
PLATAFORMA**

**ITALO GIOVANNI CARRASCO LUCERO  
ANDRÉS ALEXIS CORTÉS BARAHONA**

Profesor Guía: Ismael Figueroa Palet

Profesor Co-referente: Héctor Allende Cid

Diciembre, 2018

*Dedico esto a mis padres por haberme apoyado, ayudado y dado el soporte para llegar hasta donde he llegado. A mis hermanas en quienes he podido confiar. A mis amigos quienes han dado su disponibilidad cuando la he necesitado. A los profesores con los que cursado cada curso, sobre todo, a nuestro profesor guía quién entregó gran disponibilidad al necesitarlo.*

*-I. Carrasco*

*Dedico el presente a toda mi familia, en especial a mi madre y mi hermana, que han estado en todos y cada uno de los momentos que las he necesitado, han sido parte de la formación de mi ética y de mi comportamiento. Al esfuerzo de mi madre por el trabajo que le ha sido mantener a sus dos hijos por sí sola. A mis dos abuelos que han marcado mi camino, ayudando a tomar las decisiones correctas. Agradezco, también, a todos mis amigos que han dado soporte a mi vida, siendo un sólido apoyo. A nuestro profesor guía que, entre humor y disciplina, cooperó con todo lo que podía. Finalmente, a la universidad y compañeros, en especial a mi compañero de proyecto que en conjunto logramos complementar nuestras ideas.*

*-A. Cortés Barahona*

# Índice

Abstract .....	iii
Resumen .....	iv
Lista de Figuras .....	v
Lista Tablas .....	vi
<b>1</b> <b>Introducción.....</b>	<b>1</b>
<b>2</b> <b>Marco general del proyecto.....</b>	<b>2</b>
2.1   Marco teórico .....	2
2.2   Descripción del estado alpha.....	2
2.3   Descripción general de la situación de estudio.....	3
2.4   Objetivos .....	3
2.4.1   Objetivo general .....	3
2.4.2   Objetivos específicos .....	3
<b>3</b> <b>Situación de estudio.....</b>	<b>4</b>
3.1   Descripción de la situación actual .....	4
3.2   Problemas detectados .....	4
3.3   Estado del Arte.....	4
<b>4</b> <b>Propuesta de solución.....</b>	<b>6</b>
4.1   Descripción general de la solución .....	6
4.1.1   Habilidades .....	6
4.1.2   Mejoras.....	7
4.2   Modelo de proceso de desarrollo .....	7
4.3   Arquitectura de la solución .....	9
4.4   Herramientas de Desarrollo.....	10
<b>5</b> <b>Gestión del Proyecto.....</b>	<b>11</b>
5.1   Recursos requeridos .....	11
5.2   Estudio de factibilidad.....	11
5.3   Carta Gantt .....	12
<b>6</b> <b>Análisis de requerimientos .....</b>	<b>13</b>
<b>7</b> <b>Diseño .....</b>	<b>17</b>
7.1   Componentes de Software.....	17
7.2   Base de Datos.....	17
7.3   Interfaz.....	17
<b>8</b> <b>Implementación .....</b>	<b>19</b>
8.1   Implementación Pre Beta.....	19

8.2	Pruebas Pre Beta.....	19
8.3	Resultados Pre Beta .....	19
8.4	Implementación Beta .....	19
8.5	Pruebas Beta.....	19
8.6	Resultados Beta .....	20
8.7	Publicación .....	20
9	Conclusiones .....	21
10	Referencias.....	22
11	Bibliografía Consultada.....	23
	Anexos .....	24
	A: Diagramas utilizados en la etapa Alpha.....	25
	B: Estimación de cantidad de HH en etapa Alpha previo al desarrollo .....	30
	C: Carta Gantt.....	31
	D: Modelo de la base de datos.....	33
	E: Interfaz Gráfica.....	34
	F: Plan de pruebas .....	38

## Abstract

The purpose of this project is to continue the alpha video-game called Feronia, is a non completed video-game that required advance to the next phase, these combine two classics genres of the arcade machines. The current alpha state was developed in the “Taller de Ingenieria de Software” subject. The project Feronia wants to promote a new mixed experience for old players who knows the genres platform, beat ‘em up and RPG, players who have played on arcade in the 8 bits era. The game’s going to have an original story with original illustrations and a new mechanic of battle passing from the Platform genre to the Beat ‘em up. As final goal, is important to add new mechanics for plattform and beat ‘em up taking it to the beta phase and finally publish the game on a digital distribution platform like Steam.

---

**Keywords:** *Action, Beat ‘em up, Beta Game, Indie, Platform, Video-game*

## Resumen

El propósito del proyecto es el desarrollo beta del videojuego Feronia que actualmente se encuentra en fase alfa. Este juego integra los géneros de Plataforma, Beat 'em up y elementos de RPG. Está orientado para un público adulto joven, quienes crecieron jugando con máquinas arcade. El desarrollo de la etapa alfa se realizó en la asignatura Taller de Ingeniería de Software. La nueva etapa contempla la adición de nuevas mecánicas para el entorno de plataforma y beat 'em up. Lo esencial es que no se pierda la experiencia de mezcla entre un modo y el otro, dejar que los jugadores disfruten cada modo y sientan que las habilidades puedan ser utilizadas de manera similar en cada uno de ellos. Como meta final se busca la publicación en una tienda digital como, por ejemplo, Steam.

---

**Palabras Claves:** *Acción, Beat 'em up, Beta game, Indie, Plataforma, Videojuego.*

## Lista de Figuras

Figura 4.1 Modelo de proceso SUM.....	8
Figura 4.2 Diagrama de capas.....	9
Figura 6.1 Diagrama de escenas Godot.....	17
Figura 6.2 Modo plataforma.....	18
Figura 6.3 Modo Beat 'em up.....	18

## Lista Tablas

Tabla 4.1 Tabla de habilidades por modo .....	7
Tabla 5.1 Resumen de carta Gantt por fase.....	12
Tabla 6.1 Requerimientos Funcionales .....	13
Tabla 6.2 Requerimientos No Funcionales .....	16

# 1 Introducción

Hablar de videojuegos es similar a hablar de películas, puesto que existe una gran cantidad que se segmenta en distintos géneros, los cuales son combinables. De esta forma, el éxito de un videojuego depende de un factor abstracto, el gusto de los jugadores; hay mecánicas que para algunos jugadores son favorables o cómodas, como las hay simples y que incomodan a un jugador acostumbrado a oprimir una gran variedad de botones para hacer una acción.

Feronia es un videojuego que está orientado para jugadores que acostumbran a jugar juegos de acción en 2D, es decir, juegos plataformas que a la vez deban combatir con enemigos tanto con armas o con combinaciones de ataque cuerpo a cuerpo. Por esta razón Feronia combina dos importantes géneros, el plataforma y el beat 'em up, géneros populares en los años 90 en máquinas de arcade y que por lo mismo es donde apunta, a jugadores que en promedio tengan 25 años, quienes estén dispuestos a comprar o probar juegos independientes que les ofrezcan nuevas características.

La fase alpha del proyecto fue desarrollada en la asignatura Taller de Ingeniería de Software donde se implementaron las mecánicas más importantes para la jugabilidad, tales como el comportamiento de los enemigos y el traspaso de modo plataforma a beat 'em up. Sin embargo, no posee una completitud para el mercado de los videojuegos, ya que sólo alcanzó a tener un escenario o nivel a superar, no posee historia ni ninguna mecánica que diferencie al juego.

En el caso del proyecto que se presenta, se espera complementar la fase alpha de Feronia, además de replicar y generar nuevas mecánicas que sean integrables en los géneros beat 'em up y plataforma junto con una historia coherente y elementos de Role Playing Games (RPG) para que el jugador sienta un avance en sus niveles y atributos de su personaje.

## 2 Marco general del proyecto

### 2.1 Marco teórico

En el presente informe se tratará sobre la implementación de mecánicas para un videojuego en 2D a través de técnicas en la programación, por lo que es preciso aclarar algunos conceptos. Para la elaboración de un videojuego, por lo general, se utilizan frameworks, motores y librerías que apoyan la programación, y que permiten aplicar el paradigma orientado a objetos, el cual es un pilar fundamental en este tipo de desarrollo, tal como lo menciona el diseñador de videojuegos Tracy Fullerton en su libro “Game Design Workshop: A playcentric approach to creating innovative games”[1], los objetos son lo básico de un sistema de videojuegos, ya que es aquí donde se definen los parámetros o atributos que van a tener los elementos, se establecen propiedades de objetos abstractos que pueden ser utilizados, por ejemplo, para dar una jerarquía indefinida de enemigos o de ítems. De estos, es posible generar propiedades específicas que son comprobables en la ejecución, además de comportamientos que obligan a realizar acciones cuando se cumpla alguna condición en sus propiedades; adicionalmente, es importante tener una clara concepción de la relación entre los objetos instanciados en el videojuego.

### 2.2 Descripción del estado alpha

En la fase alpha, el videojuego llegó a contar con una estructura básica, donde se desarrolló principalmente el comportamiento de los enemigos, un nivel simple de plataformas y un nivel de batalla beat’em up. Con respecto a los enemigos se tiene un comportamiento básico el cual cumple con parámetros de búsqueda para seguir al jugador y atacar. El motor Godot 3 permitió generar un árbol de decisiones en base a nodos conectados y funciones escritas en GDscript [2]. En relación al modo plataforma, se presenta un nivel tutorial y un nivel de saltos en plataforma donde si el personaje colisiona con enemigos se cambia al modo beat ‘em up, además hay monedas coleccionables que permiten al jugador comprar ítems a un único vendedor puesto al final de la escena tutorial. Acerca del modo beat’em up, se tiene que el personaje gana monedas y experiencia al derrotar a sus enemigos en combate, cabe destacar que hasta ese momento ganar experiencia no generaba ningún beneficio para el jugador.

Se debe mencionar que en el videojuego no existe un sistema de persistencia de datos, por lo que al cambiar entre una y otra escena no se mantiene actualizada la información del personaje (vida, experiencia, monedas). Tampoco existe un sistema de guardado y carga de la partida por lo cual siempre se debe comenzar desde el inicio del juego.

La estética del videojuego apunta a lo retro, por lo que es completamente en 2D, los personajes, los ítems y los fondos animados son instanciados en Godot dándole de antemano los movimientos con un editor dentro del mismo motor. Cada imagen utilizada en el videojuego es original, pero, en la fase alpha, sólo se encuentra la ambientación de un bosque (un nivel jugable).

## 2.3 Descripción general de la situación de estudio

A pesar de que la industria de los videojuegos es considerada por muchos entretenimiento, donde no parece ser un mercado rentable más que para las compañías de altos presupuestos (AAA), la escena indie, juegos casuales y plataformas de distribuciones abiertas podrían refutar. A menudo la gente juega juegos en su smartphone para distraer la mente o acabar con su ocio; otros, en cambio, aburridos de mecánicas repetitivas, buscan nuevas opciones en desarrolladores independientes; en cualquier caso, los desarrolladores que hacen estos pequeños juegos o crean nuevas mecánicas suelen utilizar bajos presupuestos, publicando sus juegos en plataformas como Steam, App Store, Play Store, etc. y considerando que en promedio pueden vender unas veinte mil copias, hace que en realidad no sea un mal mercado[3].

En el consiguiente proyecto se pretende proseguir el desarrollo de un videojuego indie que se encuentra en fase alpha, el cual consiste en combinar los géneros de beat 'em up y platform conocidos en la industria [4] agregando nuevas mecánicas.

Para la siguiente fase se planifica ampliar las mecánicas en el modo plataforma, agregar nuevos enemigos en el modo beat 'em up y llegar a un nivel de público que pueda testear el videojuego, para posteriormente lograr la distribución a través de una plataforma digital.

Para el desarrollo del videojuego se utilizará Godot Engine versión 3.0, ya que como es de código abierto, la comunidad va mejorando y agregando nuevas funcionalidades, arreglando bugs y optimizando el rendimiento del motor [5]. Para la codificación utiliza GScript que permite un desarrollo ágil enfocado en el trabajo en equipo gracias al sistema de escenas de Godot [6].

## 2.4 Objetivos

### 2.4.1 Objetivo general

Publicar en una plataforma digital de distribución, la fase beta del videojuego Feronia que actualmente se encuentra en fase alpha de desarrollo, agregando nuevos elementos para conformar un producto completo.

### 2.4.2 Objetivos específicos

- Agregar las mecánicas de doble salto, activar y desactivar plantas y duplicado en el modo plataforma para que el jugador supere obstáculos o logre objetivos secundarios del juego.
- Desarrollar un sistema de niveles de experiencia que le permita al jugador desbloquear habilidades y subir sus estadísticas.
- Implementar un guión que entregue una historia dentro del juego.
- Agregar animaciones e ilustraciones que se adecuen a la historia del juego.
- Desarrollar un mínimo de 4 fases que el jugador deba completar.

### **3 Situación de estudio**

Feronia está orientado a un público joven-adulto, de entre dieciocho y treinta años, jugadores que ya hayan experimentado los géneros implementados y que noten las referencias al ambiente retro de las máquinas arcade.

#### **3.1 Descripción de la situación actual**

Actualmente el videojuego Feronia se encuentra en una fase alpha, esto quiere decir que está incompleto y no es apto para el mercado. El videojuego no posee ningún sonido ni música ambiental; posee dos niveles, uno es el nivel tutorial y el otro es un nivel que posee varias plataformas y solamente un tipo de enemigo repetido en varias ocasiones; contiene dos combos distintos de tres golpes cada uno; además de que no posee una escalabilidad de niveles, o sea, no puede subir sus atributos o ganar nuevos poderes. En sí, el videojuego posee el traspaso de plataforma a modo pelea (o beat 'em up), donde los enemigos tienen secuencias para actuar dependiendo del estado en que se encuentre el personaje jugador, pero es insuficiente sólo un patrón de movimiento, sólo un enemigo y sólo un tipo de escenario.

#### **3.2 Problemas detectados**

El videojuego posee falencias al pasar datos de una escena a otra (del nivel tutorial al nivel uno) al no almacenar las variables en ninguna base de datos perdurable. Además, carece de mecánicas llamativas que se escapen de lo visto en cada género al que el video juego apunta (plataforma, beat 'em up y RPG). Al poseer solamente un tipo de enemigo y sólo un escenario el videojuego se torna repetitivo, además la secuencia que siguen los enemigos responde a patrones que a la larga se vuelven predecibles.

#### **3.3 Estado del Arte**

Existe una gran cantidad de videojuegos y actualmente siguen publicando más cada día, no sólo existe el mercado de distribución AAA (que es la industria de videojuegos de alto presupuesto), si no que los desarrolladores independientes han crecido en desarrollo, llevándose una cantidad importante del público. Por estas razones es difícil contemplar la totalidad de videojuegos y negar la existencia que cumpla con lo que se propone lograr en éste, pero es posible indagar con los videojuegos que comercialmente han tenido éxito y comparar las mecánicas que individualmente están involucradas.

En primer lugar, los videojuegos de plataformas han tenido un extenso crecimiento y segmentación, la referencia más conocida de plataforma es Super Mario Bros. (1985), en este se presenta el clásico sistema donde el jugador sólo debe saltar y avanzar. Muchos otros videojuegos han logrado unir mecánicas bajo el mismo sistema, un ejemplo es Metroid (1986) o Megaman (1987) donde los personajes disparaban en todo momento a sus enemigos y podían adquirir mejoras. Para el sistema beat 'em up o modo pelea, se encuentran, en su gran mayoría, videojuegos de acción, donde el personaje tiene que llegar hasta el final (avanzando

hacia la derecha) moviéndose en ocho direcciones y al encontrarse con enemigos, no podrá seguir avanzando al menos que los elimine a todos. Los personajes en estos juegos tienen combinaciones de ataque que son similares a los videojuegos de lucha (fighting games), entre algunos ejemplos encontramos a Double Dragon (1987), Final Fight (1989) o Golden Axe (1989).

Con respecto a las herramientas de desarrollo de videojuegos, en el caso de la escena indie se utilizan con mayor frecuencia Unity y Unreal Engine, estas son recomendadas gracias a la facilidad con la que se pueden comenzar a desarrollar un proyecto, la inmensidad de tutoriales y libros existentes que explican desde cómo programar en los lenguajes soportados por estos motores, desarrollar mecánicas específicas o incluso hasta nos muestran paso a paso cómo desarrollar un videojuego simple. A la vez, estas herramientas también son recomendadas debido a que ambas presentan sistemas para desarrollar un videojuego sin saber programar, Blueprints en el caso de Unreal y Visual Script en el de Unity.

Las posibles plataformas de publicación son las siguientes:

Itch.io: un marketplace web para la publicación de juegos indie gratuitos o de pago, su uso es gratuito, pero permite a los creadores que decidan vender sus videojuegos, donar cierto porcentaje de sus ganancias, si lo desean, para poder mantener los servidores y sus servicios. soporta distintos formatos de videojuegos web como unity, flash, html5[7].

Steam: es una plataforma de distribución digital de videojuegos más usadas en computadoras con 18.5 millones de usuarios concurrentes [8], permite publicar videojuegos al pagar una cuota de ingreso de \$65.000[9] y pasar una revisión del software comprobando si funciona correctamente [10].

## **4 Propuesta de solución**

### **4.1 Descripción general de la solución**

La solución ante la incompletitud de la fase alpha es llevar el videojuego a una etapa más completa, donde el videojuego tenga al menos tres niveles fuera del nivel tutorial, al menos un enemigo final (conocido en los videojuegos como el jefe final), al menos tres tipos de enemigos distintos y unos tres escenarios distinguibles visualmente que complementen los niveles. Esto, ante la monotonía y falta de diversidad en el juego; con respecto a las mecánicas, se deben agregar nuevos combos y acciones en el beat 'em up que también puedan ser utilizadas en el modo plataforma, un ejemplo sencillo es el doble salto: una habilidad que permite saltar una vez cuando ya estás en el aire. De estas nuevas mecánicas tienen que existir al menos tres, cada una desbloqueable en un nivel. Debe implementarse un sistema donde al acumular cierta experiencia el nivel del personaje se incremente y le permita aumentar los atributos de salud, resistencia o ataque.

El videojuego, en esta ocasión debe presentar un sistema de diálogos que le permita al usuario seguir la historia en la cadena de acontecimientos, esto se traduce en un sistema lineal tal como son utilizados en los videojuegos de novelas visuales.

Por último, para mantener la persistencia de los datos entre escenas (ítems que se adquieran, monedas o la misma salud del personaje), deberán ser guardados en una variable global de tipo diccionario y cargados cada vez que se cambie de escena, por otro lado, para guardar y cargar una partida, estos datos serán escritos en un archivo tipo json con una estructura similar al diccionario, para posteriormente ser leídos por el juego y se cargue una partida anterior desde el menú principal.

#### **4.1.1 Habilidades**

Las habilidades, como se mencionó deberían ser mecánicas que permitan al personaje jugador (PJ) realizar proezas tanto en el modo plataforma como en el modo beat 'em up. Para esto deberá completar los escenarios y avanzar. Cada escenario debe dar al usuario una nueva habilidad, para esta fase se han pensado en tres mecánicas:

Tabla 4.1 Tabla de habilidades por modo

Habilidad	Descripción
Doble Salto	Permite al PJ saltar una vez al estar en el aire.
Duplicado	Permite al PJ crear un clon fantasmal de sí mismo, el cual consigue atravesar objetos que normalmente no podría.
Trampas	El PJ podrá activar o desactivar trampas que le impidan el avance

### 4.1.2 Mejoras

Las mejoras son obtenidas al subir de nivel o por la compra de un ítem, esto es, juntando la cantidad de experiencia según necesite el PJ o comprando un arma con un vendedor dentro del videojuego. Los atributos mejorables son:

- **Salud:** Llamada en otros juegos hit points (HP), es un atributo numérico que de ser cero indica la muerte del PJ. Se puede traducir como la capacidad máxima a la que el PJ puede recibir daño. Al aumentar su salud, podrá recibir más golpes antes de morir. La salud debe aumentar en un punto cada dos niveles.
- **Ataque:** Este atributo es directamente aplicado en el daño que causa el PJ sobre el enemigo, por lo que su aumento llevará al PJ facilitarle la victoria contra hordas de enemigos. El ataque es aumentado mediante ítems.

## 4.2 Modelo de proceso de desarrollo

La metodología SUM basada en Scrum presenta etapas iterativas donde es necesario tener feedback con usuarios al que el software apunta, esto para poder mejorar jugabilidad [11]. La base sigue siendo el desarrollo y comunicación con el cliente, en este caso los usuarios, tal como en Scrum, para poder avanzar a la siguiente iteración.

El modelo utilizado para planificar el desarrollo del software toma, tal como se mencionó, como base la metodología ágil SUM, para lo que es necesario utilizar, al menos, una iteración en las fases intermedias: Planificación, Elaboración y Beta. En estas fases se debe involucrar las etapas anteriores del proyecto en estado alpha, para así continuar con el concepto dado en las fases tempranas del desarrollo.



Figura 4.1 Modelo de proceso SUM

Implementado a lo que es necesario para el desarrollo de la fase beta y siguiendo el modelo, las fases estarían dadas de la siguiente forma:

- **Definición:** Antes de la primera etapa de Planificación se deberá incluir el marco teórico y los nuevos problemas detectados con su respectiva documentación.
- **Primera planificación:** Para esta etapa se deberá hacer uso de los requerimientos y factores que influyen en la la definición de nuevos. Armar ideas y soluciones.
- **Elaboración pre beta:** Se incluirá todo lo planificado y se codificará lo necesario, se agregarán los elementos necesarios para cubrir los requerimientos nacidos en la etapa de planificación.
- **Pre Beta:** Se deberán probar las nuevas actualizaciones en el software, detectar errores y bugs del videojuego.
- **Segunda planificación:** Se deberán recolectar los errores nacidos de la fase pre beta y analizar las soluciones posibles, agregar requerimientos nuevos de ser necesario para cumplir con la correcta funcionalidad del videojuego
- **Elaboración beta:** En esta etapa se deben implementar todas las correcciones que se planificaron en la etapa anterior
- **Beta:** Se deben hacer las últimas pruebas de software para corroborar su funcionamiento.
- **Cierre:** Finalización de la fase beta, exportación de un modelo ejecutable y jugable. Conclusiones y proyecciones.



Es necesario que el videojuego presente el patrón de diseño Singleton para restringir la instanciación de los atributos globales del juego y del jugador, esto es para lograr hacer las llamadas a los atributos desde cualquier escena que sea necesaria sin recargar y perder los datos de avance mientras se ejecuta el software.

## **4.4 Herramientas de Desarrollo**

Se proseguirá con el motor Godot 3, con el cual fue desarrollada la fase alpha. Al estar programado en GDScript, mantiene mucha logística propia del lenguaje, por lo que es difícil cambiar de motor sin tener que generar un código nuevo. Sin embargo, Godot 3 ofrece bastante versatilidad a la hora de programar y, por su comunidad, es fácil tener acceso a respuestas de dudas que podrían surgir a la hora de desarrollar código en una aplicación que utilice este motor [12].

## 5 Gestión del Proyecto

Para lograr los objetivos, es importante saber si es posible lograr lo propuesto teniendo en cuenta el alcance de la fase del proyecto, esto quiere decir que las operaciones, recursos y la factibilidad deben estar a favor de los tiempos de desarrollo y ser planificados con anterioridad.

### 5.1 Recursos requeridos

Los recursos necesarios para el desarrollo de la fase beta del proyecto no varían mucho de la fase alpha, puesto que se seguirá trabajando bajo el mismo modelo, utilizando las mismas herramientas de desarrollo.

De todas maneras, se debe tener en cuenta que esta versión va orientada a llegar al mercado, por lo que en los pasos finales del proyecto debe ser empleado recursos para la distribución y publicidad. Dentro de las plataformas de distribución, la más tentativa viene siendo Steam, como el candidato donde debe ser publicado el juego, para esto no es necesario ningún recurso externo.

Por otro lado, la publicidad es necesaria y es factible utilizar medios sociales como Facebook e Instagram para llegar a un público previo, para esto se necesitan ilustraciones del videojuego y encargados de mantener la atención del público tal como un social-media.

### 5.2 Estudio de factibilidad

La factibilidad es, en varios aspectos, dada por las limitaciones que surgen a la hora del desarrollo. Primero, frente a lo económico se puede estimar un gasto que considera en mayor medida las horas dedicadas al proyecto. En la fase alpha, estas estimaciones se hicieron considerando el tipo de trabajo realizado como el de un profesional, por lo cual las horas hombre que fueron calculadas eran altas. Esto, fuera de ser un problema, ayudó a estimar un precio para el cálculo de venta del producto considerando las horas utilizadas por los integrantes dadas en la carta Gantt (véase el anexo B). Para esta etapa beta, debiese reformularse dichos costos teniendo en cuenta las horas reales que fueron utilizadas en la fase anterior y el costo de HH para estudiantes y no profesionales. Aun así, bajo el análisis económico de los gastos en la fase alpha, el videojuego no debiese superar un valor de venta de cinco mil pesos chilenos (CLP).

En segundo lugar, las limitaciones técnicas, como el caso de las herramientas de desarrollo, no competen en una gran dificultad de superar, puesto que las librerías de Godot (motor por el cual se desarrolla el software) son de acceso abierto y la comunidad está atenta a peticiones y solucionar problemas. El problema es cuando surgen inconvenientes no expuestos y cuya solución solamente puede recaer en una actualización del motor, ya en la fase alpha se comenzó a utilizar la versión Godot 3 del motor, la cual lleva poco tiempo en funcionamiento.

Por último, dentro de los problemas que pueden surgir durante el desarrollo del software, está el de limitarse a no utilizar ningún recurso externo, ya que se debe tener estricto cuidado de no pasar a llevar los derechos de autor de los creadores originales.

### 5.3 Carta Gantt

Como anteriormente se mencionó, se consideró el modelo SUM para asignar las etapas y con ello repartir las tareas. Bajo esto se asignaron fechas para lograr tener una mejor visualización sobre lo que se ha trabajado. A continuación, se presenta una tabla con la integración de los puntos de fases que tendrá que atravesar el proyecto en su desarrollo.

Tabla 5.1 Resumen de carta Gantt por fase

Nombre de la fase	Fecha de inicio	Fecha de fin
Revisión del estado alpha	23/08/2018	27/08/2018
Revisión del estado actual del motor	27/08/2018	30/08/2018
Propuesta de solución	30/08/2018	04/09/2018
Etapas de desarrollo 1	05/09/2018	03/10/2018
Etapas de desarrollo 2	04/10/2018	28/11/2018

En el anexo C se presenta la carta Gantt completa con las actividades de cada fase.

## 6 Análisis de requerimientos

Los requerimientos funcionales y no funcionales para esta fase del proyecto se presentan a continuación en la siguiente tabla:

Tabla 6.1 Requerimientos Funcionales

Requerimientos Funcionales			
ID	Requerimiento	Descripción	Prioridad
RF01	Generar ejecutable	El videojuego debe tener un archivo ejecutable para ser utilizado.	1
RF02	Sistema de diálogos	Dentro del videojuego debe existir un sistema diálogos que dinámico, donde se pueda controlar los personajes a dialogar.	3
RF03	Secuencia de diálogos	En los diálogos, mediante avance la conversación, debe existir un hablante y un oyente; el hablante debe aparecer con sus respectivos colores inalterados, en cambio la ilustración del oyente debe estar con bajo brillo.	3
RF04	Combinación de ataque	El personaje debe lograr hacer combinaciones de ataque con las teclas S y D, en específico las siguientes combinaciones: S, S, S ; S, S, D ; D, D, D ; D, D, S	1
RF05	Esquivar en el modo beat 'em up	El personaje debe lograr moverse de forma invulnerable al presionar la tecla Shift para esquivar a los enemigos en el modo beat 'em up.	2
RF06	Asignar monedas tras victoria	Al completar una fase beat 'em up, deben asignarse monedas al personaje que dependan de la cantidad y tipo de enemigo.	2
RF07	Asignar experiencia tras victoria	Al completar una fase beat 'em up, debe asignarse la experiencia correspondiente al personaje y que dependa de la cantidad y tipo de enemigo.	2
RF08	Salir tras perder en modo beat 'em up	Debe existir una opción al perder en el modo beat 'em up que permita cerrar el juego.	1
RF09	Salir tras perder en modo plataforma	Debe existir una opción al perder en el modo plataforma que permita cerrar el juego.	1

RF10	Continuar tras perder en modo beat 'em up	Debe existir una opción al perder en el modo beat 'em up que permita continuar el juego.	1
RF11	Continuar tras perder en modo plataforma	Debe existir una opción al perder en el modo plataforma que permita continuar el juego.	1
RF12	Autoguardado de partida	Deben existir puntos donde se guarden las variables automáticamente al pasar por ellos.	1
RF13	Continuar partida desde menú principal	En el menú de inicio debe existir un botón con la etiqueta "cargar partida" que continúe la última partida guardada en el último punto.	1
RF14	Correspondencia de localidad en el mapa	Al pasar de un escenario a otro, el personaje debe corresponder a la dirección que se toma (si el personaje pasa de un escenario de izquierda a derecha, debe aparecer a la izquierda del siguiente mapa y viceversa).	1
RF15	Mostrar subida de nivel del personaje	Al juntar la experiencia suficiente en las batallas, el personaje debe subir de nivel, ésta subida debe ser mostrada al ganar la experiencia.	2
RF16	Asignar punto de salud al subir dos niveles del personaje	Cada $1+2n$ (con $n$ entero y mayor o igual a 1) el personaje debe adquirir un punto mas de salud máxima (max HP).	1
RF17	Mostrar nivel en el menú del personaje	Al abrir el menú del personaje, debe aparecer visualmente el nivel actual del personaje.	2
RF18	Crear altares dinámicos	Deben existir un objeto tipo altar o símil, que pueda ser instanciado dentro del juego cuyas variables sean la habilidad a entrar, el diálogo de la diosa y el punto de teletransporte	1
RF19	Teletransporte entre altares.	Al activar un altar dentro del juego, éste debe mostrar el lugar actual y los destinos de otros altares activados anteriormente.	2
RF20	Habilitar habilidad a personaje desde altar.	Al activar un altar debe activarse la habilidad correspondiente al escenario instanciada como variable del altar. Ésta debe tener una explicación de uso a través de un diálogo.	1
RF21	Implementar habilidad de doble salto	Debe existir una habilidad que permita al personaje saltar estando una vez en el aire.	1
RF22	Implementar habilidad	Debe existir una habilidad que permita desactivar	1

	de activar/desactivar trampas de plantas	ciertas plantas que bloqueen el camino, a la vez, debe poder volver a activarlas.	
RF23	Implementar habilidad de desdoblamiento	Debe existir una habilidad que permita al personaje duplicarse; el sprite del personaje debe quedar inactivo, mientras que el duplicado debe poder moverse libremente por el mapa durante unos momentos.	1
RF24	Recuperación de salud a través de pociones	Debe existir una tecla para utilizar pociones adquiridas, ésta debe restablecer a lo más tres puntos de salud. Si la salud del personaje está al máximo, debe ser imposible utilizar la poción.	2
RF25	Aumento de ataque al equipar armas	Al equipar un arma comprada, el daño que hace el personaje debe aumentar, esto significa que al golpear a un enemigo la resta de salud del enemigo debe ser mayor a la cuando el arma no está equipada.	2
RF26	Insertar cuatro tipos de enemigos dentro del juego	Dentro del juego, debe existir una diversidad de al menos cuatro enemigos distintos, cuya experiencia entregada, velocidad, animaciones y oro que entrega deben ser distintos entre sí.	2
RF27	Agregar nivel que requiera habilidad de doble salto	Se debe diseñar y desarrollar un nivel que sea imposible de terminar sin haber adquirido la habilidad de doble salto.	1
RF28	Agregar nivel que requiera habilidad de activar/desactivar plantas	Se debe diseñar y desarrollar un nivel que sea imposible de terminar sin haber adquirido la habilidad de desactivar/activar trampas de plantas.	1
RF29	Agregar nivel que requiera habilidad de desdoblamiento	Se debe diseñar y desarrollar un nivel que sea imposible de terminar sin haber adquirido la habilidad de desdoblamiento/duplicado.	1
RF30	Fijar un enemigo final en el videojuego	Debe existir un enemigo final que se encuentre en el nivel que utiliza la habilidad de duplicado, éste, al ser derrotado, debe dar un final al videojuego.	2
RF31	Utilizar base de datos para contener armas de los vendedores	Los ítems (pociones y armas) que son vendidos dentro del videojuego deben estar almacenados en una base de datos.	1

Tabla 6.2 Requerimientos No Funcionales

Requerimientos No Funcionales			
ID	Requerimiento	Descripción	Prioridad
RNF01	Término de diálogo	Al finalizar un diálogo, el objeto que controle la conversación debe ser destruído.	1
RNF02	Tiempo de autoguardado	El autoguardado debe durar menos del tiempo en que el ícono correspondiente se muestre en el juego.	2
RNF03	Visibilidad de interfaz de usuario	La interfaz de usuario (salud del personaje, pociones y monedas recolectadas) debe ser visible en todo momento que el usuario pueda controlar al personaje.	2

## 7 Diseño

### 7.1 Componentes de Software

El diseño del videojuego se presentan por los componentes dados por la interfaz de Godot, las escenas y nodos, donde su estructura es programable en el lenguaje de GDScript. En la fase alpha se presentó un diagrama de cómo estos componentes pueden ser conectados para el propósito del proyecto, a continuación se mostrará dicho diagrama:

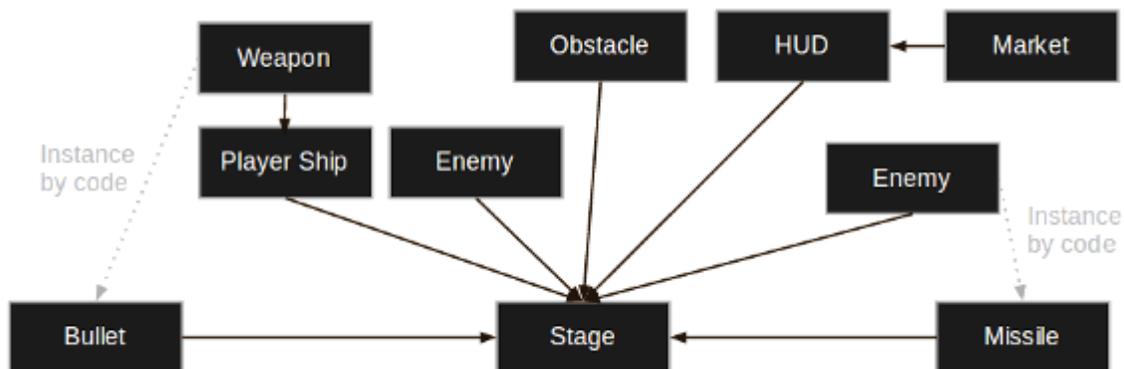


Figura 7.1 Diagrama de escenas Godot

### 7.2 Base de Datos

Se utilizará una base de datos para almacenar información de todos los ítems que el jugador puede equipar, ya sean armas, armaduras o accesorios, además del registro de los vendedores y sus inventarios, los cuales serán cargados dentro del juego en la medida que sean necesarios. Ver anexo D para diseño de la base de datos.

### 7.3 Interfaz

Para la interfaz del videojuego se debe apreciar la fase alpha como base y estructura, pero es muy importante que todo apartado gráfico en la fase anterior puede ser mejorado; por ejemplo, muchos de los cuadros de diálogos con los que el usuario debe interactuar (como en el menú de inicio) no tienen una capa visual adecuada a la ambientación. Aun así, el esquema visual principal se encuentra en la fase inicial.

Lo nuevo en cuanto a lo visual vendrían siendo los diálogos, que, si bien en la fase temprana no eran necesarios, esta vez deberán ser implementados. Para esto una interfaz amigable para los diálogos es importante para la comprensión de la historia. En las nuevas mecánicas, se introduce el crecimiento de niveles de experiencia, esto es la acumulación de experiencia para adquirir una nueva habilidad, por lo que se debe agregar a la interfaz de

usuario para que note los cambios cuando el PJ suba de nivel. A continuación, se presenta la interfaz gráfica en el modo plataforma y beat 'em up, adicionalmente, en el anexo E se podrá divisar la interfaz del actual desarrollo.

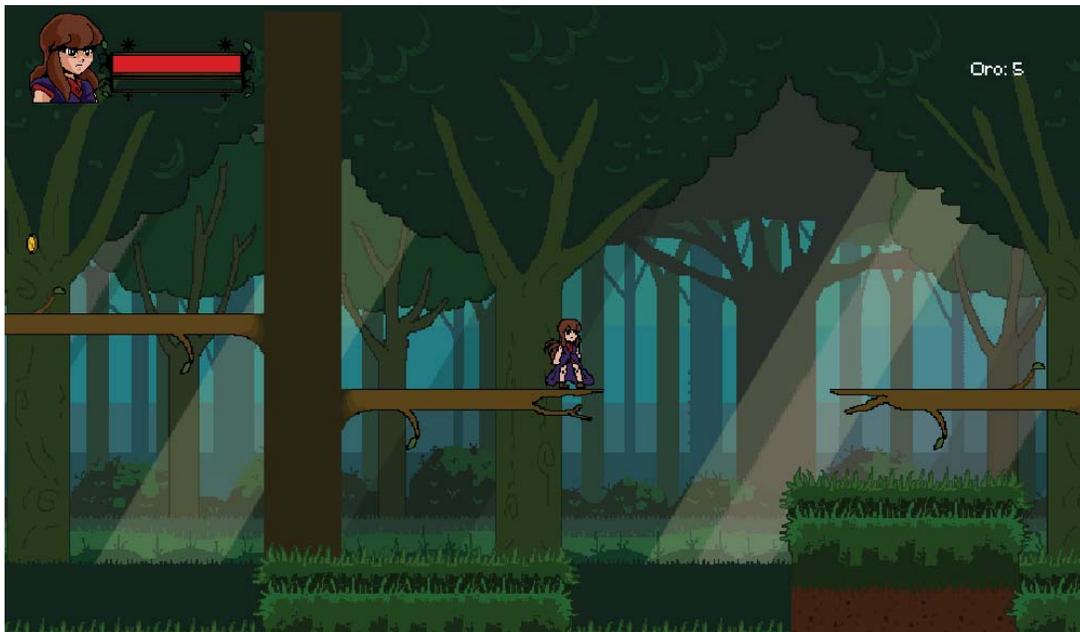


Figura 7.2 Modo plataforma



Figura 7.3 Modo Beat 'em up

## **8 Implementación**

De acuerdo a lo planificado, se ejecutaron dos fases de implementación en el desarrollo del software, donde se realizaron, en primera instancia, pruebas internas de consistencia, y en segunda, pruebas con usuarios externos que cumplen con el perfil de público objetivo.

### **8.1 Implementación Pre Beta**

Se desarrollaron las dos primeras mecánicas del modo plataforma en conjunto con los niveles necesarios para usarlas, estas son el doble salto y activar/desactivar trampas de planta. Además de desarrollar el sistema de diálogos y la consistencia en el cambio de escenarios.

### **8.2 Pruebas Pre Beta**

Las pruebas que se realizaron en la primera fase fueron internas, por lo que no fue necesario un plan de pruebas. Aún así se debieron comparar los objetivos junto a los requerimientos para probar la consistencia del software.

### **8.3 Resultados Pre Beta**

La consistencia para guardar datos aún no estaba completa, al pasar de un escenario a otro, los niveles que había aumentado el personaje no se guardaban en la escena global, impidiendo mantener los avances y aumento de salud. Fuera de estos problemas el programa respondía de acuerdo al avance.

### **8.4 Implementación Beta**

La fase final beta del proyecto debía contar con todos los requerimientos en marcha sobre el software, cumpliendo por otro lado, los objetivos específicos planteados.

Se agregaron las escenas que faltaban, tanto escenarios como los objetos que interactúan con el jugador, ya sea enemigos, ítems o parte de las habilidades. Para esto se desarrollaron objetos cuyo comportamiento se ve afectado por la escena (objeto) Player (personaje jugable). Exceptuando el doble salto, las habilidades (activar/desactivar plantas y desdoblamiento) dependen directamente de su propia escena, teniendo propiedades propias que interactúan con Player.

### **8.5 Pruebas Beta**

Para la fase beta, el software ha de estar completamente funcional, por lo tanto se contó con usuarios posibles para realizar las pruebas, éstos debieron seguir un plan de pruebas que corrobora los requerimientos más importantes del videojuego. Ver anexo F para el plan de pruebas.

## **8.6 Resultados Beta**

Bajo el estudio de quince sujetos sostenidos a las pruebas del plan presentado anteriormente, se obtuvieron los resultados esperados en trece de las quince ocasiones, donde, en el primer caso fallido, la prueba no lograda fue la primera (P1), por no cumplir con las actualizaciones de Open GL en la versión 3.3; en la segunda ocasión falla el segundo caso de prueba, debido a una inconsistencia para la ejecución en equipos con sistema operativos IOS.

## **8.7 Publicación**

Tras verificar los requerimientos, el software fue actualizado, generando un ejecutable (FERONIA\_beta.EXE) para disponer en una plataforma de distribución digital. La escogida fue Itch.io [13], la razón principal es el alto costo de publicación en Steam (US\$100 por publicación de videojuego).

## 9 Conclusiones

Es difícil no comparar la fase alpha del proyecto con la desarrollada en esta instancia, ya que el diferencial entre estas dos fases demuestra los cambios que se lograron, los avances que se implementaron y las proyecciones nuevas que se alcanzaron. Partiendo por el hecho de que se lograron cubrir características incompletas en la fase alpha, tales como los escenarios, el aumento de nivel o el sistema de diálogo, esto permitió analizar de mejor manera el estado actual del proyecto para luego lograr diseñar las mecánicas actuales, ya sea las habilidades, los altares o el sistema de progreso.

Por otro lado, las pruebas de software concluyeron en su mayoría en resultados positivos, pero aun así existieron dos casos donde fallaron, esto es en parte por los requerimientos de uso, pues el videojuego aún no es compatible en su totalidad en equipos con sistemas operativos IOS, se sospecha que el error es producido por el sistema de guardado automático en rutas no válidas para dicho sistema operativo; en el caso de la inconsistencia en Windows, el ejecutable no arrancaba por la falta de actualizaciones en su tarjeta gráfica. Por estas razones es necesario dejar en claro los requerimientos de uso para cualquier tipo de usuario.

Como proyección, el videojuego debe seguir siendo mantenido con actualizaciones, aunque ya se encuentra publicado, es prudente mejorar la consistencia en otros sistemas operativos y cambiar la forma portable por un instalador donde se compruebe si la tarjeta gráfica es compatible y si se encuentra actualizada.

En un futuro se piensa agregar modos de juego multiplayer, donde los escenarios principales serían el modo batalla jugador versus jugador y carreras en el modo plataforma. Además, si bien el juego ha sido publicado, teniendo estas últimas actualizaciones se pretende publicar el videojuego en la plataforma Steam pagando el debido precio por este proceso.

## 10 Referencias

Las referencias son, en su totalidad, digitales, por lo que los contenidos en los vínculos pueden variar o dejar de existir.

- [1] Fullerton, T. (2008). Game Design Workshop: A playcentric approach to creating innovative games 2<sup>nd</sup> Edition. Recuperado de: <http://host.conseiljedi.com/~kira/Game%20Design%20Workshop-A%20playcentric%20approach%20to%20creating%20innovative%20games-2nd%20Edition.pdf>. Revisada por última vez el 7 de diciembre de 2018.
- [2] Godot Engine 3.0. (s.f). Recuperado de: <https://godotengine.org/article/godot-3-0-released>. Revisada por última vez el 7 de diciembre de 2018.
- [3] Galyonk, S. (2017). The indie games are too damn cheap. Recuperado de: <https://galyonk.in/the-indie-games-are-too-damn-cheap-11b8652fad16>. Revisada por última vez el 7 de diciembre de 2018.
- [4] Tran, J. (2013). The evolution of the niche beat'em up genre. Recuperado de: <https://venturebeat.com/community/2013/02/09/in-retrospect-the-beat-em-up-genre/>. Revisada por última vez el 7 de diciembre de 2018.
- [5] Godot Engine Features. (s.f). Recuperado de: <https://godotengine.org/features>. Revisada por última vez el 7 de diciembre de 2018.
- [6] GDScript: An introduction to dynamic languages. (s.f). Recuperado de: [http://docs.godotengine.org/en/3.0/getting\\_started/scripting/gdscript/gdscript\\_advanced.html](http://docs.godotengine.org/en/3.0/getting_started/scripting/gdscript/gdscript_advanced.html). Revisada por última vez el 7 de diciembre de 2018.
- [7] Plataforma de distribución digital ITCH.IO (s.f). Recuperado de: <https://itch.io>. Revisada por última vez el 7 de diciembre de 2018.
- [8] Staff Tarreo. (2018). Steam alcanza nuevo récord con 18.5 millones de usuarios concurrentes Recuperado de: <http://www.tarreo.com/noticias/454574/Steam-alcanza-nuevo-record-con-18-5-millones-de-usuarios-concurrentes>. Revisada por última vez el 7 de diciembre de 2018.
- [9] Steam Direct Product Submission Fee. (s.f). Recuperado de: <https://store.steampowered.com/sub/163632/>. Revisada por última vez el 7 de diciembre de 2018.
- [10] Steam Direct. (s.f). Recuperado de: <https://partner.steamgames.com/steamdirect>. Revisada por última vez el 7 de diciembre de 2018.
- [11] Nicolás A. et al (2009) Una Metodología para Desarrollo de Videojuegos. Recuperado de: [https://www.fing.edu.uy/sites/default/files/biblio/22811/asse\\_2009\\_16.pdf](https://www.fing.edu.uy/sites/default/files/biblio/22811/asse_2009_16.pdf). Revisada por última vez el 7 de diciembre de 2018.
- [12] Godot Engine Features. (s.f). Recuperado de: <https://godotengine.org/features>. Revisada por última vez el 7 de diciembre de 2018.
- [13] Feronia Beta (2018). Recuperado de: <https://feronygames.itch.io/feronia-prebeta>. Revisada por última vez el 7 de diciembre de 2018.

## 11 Bibliografía Consultada

1. Prieto, R., Medina, N., Paderewski, P., Gutierrez, F.L., (s.f). Design methodology for educational games based on interactive screenplays. Recuperado de: [http://ceur-ws.org/Vol-1394/paper\\_9.pdf](http://ceur-ws.org/Vol-1394/paper_9.pdf)
2. Fernández, J. (2015). La industria del videojuego factura más de 1.000 millones de euros en España. Recuperado de: <http://www.expansion.com/economia-digital/companias/2016/06/02/57501ca5ca4741453f8b4621.html>
3. Dorta, A. (2017). Videojuegos, la industria digital que más ingresos genera. Recuperado de: <http://www.mexiconewsnetwork.com/es/noticias/videojuegos-la-industria-digital-que-mas-ingresos-genera/>
4. Parrilla, J. (2017). El sector de los videojuegos, el de mayor facturación de la industria digital. Recuperado de: <https://www.hobbyconsolas.com/noticias/sector-videojuegos-mayor-facturacion-industria-digital-100378>
5. Pérez, D. (2015). Reglamento que regula el contenido, forma, dimensiones y demás características de la leyenda que deben exhibir los videojuegos. Recuperado de: <https://www.leychile.cl/Navegar?idNorma=1099919>
6. Bello, E., Ward, C. A., Dartnell, P., Maza, J. (1925). DL-345. Recuperado de: <https://www.leychile.cl/Navegar?idNorma=1016627>
7. Pro-Chile. (2012). Estudio de mercado videojuegos en Estados Unidos. Recuperado de: [https://www.prochile.gob.cl/wp-content/files\\_mf/documento\\_12\\_11\\_12175027.pdf](https://www.prochile.gob.cl/wp-content/files_mf/documento_12_11_12175027.pdf)

## **Anexos**

## A: Diagramas utilizados en la etapa Alpha

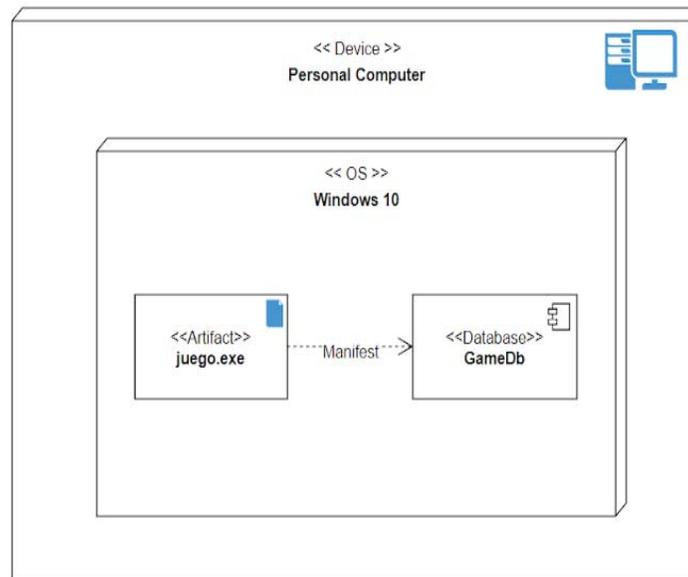


Diagrama de despliegue

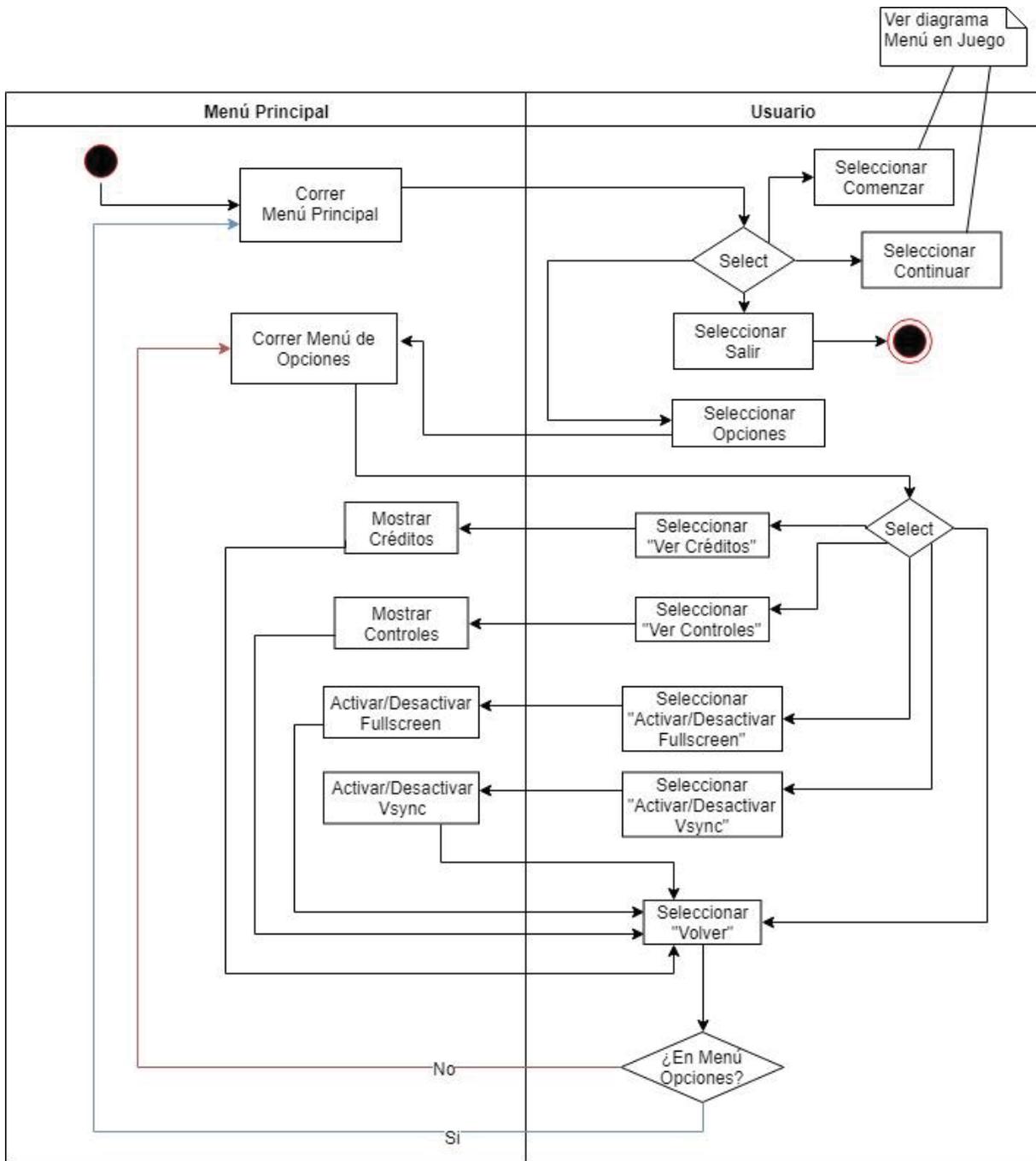


Diagrama de Actividad de Menú Opciones

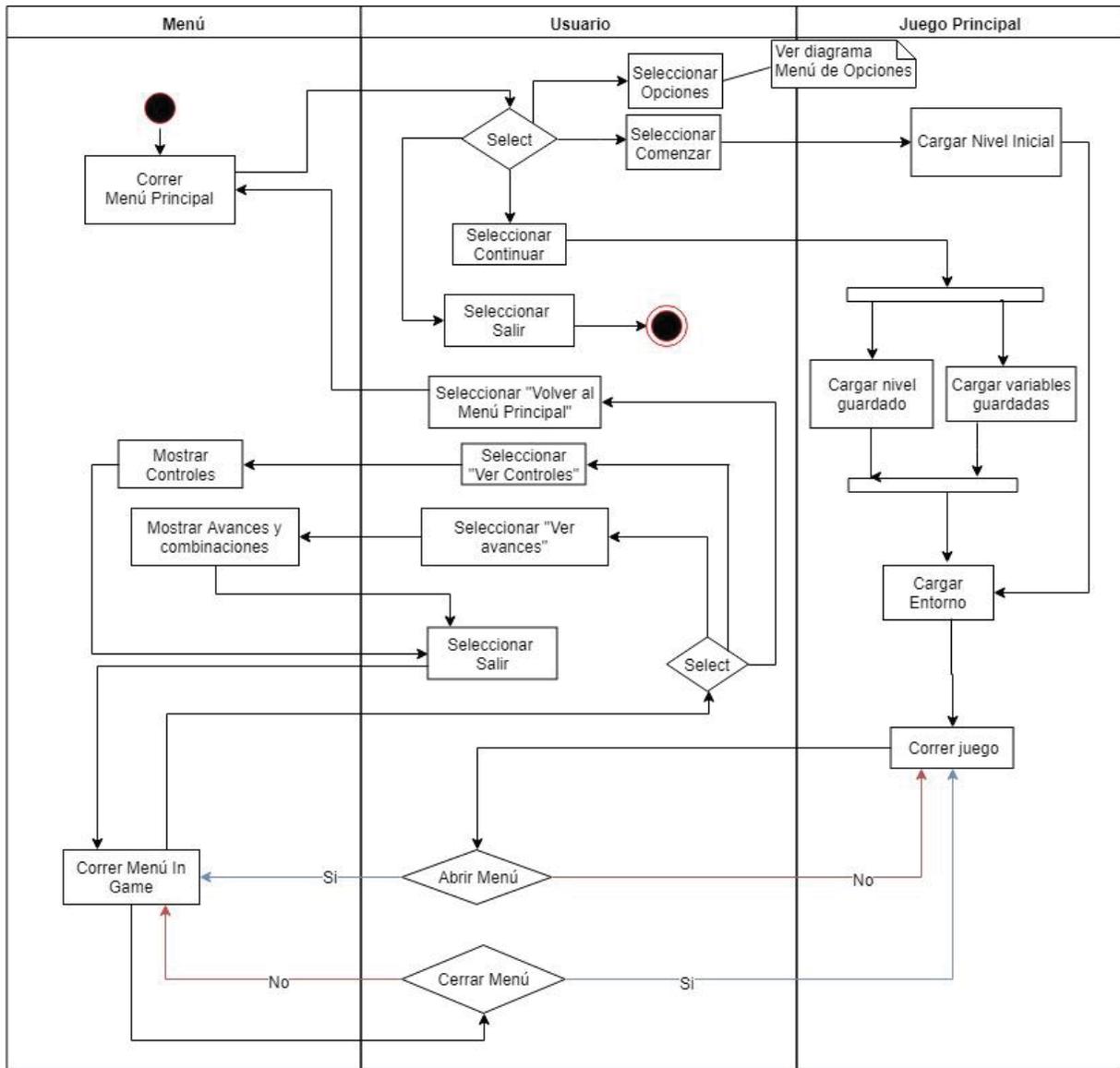


Diagrama de Actividad de Menú en Juego

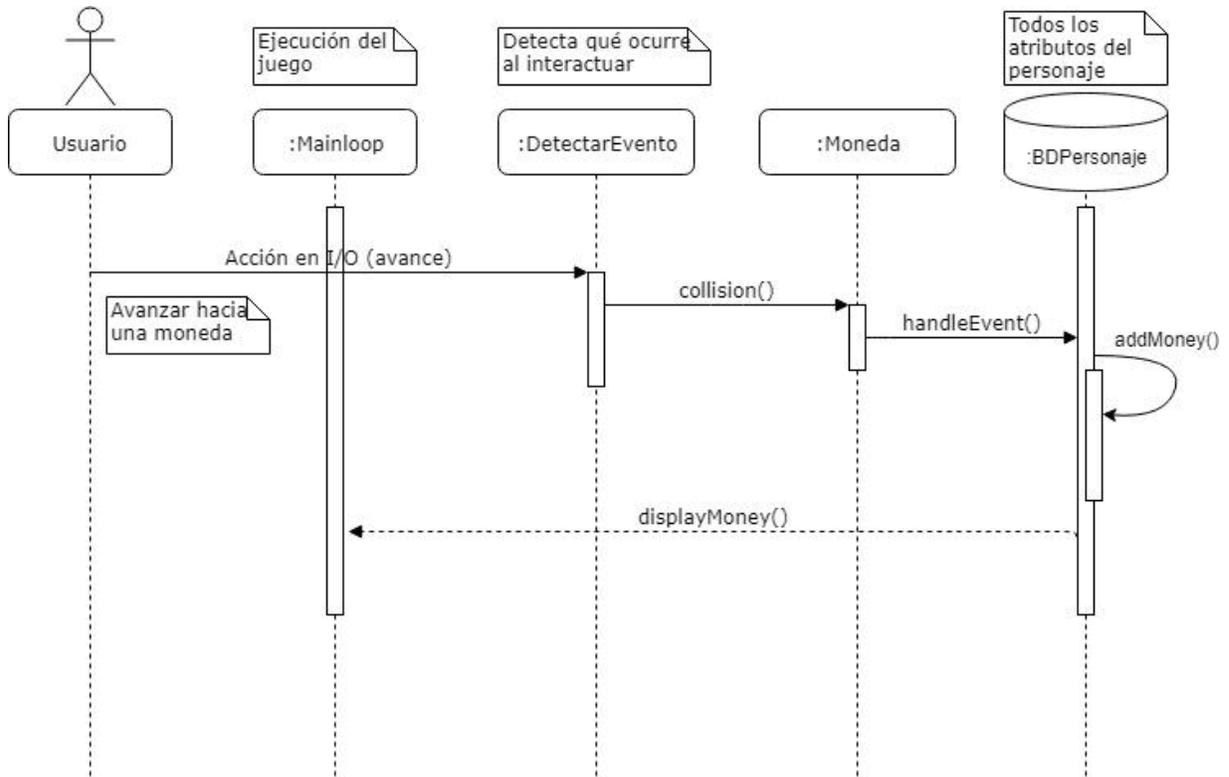


Diagrama de Secuencia de Recoger monedas

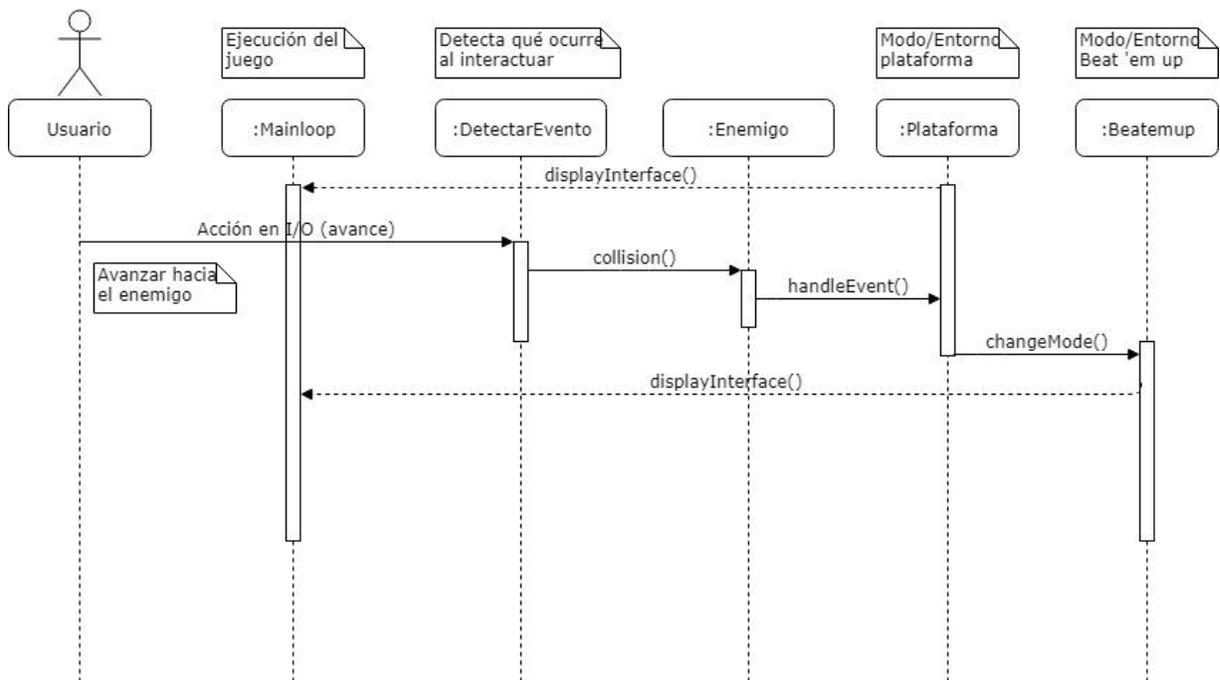


Diagrama de Secuencia de entrar a modo beat 'em up

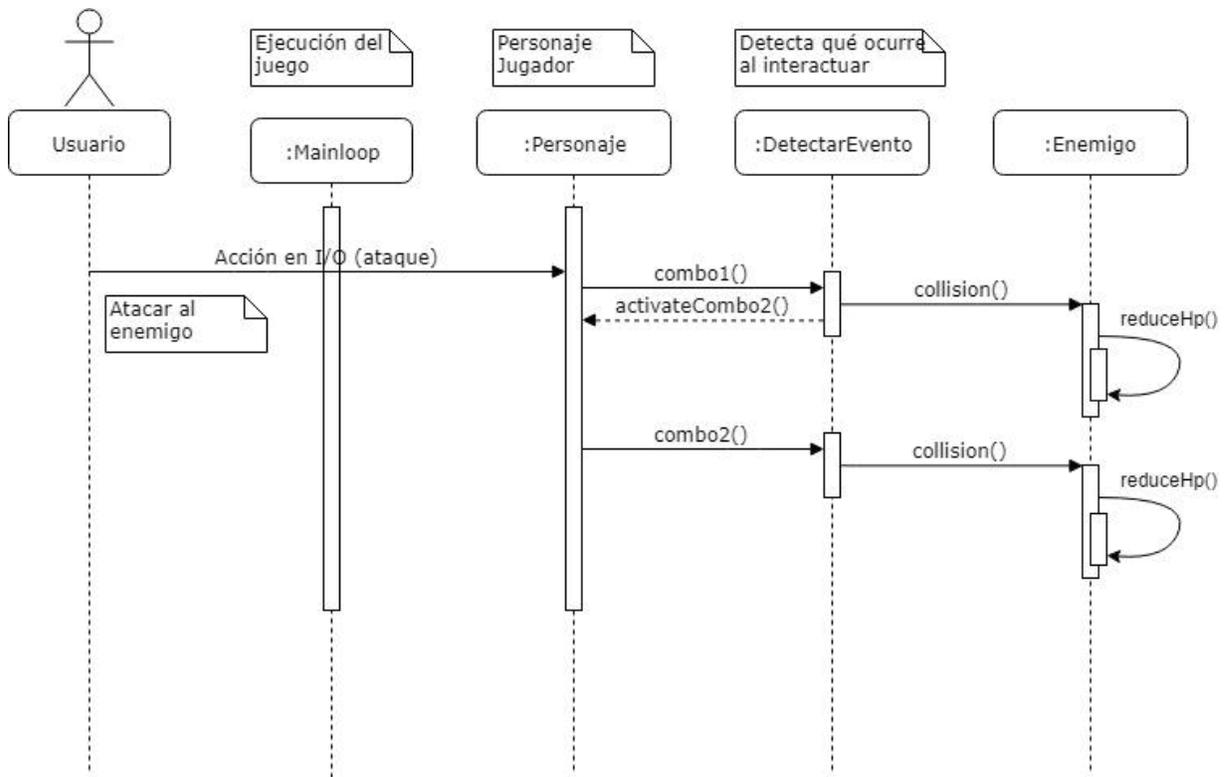


Diagrama de Secuencia de Atacar al enemigo

## **B: Estimación de cantidad de HH en etapa Alpha previo al desarrollo**

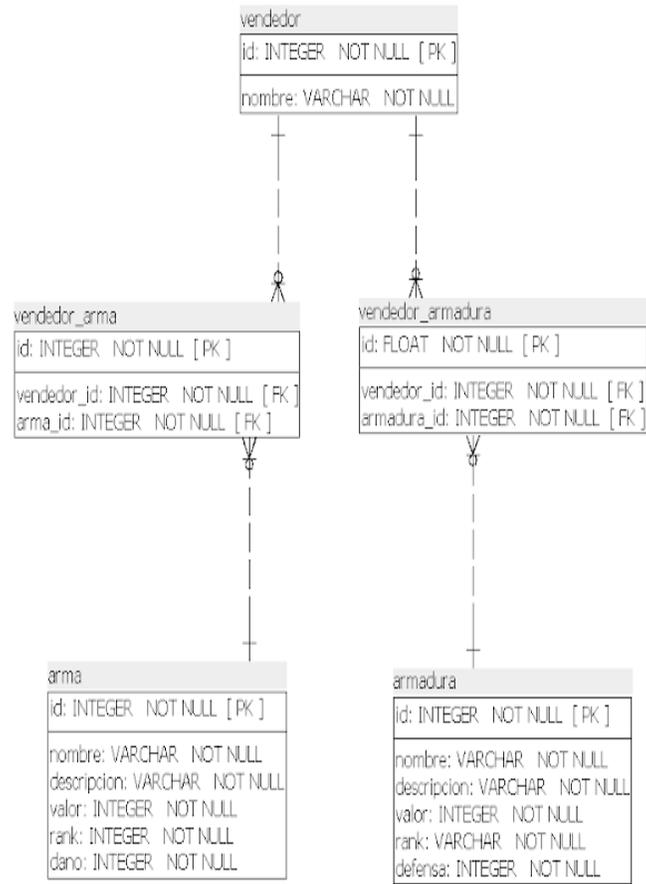
	Diseño	Programación	Guión	Tester	Documentación	TOTAL
Programador	60 HH	180 HH	30 HH	10 HH	30 HH	310 HH
Diseñador	45 HH	45 HH	45 HH	10 HH	60 HH	205 HH
Analista de Sistema	45 HH	60 HH	30 HH	10 HH	45 HH	190 HH
Planificador y mantenedor	60 HH	45 HH	60 HH	10 HH	75 HH	250 HH
Game Tester	-	-	-	20 HH	-	20 HH
TOTAL	210 HH	330 HH	165 HH	60 HH	210 HH	975 HH

## C: Carta Gantt

Nombre	Fecha de inicio	Fecha de fin
☐ • Revisión del estado del Tema	23/08/18	27/08/18
• Revisión del estado actual del software	23/08/18	24/08/18
• Revisión de estado actual del motor	24/08/18	25/08/18
• Investigación de mecánicas en juegos similares	25/08/18	27/08/18
• Revisión e Investigación de mercado	27/08/18	27/08/18
☐ • Revisión de estado actual del motor	27/08/18	30/08/18
• Definición de responsabilidades	27/08/18	27/08/18
• Definición de metodología	27/08/18	28/08/18
• Definición de modelo de proceso	28/08/18	29/08/18
• Documentación	30/08/18	30/08/18
☐ • Propuesta de Solución	30/08/18	4/09/18
• Análisis de riesgo	30/08/18	31/08/18
• Análisis de factibilidad de desarrollo	31/08/18	1/09/18
• Definición de concepto para continuidad	1/09/18	4/09/18
• Definición de herramientas necesarias	3/09/18	3/09/18
• Definición de motor de base de datos	4/09/18	4/09/18
☐ • Etapa de Desarrollo 1	5/09/18	3/10/18
☐ • Diseño	5/09/18	12/09/18
• Definición de nuevos requerimientos	5/09/18	6/09/18
• Diseño de guión textualizado en un documento	6/09/18	8/09/18
• Diseño de base de datos	6/09/18	8/09/18
• Diseño de niveles	7/09/18	10/09/18
• Diseño de nuevas mecánicas de ataque	10/09/18	12/09/18
• Reunión de feedback	12/09/18	12/09/18
☐ • Desarrollo	13/09/18	25/09/18
• Desarrollo de base de datos	13/09/18	15/09/18
• Desarrollo de nuevas mecánicas en plataforma	14/09/18	18/09/18
• Ilustración de Sprites	15/09/18	18/09/18
• Desarrollo de nuevas mecánicas en beat 'em up	15/09/18	19/09/18
• Ilustración de escenarios	19/09/18	21/09/18
• Redacción de guión	20/09/18	22/09/18
• Documentación del desarrollo	22/09/18	25/09/18
• Reunión de feedback	25/09/18	25/09/18
☐ • Implementación	26/09/18	29/09/18
• Implementación de base de datos	26/09/18	28/09/18
• Documentación de avances	26/09/18	29/09/18

● Implementación de mecánicas en los niveles	27/09/18	29/09/18
● Implementación del guión	27/09/18	29/09/18
☐ ● Revisión de adelanto	28/09/18	29/09/18
● Entrega Informe	28/09/18	28/09/18
● Presentación de Avance	1/10/18	1/10/18
☐ ● Pruebas	1/10/18	3/10/18
● Pruebas de avance	1/10/18	2/10/18
● Documentación de resultados	1/10/18	3/10/18
☐ ● Etapa de Desarrollo 2	4/10/18	28/11/18
● Análisis de nueva etapa	4/10/18	5/10/18
☐ ● Diseño	5/10/18	16/10/18
● Definición de nuevos requerimientos	5/10/18	8/10/18
● Corregir árbol del guión de diálogos	5/10/18	8/10/18
● Correcciones en la base de datos	6/10/18	10/10/18
● Diseño de niveles	8/10/18	12/10/18
● Diseño de nuevas mecánicas de ataque	11/10/18	15/10/18
● Correcciones en nuevas mecánicas de plataforma	13/10/18	16/10/18
☐ ● Desarrollo	17/10/18	5/11/18
● Agregación de correcciones en la base de datos	17/10/18	22/10/18
● Agregación de mecánicas modificadas(plataforma)	19/10/18	24/10/18
● Agregación de mecánicas beat 'em up	22/10/18	27/10/18
● Ilustración de sprites faltantes	23/10/18	29/10/18
● Ilustración de detalles	30/10/18	1/11/18
● Desarrollo corregido del guión	30/10/18	3/11/18
● Documentación del desarrollo	31/10/18	5/11/18
☐ ● Implementación	6/11/18	16/11/18
● Implementación de correcciones en la BD	6/11/18	10/11/18
● Implementación de mecánicas en los niveles	9/11/18	13/11/18
● Implementación del guión	14/11/18	16/11/18
☐ ● Pruebas	17/11/18	28/11/18
● Pruebas de avance	17/11/18	19/11/18
● Pruebas con público posible	20/11/18	22/11/18
● Documentación de resultados	21/11/18	28/11/18
☐ ● Cierre	30/11/18	6/12/18
● Entrega Software Terminado	30/11/18	30/11/18
● Revisión de Software	3/12/18	3/12/18
● Entrega Informe Final	7/12/18	7/12/18

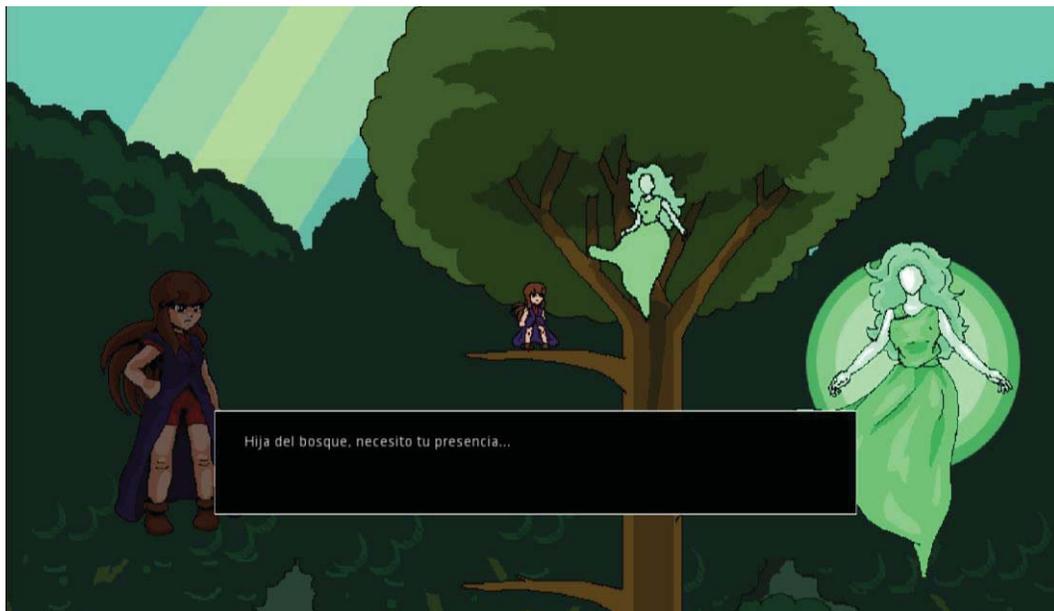
## D: Modelo de la base de datos



## E: Interfaz Gráfica



Menú principal del videojuego.



Escena inicial, diálogo introductorio de Feronia.



Enfrentamiento en escenario bajo tierra.



Menú de Teletransporte.



Compra de pociones.



Doble salto en lugar imposible de llegar con salto simple.



Utilizando habilidad de activar/desactivar plantas.



Utilizando habilidad de desdoblamiento/duplicado.

## F: Plan de pruebas

Identificación de la prueba	
Id de prueba	P1
Nombre de prueba	Inicio del juego
Descripción de la prueba	Inicio del videojuego, donde se debe mostrar la guía o botones de uso y una breve introducción.
Identificador de requerimiento(s)	RF01
Dependencia de prueba	Sin dependencia
Ejecutor	Usuario
Pre condiciones	Debe tener los archivos necesarios para ejecutar el videojuego.
Actividades	
<ul style="list-style-type: none"> <li>• Ejecutar el archivo .exe del videojuego (FERONIA_beta.exe)</li> <li>• Debe seleccionar “Iniciar nueva partida”.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Automáticamente, al iniciar el videojuego, la pantalla debe mostrar los botones básicos de uso.</li> <li>• Se debe mostrar el menú principal del videojuego.</li> <li>• Al iniciar la partida, debe mostrarse una breve introducción antes de comenzar a manipular al personaje.</li> </ul>	

Identificación de la prueba	
Id de prueba	P2
Nombre de prueba	Sistema de diálogos
Descripción de la prueba	Esta prueba se orienta al seguimiento adecuado de los diálogos que ocasionalmente aparecen dentro del videojuego.
Identificador de requerimiento(s)	RF02, RF03

Dependencia de prueba	P1
Ejecutor	Usuario
Pre condiciones	Iniciar una nueva partida.
Actividades	
<ul style="list-style-type: none"> <li>• Utilizar la tecla “D” para avanzar en los diálogos que aparecen en la introducción.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Mostrar los dos personajes en escena que dialogan.</li> <li>• El personaje que habla debe resaltarse con colores y nitidez correctos para entender que es quien habla.</li> <li>• El personaje que no habla debe verse con brillo bajo para entender que es el personaje que no habla.</li> <li>• Al finalizar el último diálogo el juego debe continuar.</li> </ul>	

Identificación de la prueba	
Id de prueba	P3
Nombre de prueba	Cambio de modo
Descripción de la prueba	Prueba que demuestra que al colisionar con un enemigo, el modo de juego cambia a modo pelea (beat ‘em up).
Identificador de requerimiento(s)	Esta prueba comprueba la integridad del sistema luego de las modificaciones en el estado alpha.
Dependencia de prueba	P2
Ejecutor	Usuario
Pre condiciones	Sin precondition
Actividades	
<ul style="list-style-type: none"> <li>• Encontrar un enemigo avanzando en el modo plataforma.</li> <li>• Colisionar con dicho enemigo.</li> </ul>	
Resultados esperados	

- Al colisionar con el enemigo debe mostrarse un leve fade out que cambie la escena.
- Se debe mostrar un mensaje que indique el inicio de pelea junto con los enemigos.

Identificación de la prueba	
Id de prueba	P4
Nombre de prueba	Combinaciones de ataque
Descripción de la prueba	La correcta ejecución de ataque en el modo pelea debe posibilitar la victoria en dicho modo.
Identificador de requerimiento(s)	RF04, RF05
Dependencia de prueba	P3
Ejecutor	Usuario
Pre condiciones	Haber pasado al modo beat 'em up.
Actividades	
<ul style="list-style-type: none"> <li>• Usar la tecla D una vez cerca de un enemigo.</li> <li>• Presionar la tecla S una vez cerca de un enemigo.</li> <li>• Utilizar la tecla Shift.</li> <li>• Usar tres veces consecutivas la tecla D cerca de un enemigo.</li> <li>• Presionar tres veces consecutivas la tecla S cerca de un enemigo.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Ejecutar animación de patada primaria y si colisiona con enemigo restar vida al enemigo.</li> <li>• Ejecutar animación de golpe de puño primario y si colisiona con enemigo restar vida al enemigo</li> <li>• Ejecutar animación de esquivar.</li> <li>• Ejecutar combinación de patadas finalizando con un golpe mágico (animación que lanza un sprite blanco)</li> <li>• Ejecutar combinación de golpes de puño finalizando con un golpe de una planta que sale desde el piso.</li> </ul>	

Identificación de la prueba

Id de prueba	P5
Nombre de prueba	Victoria en batalla
Descripción de la prueba	Muestra la interfaz de victoria con los respectivos mensajes luego de derrotar a los enemigos en modo beat 'em up.
Identificador de requerimiento(s)	RF06, RF07
Dependencia de prueba	P3
Ejecutor	Usuario
Pre condiciones	Estar en modo pelea (beat 'em up)
Actividades	
<ul style="list-style-type: none"> <li>• Derrotar a los enemigos en el modo beat 'em up</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Se muestran las monedas adquiridas por ganar.</li> <li>• Se muestra la experiencia total obtenida.</li> <li>• Cambia al modo plataforma</li> </ul>	

Identificación de la prueba	
Id de prueba	P6
Nombre de prueba	Derrota en batalla y salir del juego
Descripción de la prueba	Esta prueba demuestra que puedes ser derrotado en el modo beat 'em up y luego cerrar el juego para no continuar jugando.
Identificador de requerimiento(s)	RF08
Dependencia de prueba	P3
Ejecutor	Usuario
Pre condiciones	Debe estar en el modo beat 'em up

Actividades
<ul style="list-style-type: none"> <li>• Comenzar una batalla en el modo beat ‘em up.</li> <li>• Dejar que los enemigos hagan daño al personaje hasta vaciar su barra de vida (HP).</li> <li>• Una vez derrotado, seleccionar “salir”.</li> </ul>
Resultados esperados
<ul style="list-style-type: none"> <li>• Se ejecuta la animación de muerte cuando la barra de HP se vacía.</li> <li>• Aparece la pantalla de “Game Over”.</li> <li>• El juego debe cerrarse.</li> </ul>

Identificación de la prueba	
Id de prueba	P7
Nombre de prueba	Cargar partida
Descripción de la prueba	Comprobar que es posible cargar una partida
Identificador de requerimiento(s)	RF13
Dependencia de prueba	P1
Ejecutor	Debe, al menos, avanzar hasta donde los controles del jugador se active.
Pre condiciones	Cerrar el juego luego de haber jugado al menos el nivel que aparece a continuación de la introducción.
Actividades	
<ul style="list-style-type: none"> <li>• Ejecutar nuevamente el juego FERONIA_beta.exe.</li> <li>• Seleccionar la opción “continuar”.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Aparecerá en el último punto de guardado(checkpoint) que alcanzó en el juego, si sólo jugó hasta luego del intro debe aparecer en la copa del árbol inicial.</li> </ul>	

Identificación de la prueba
-----------------------------

Id de prueba	P8
Nombre de prueba	Cambio de escenas entre mapas
Descripción de la prueba	Al pasar de un mapa a otro, el punto de aparición debe ser acorde al de donde precedía.
Identificador de requerimiento(s)	RF14
Dependencia de prueba	P1
Ejecutor	Usuario
Pre condiciones	
Actividades	
<ul style="list-style-type: none"> <li>• En el modo plataforma del juego, desde el escenario inicial, caminar hacia la derecha hasta cambiar al siguiente escenario.</li> <li>• Una vez en el segundo escenario, devolverse.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Al cambiar al segundo escenario (desde el primero) el personaje debe aparecer a la izquierda del mapa.</li> <li>• Al retroceder al mapa anterior, el personaje debe aparecer en el extremo derecho del escenario.</li> </ul>	

Identificación de la prueba	
Id de prueba	P9
Nombre de prueba	Subir de nivel
Descripción de la prueba	Se pretende comprobar que al superar cierta cantidad de batallas el juego debe indicar que el personaje subió de nivel. Cada dos niveles que adquiriera, se sumará un punto de salud máxima.
Identificador de requerimiento(s)	RF15, RF16, RF17
Dependencia de prueba	P1

Ejecutor	Usuario
Pre condiciones	Estar en el segundo escenario (primer escenario a la derecha del inicial).
Actividades	
<ul style="list-style-type: none"> <li>• Colisionar con todos los enemigos del segundo escenario (escenario a la derecha del inicial en el modo plataforma).</li> <li>• Derrotar a cada uno de los enemigos con los que se ha colisionado.</li> <li>• Presionar tecla I.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Al finalizar una cierta cantidad de victorias se debe mostrar un mensaje con la frase “Subida de nivel”.</li> <li>• Al derrotar los enemigos restantes debe volver a aparecer el mensaje de “Subida de nivel”.</li> <li>• Al presionar la tecla I se debe mostrar un menú, bajo la imagen del personaje debe aparecer dos niveles más de los que se encontraba en el inicio de la prueba (nivel 3).</li> <li>• Al subir dos niveles la salud máxima aumenta en una unidad (comprobable mediante la batalla con otro enemigo)</li> </ul>	

Identificación de la prueba	
Id de prueba	P10
Nombre de prueba	Adquisición de habilidades
Descripción de la prueba	Esta prueba permite corroborar la adquisición de las habilidades que tienen funciones especiales en el modo plataforma.
Identificador de requerimiento(s)	RF12, RF18, RF20
Dependencia de prueba	P9
Ejecutor	Usuario
Pre condiciones	Superar el primer nivel, llegando hasta el final del primer escenario (a la derecha del inicial).
Actividades	

<ul style="list-style-type: none"> <li>• Presionar tecla E frente o sobre al sprite del altar (sprite gris).</li> <li>• Terminar el diálogo.</li> </ul>
Resultados esperados
<ul style="list-style-type: none"> <li>• Iniciar dialecto de la diosa donde enseña cómo utilizar la habilidad adquirida.</li> <li>• Ejecutar autoguardado (checkpoint) anunciado por el ícono de un disco azul en la esquina inferior derecha.</li> </ul>

Identificación de la prueba	
Id de prueba	P11
Nombre de prueba	Muerte en plataforma y continuar partida.
Descripción de la prueba	Comprobación de que el personaje puede perder cayendo al vacío en el modo plataforma y luego puede continuar la partida desde el último punto de guardado.
Identificador de requerimiento(s)	RF11
Dependencia de prueba	P10
Ejecutor	Usuario
Pre condiciones	Permanecer en el modo plataforma
Actividades	
<ul style="list-style-type: none"> <li>• En el modo plataforma avanzar hacia cualquier precipicio y dejarse caer.</li> <li>• Seleccionar “continuar”.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• El juego debe mostrar un “fade out”, seguido de un mensaje de “Game over”.</li> <li>• Al seleccionar continuar, el personaje debe aparecer en el último punto de control (checkpoint) donde la partida se guardó.</li> </ul>	

Identificación de la prueba
-----------------------------

Id de prueba	P12
Nombre de prueba	Salir del videojuego
Descripción de la prueba	Comprobar que es posible salir del videojuego a través del menú dentro del juego (in-game)
Identificador de requerimiento(s)	Esta prueba comprueba la integridad del sistema luego de las modificaciones en el estado alpha.
Dependencia de prueba	P1
Ejecutor	Usuario
Pre condiciones	Mantener el videojuego en ejecución.
Actividades	
<ul style="list-style-type: none"> <li>• Presionar la tecla ESC.</li> <li>• Seleccionar la opción “salir” con la tecla D</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Se debe desprender un menú con los que indique la opción de ver los controles o salir.</li> <li>• Al seleccionar salir el juego debe cerrarse.</li> </ul>	

Identificación de la prueba	
Id de prueba	P13
Nombre de prueba	Habilidad de doble salto
Descripción de la prueba	Corroborar que es posible utilizar la habilidad de doble salto una vez adquirida.
Identificador de requerimiento(s)	RF21
Dependencia de prueba	P10
Ejecutor	Usuario
Pre condiciones	Llegar hasta el final del primer escenario a la derecha del escenario final.

	Activar el altar.
Actividades	
<ul style="list-style-type: none"> <li>• Presionar una vez la barra espaciadora</li> <li>• Una vez el personaje se encuentre en el aire volver a presionar la barra espaciadora.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• El personaje debe saltar en el lugar.</li> <li>• El personaje debe volver a saltar estando en el aire realizando otra animación.</li> </ul>	

Identificación de la prueba	
Id de prueba	P14
Nombre de prueba	Habilidad de activar/desactivar trampas
Descripción de la prueba	Una vez adquirida esta habilidad, el jugador debe lograr desactivar o activar plantas que bloquean el camino
Identificador de requerimiento(s)	RF22
Dependencia de prueba	P13
Ejecutor	Usuario
Pre condiciones	Conseguir la habilidad de doble salto.
Actividades	
<ul style="list-style-type: none"> <li>• Una vez conseguido la habilidad de doble salto, el jugador deberá ir a la izquierda, atravesando el escenario inicial, saltando hacia la izquierda.</li> <li>• Llegar hasta el final del escenario de la izquierda y bajar por el precipicio.</li> <li>• Avanzar por el nivel subterráneo hacia la derecha evitando caer al precipicio.</li> <li>• Activar el altar.</li> <li>• Acercarse a la planta que bloquea el camino a la derecha.</li> <li>• Utilizar tecla E cerca de la planta.</li> <li>• Presionar nuevamente la tecla E.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Una vez en el altar, la diosa debiera explicar cómo funciona la nueva habilidad.</li> </ul>	

- Junto a la planta, al presionar la tecla E, la planta deberá desaparecer (bajar para permitir al jugador avanzar).
- Al presionar nuevamente, la planta debería aparecer nuevamente boqueando el camino.

Identificación de la prueba	
Id de prueba	P15
Nombre de prueba	Habilidad de Desdoblamiento/Duplicado
Descripción de la prueba	Esta prueba se orienta al correcto funcionamiento de la habilidad de duplicarse como un ente fantasmal dentro del juego.
Identificador de requerimiento(s)	RF23
Dependencia de prueba	P14
Ejecutor	Usuario
Pre condiciones	El jugador debe haber avanzado hasta el escenario dentro del árbol.
Actividades	
<ul style="list-style-type: none"> <li>• Se debe avanzar hasta el final del nivel dentro del árbol (hacia la izquierda)</li> <li>• No se debe caer al precipicio y debe sobrevivir a las batallas.</li> <li>• Al llegar al final, el jugador debe presionar la tecla E junto al altar.</li> <li>• Una vez terminado el diálogo con la diosa, presionar la tecla R.</li> <li>• Moverse libremente con las flechas direccionales.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Aparece diálogo de la diosa quien explica cómo utilizar la nueva habilidad.</li> <li>• Aparece el sprite del ente astral del jugador (fantasma).</li> <li>• Al moverse puede atravesar paredes y pisos.</li> <li>• Luego de un breve momento el ente astral desaparece.</li> </ul>	

Identificación de la prueba	
Id de prueba	P16

Nombre de prueba	Compra y uso de pociones
Descripción de la prueba	Prueba que consiste en comprobar que puede comprar pociones en la tienda y usarlas cuando sea necesario.
Identificador de requerimiento(s)	RF24
Dependencia de prueba	Tener al menos 50 de oro adquirido en el videojuego.
Ejecutor	Usuario
Pre condiciones	Estar al final del segundo escenario (primero a la derecha del inicial), además de tener al menos un punto de vida menos del máximo.
Actividades	
<ul style="list-style-type: none"> <li>• Presionar la tecla E frente al vendedor de pociones.</li> <li>• Seleccionar “Poción de Vida”</li> <li>• Presionar la tecla S.</li> <li>• Presionar la tecla 1.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Se despliega el menú del vendedor.</li> <li>• Si el jugador tiene al menos 50 de oro, se agrega una poción al jugador, si no, en el menú del vendedor aparecerá “No tienes dinero suficiente”.</li> <li>• Consume la poción y recupera un máximo de 3 puntos de vida.</li> </ul>	

Identificación de la prueba	
Id de prueba	P17
Nombre de prueba	Compra y uso de arma
Descripción de la prueba	Prueba que consiste en comprobar que puede comprar pociones en la tienda y usarlas cuando sea necesario.
Identificador de requerimiento(s)	RF25
Dependencia de prueba	Tener al menos 5 de oro para obtenidas en el videojuego.

Ejecutor	Usuario
Pre condiciones	Estar al final del tercer escenario (primero a la izquierda del inicial), además de tener la habilidad de activar y desactivar trampas de plantas
Actividades	
<ul style="list-style-type: none"> <li>• Desactivar planta que bloquea el camino con la tecla E.</li> <li>• Frente al vendedor presionar la tecla E.</li> <li>• Seleccionar “Puños de Feronia”.</li> <li>• Confirmar la compra.</li> <li>• Presionar S.</li> <li>• Presionar I, seguido de la tecla D.</li> <li>• Presionar D y finalmente de la tecla S.</li> <li>• Entrar a una batalla contra cualquier enemigo.</li> <li>• Dar un golpe que colisione con el enemigo.</li> </ul>	
Resultados esperados	
<ul style="list-style-type: none"> <li>• Se debe abrir el menú del vendedor mostrando los ítems en venta.</li> <li>• Debe aparecer un mensaje de confirmación de compra.</li> <li>• Se debe desplegar el inventario de Feronia.</li> <li>• Se debe entrar al menú de armas compradas.</li> <li>• Debe quedar equipada el arma comprada indicado por un ícono amarillo con la letra E.</li> <li>• Al entrar en una batalla, el daño que haga el personaje debe ser de cinco puntos.</li> </ul>	

Identificación de la prueba	
Id de prueba	P18
Nombre de prueba	Escena final
Descripción de la prueba	Se prueba la existencia de un enemigo final, de la cual al obtener la victoria el juego acaba.
Identificador de requerimiento(s)	RF30
Dependencia de prueba	P13, P14, P15
Ejecutor	Usuario
Pre condiciones	Avanzar hasta el último escenario a la derecha.

Actividades
<ul style="list-style-type: none"><li>• Acercarse al último enemigo del último escenario posible de llegar hacia la derecha.</li><li>• Derrotar al enemigo usando las combinaciones de teclas anteriormente aprendidas.</li></ul>
Resultados esperados
<ul style="list-style-type: none"><li>• Se debe mostrar un diálogo entre el enemigo final y el personaje antes de colisionar.</li><li>• Al derrotar al enemigo debe aparecer un nuevo diálogo que dé cierre al juego.</li></ul>