

Universidad ORT Uruguay

Facultad de Ingeniería Bernard Wand-Polak

Crystal Clash - Videojuego 2D para dispositivos móviles

Entregado como requisito para la obtención del título Ingeniero en Sistemas.

Federico Brunstein – 143244

Gonzalo Cumini – 159085

Iuval Goldansky – 161002

Sebastián Leonardi – 161539

Tutor: Luis Calabria

2014

1. Declaración de Autoría

Nosotros, Federico Brunstein, Gonzalo Cumini, Iuval Goldansky y Sebastián Leonardi, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto de grado.
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra.
- En la obra, hemos acusado recibo de las ayudas recibidas.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Federico Brunstein
6 de Marzo de 2014



Gonzalo Cumini
6 de Marzo de 2014



Iuval Goldansky
6 de Marzo de 2014



Sebastián Leonardi
6 de Marzo de 2014

2. Agradecimientos

Nos parece importante agradecer a todas aquellas personas que nos brindaron su apoyo y colaboración durante el desarrollo del proyecto.

En primer lugar agradecemos a nuestro tutor, Luis Calabria, quien constantemente compartió sus ideas, su tiempo y nos guió de forma estupenda durante el camino. Siempre estuvo disponible y de ánimo para brindarnos su ayuda en lo que precisáramos.

Agradecemos también a Eli Barnett, quien ofició de asesor y consultor técnico. Su experiencia en la industria fue vital para guiarnos en ciertos detalles técnicos del desarrollo.

Queremos agradecer a nuestras familias quienes durante el proyecto, como durante la carrera, nos dieron y continúan dando el apoyo necesario para lograr nuestros objetivos sin importar cuáles sean.

A los revisores, por sus sinceras críticas y comentarios constructivos, de los cuales hemos tratado de aprender y aplicar en el proyecto.

Agradecemos a Gonzalo Lema por realizar de manera sumamente profesional la música del juego, y a Elian Cesarkas y Carina David por interpretar las voces de algunos de nuestros personajes.

3. Abstract

Este documento refleja el trabajo realizado por cuatro estudiantes durante el período entre abril del 2013 y marzo del 2014, para la obtención del título de Ingeniero en Sistemas de la Facultad de Ingeniería, de la Universidad ORT. El mismo, narra cómo se llevó a cabo el plan del proyecto cuyo resultado final fue la realización de un videojuego para dispositivos móviles.

Al comenzar este proyecto, los participantes tenían como objetivo desarrollar un videojuego 2D de buena calidad para todo tipo de público. Dicha aplicación se realizó bajo los procesos y prácticas que se emplean actualmente en la industria de los videojuegos y ORTs. Al finalizar la tesis los integrantes del proyecto pretendían obtener un primer acercamiento al desarrollo de un videojuego, y su posterior publicación en una tienda de aplicaciones para dispositivos móviles.

El ciclo de vida seleccionado para realizar esta labor constó de tres grandes etapas: Pre-Producción, Producción y Liberación. En la Pre-Producción se confeccionó la idea del videojuego, se especificaron los requerimientos en dos documentos (GDD y TDD) y se realizó el plan del proyecto. Debido a la falta de experiencia del equipo en el desarrollo de aplicaciones, se realizó además investigación y capacitación. La misma culminó con la realización de distintos prototipos. Posteriormente se pasó a la etapa de Producción, en la cual se desarrolló el videojuego en cinco iteraciones de codificación y pruebas. Finalmente se pasó a la etapa de Liberación, en la cual se realizaron alpha tests, beta tests y corrección de los errores encontrados en estos.

Luego de ejecutado el proyecto, se obtuvo un videojuego para la plataforma Android llamado Crystal Clash (también se cuenta con una versión de escritorio). Debido a ciertos atrasos en el cronograma, se dejaron unos pocos requerimientos como pendientes, los cuales serán realizados en el futuro. Una vez que esto ocurra, el videojuego estará listo para ser comercializado en la tienda de aplicaciones.

4. Palabras Clave

- Android
- Game Concept Document (GCD)
- Game Design Document (GDD)
- LibGDX
- Móvil
- Plan de Proyecto
- Plan de SCM
- Plan de SQA
- Proyecto de Grado
- Simultaneos Turn-Based Tactics
- Technical Design Document (TDD)
- Videojuego

Índice

1.	Declaración de Autoría	2
2.	Agradecimientos.....	3
3.	Abstract	4
4.	Palabras Clave.....	5
5.	Glosario.....	11
6.	Descripción del proyecto.....	14
6.1.	Introducción	14
6.2.	Objetivo del proyecto	14
6.3.	Objetivo del producto.....	15
6.4.	Descripción del equipo	15
6.5.	Alcance del proyecto	16
6.6.	Descripción del Proceso	17
6.6.1.	Funcional	19
6.6.2.	Organizacional	19
6.6.3.	Resultados	20
6.6.4.	Metodología	20
6.6.5.	Comportamiento	22
7.	Ingeniería de Requerimientos	23
7.1.	Introducción	23
7.2.	Conceptualización del problema y su contexto	23
7.3.	Clientes y usuarios	24
7.4.	Requerimientos Funcionales	24
7.4.1.	Introducción	24
7.4.2.	Historia.....	24
7.4.3.	Requerimientos Fundamentales	27
7.4.4.	Requerimientos Importantes	28
7.4.5.	Requerimientos Deseables	29
7.4.6.	Ingresar/Registro	29
7.4.7.	Tutorial.....	30
7.4.8.	Ver Historia	30

7.4.9.	Menú Principal.....	30
7.4.10.	In-Game Name	31
7.4.11.	Estadísticas.....	31
7.4.12.	Posicionar Unidades.....	31
7.4.13.	Ver Turno	32
7.4.14.	Toma de Decisiones	33
7.4.15.	Confirmar Turno.....	34
7.4.16.	Taberna (Tienda virtual).....	34
7.4.17.	Servidor	35
7.4.18.	Actualización	35
7.5.	Requerimientos no funcionales.....	35
7.5.1.	Requerimientos de eficiencia	35
7.5.2.	Requerimientos de jugabilidad.....	36
7.5.3.	Requerimientos de robustez	36
7.5.4.	Requerimientos de modificabilidad	36
7.5.5.	Requerimientos de Idioma	37
7.5.6.	Restricciones de hardware	37
7.5.7.	Restricciones de software	37
7.6.	Prioridad de los requerimientos.....	38
7.6.1.	Fundamentales	38
7.6.2.	Importantes	39
7.6.3.	Deseables.....	39
7.7.	Investigación de herramientas de desarrollo.....	40
7.8.	Estrategia de relevamiento	40
7.8.1.	Idea Fuerza	41
7.9.	Importancia de los recursos	42
7.10.	Game Design Document y Technical Design Document	42
7.11.	Prototipos	43
7.12.	Validación de los requerimientos.....	43
8.	Diseño Arquitectónico	44
8.1.	Introducción	44
8.2.	Elección de plataforma de desarrollo.....	44

8.2.1.	Unity3D	45
8.2.2.	Visual Studio 2010 + MonoGame	45
8.2.3.	Eclipse + LibGDX	46
8.2.4.	Decisión	46
8.3.	Descripción de la solución	47
8.3.1.	Características de calidad	47
8.3.2.	Descripción y justificación de la arquitectura seleccionada	48
8.3.3.	Descripción de los principales mecanismos	50
8.3.4.	Restricciones.....	53
8.4.	Estrategia de desarrollo.....	54
8.4.1.	Plan de versiones	54
8.4.2.	Forma de división del desarrollo	57
9.	Gestión de la calidad	58
9.1.	Introducción	58
9.2.	Proceso de aseguramiento de calidad del producto.....	58
9.2.1.	Proceso de SQA y sus actividades	58
9.2.2.	Definición de estándares	59
9.2.3.	Criterios de aceptación.....	61
9.2.4.	Validación GDD y TDD	63
9.2.5.	Inspecciones	63
9.2.6.	Pruebas	64
9.2.7.	Registro de errores	65
9.2.8.	Refactoreo	66
9.3.	Métricas del producto y proceso.....	66
9.3.1.	Métricas de Proceso	67
9.3.2.	Métricas de Producto	68
9.3.3.	Post-Liberación	75
9.4.	Resultados de las actividades de SQA	76
10.	Gestión de los elementos de configuración	77
10.1.	Introducción	77
10.2.	Proceso de SCM.....	77
10.2.1.	Conceptos	78

10.2.2.	Tecnologías	78
10.2.3.	Código	79
10.2.4.	Recursos artísticos	80
10.2.5.	Documentos	81
10.2.6.	Ciclo de Control de Cambios	81
10.3.	Repositorio	82
10.3.1.	Estructura	82
10.3.2.	Respaldos	83
10.4.	Resultados de las actividades de SCM	83
11.	Gestión de Proyecto	84
11.1.	Introducción	84
11.2.	Ciclo de vida	84
11.2.1.	Pre-Producción	85
11.2.2.	Producción	86
11.2.3.	Liberación	88
11.3.	Cronograma	89
11.3.1.	Cronograma Recursos Gráficos	90
11.3.2.	Cronograma Recursos Sonido	91
11.4.	Entregables por Fase	91
11.4.1.	Entregables de la Pre-Producción	91
11.4.2.	Entregables de la Producción	93
11.4.3.	Entregables de la Liberación	93
11.5.	Esfuerzo	93
11.6.	Pre-Producción	94
11.7.	Producción	95
11.8.	Liberación	98
11.9.	Total del proyecto	100
11.10.	Análisis de Riesgos	103
11.10.1.	Identificación de Riesgos	103
11.10.2.	Escalas para probabilidad e impacto	106
11.10.3.	Análisis cualitativo	106
11.10.4.	Selección de estrategias	107

11.10.5.	Plan de respuesta.....	108
11.10.6.	Reevaluación de riesgos.....	112
11.11.	Desviaciones y Conclusiones	116
11.11.1.	Pre-Producción.....	118
11.11.2.	Producción	119
12.	Conclusiones.....	121
12.1.	Conclusiones generales	121
12.2.	Productos resultantes.....	122
12.3.	Logros obtenidos	122
12.4.	Problemas encontrados.....	123
12.5.	Lecciones aprendidas	123
13.	Bibliografía.....	125
14.	Estructura de la documentación CD.....	126
15.	Anexos	127

5. Glosario

Alpha test: Test que se lleva a cabo por usuarios potenciales en el lugar de desarrollo, en un entorno controlado.

App Store: Es un servicio (tienda virtual) para el iPhone y otros productos de la marca Apple, que permite a los usuarios descargar a través de la web diversos software para sus dispositivos.

Arquitecto: Encargado del diseño de la aplicación.

Art Designer: Encargado del diseño y confección artística del juego.

Audio Designer: Encargado del diseño sonoro del juego.

Beta test: Test que se lleva a cabo por usuarios potenciales en entornos externos al área de desarrollo, que no pueden ser controlados por el desarrollador.

Bug: Defecto de software.

ECS: Elemento de configuración de software.

FPS: Frames per second.

Game Designer: Encargado de diseñar el gameplay y las reglas del videojuego.

Game Engine: Es un framework específico para el desarrollo de videojuegos.

Google Play: Es un servicio (tienda virtual) que brinda Google como tienda de aplicaciones, que permite a los usuarios descargar a través de la web diversos software para sus dispositivos.

GPRO: Gerente de proyecto.

GDD: Game Design Document.

HUD: Head-up display. Forma gráfica de presentar información al usuario.

IDE: Integrated Development Environment.

ING: In-Game Name.

Jugabilidad: Grado de interacción entre el usuario y el sistema.

Kanban: Método para gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, mientras no se sobrecargan a los miembros del equipo.

Kbps: Kilobit por segundo. Es una unidad de medida que se usa en telecomunicaciones e informática para calcular la velocidad de transferencia de información a través de una red.

LSCM: Líder de SCM.

LSQA: Líder de SQA.

Paralax Scrolling: Técnica para trabajar con gráficos en computación para simular profundidad, utilizando distintas capas del fondo que se mueven a distintas velocidades.

SCM: *Software Configuration Management.*

Slideshow: Presentación de una series de imágenes en secuencia, previamente ordenadas según lo deseado.

Smartphones: teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos teléfono móvil convencional y capaz realizar actividades semejantes a una minicomputadora.

SQA: Aseguramiento de calidad de software (Software Quality Assurance).

Tap: Touch Gesture que consiste golpear suavemente la pantalla con la punta del dedo. Es el equivalente al click del mouse en el mundo de las pantallas táctiles.

TDD: Technical Design Document.

Touch Gesture: Método de interacción táctil con la pantalla del dispositivo móvil, involucra el uso de uno o más dedos, y cierto movimiento de los mismos. Resulta en un comando táctil que puede ser interpretado por la aplicación. Existen muchos Touch Gestures diferentes.

2D: Dos dimensiones.

3D: Tres dimensiones.

6. Descripción del proyecto

6.1. Introducción

En los últimos años la venta de dispositivos móviles se mantuvo en constante crecimiento, esto tiene como consecuencia un aumento en las ventas de aplicaciones para dichos dispositivos. Estratégicamente el equipo del proyecto tiene como objetivo incursionar en dicho mercado en crecimiento, especialmente en el área de los videojuegos, ya que este género de aplicaciones para smartphones es el más exitoso de todos.

A través del proyecto el equipo procura investigar los procesos y prácticas que se aplican actualmente en la industria de los videojuegos. Conocer los pasos que permitan al equipo tener un primer acercamiento al desarrollo y venta de juegos a través la tienda de aplicaciones (Google Play o App Store).

Además, el equipo pretende lograr un videojuego 2D para dispositivos móviles que cuenten con sistema operativo Android o iOS. El mismo se desarrollará utilizando un Game Engine multiplataforma que permitirá desarrollar para las plataformas móviles más populares, y un servidor que actuara como nexo entre los jugadores.

El videojuego además contara con una tienda in-game, que permitirá al jugador adquirir nuevo contenido, lo que permitirá monetizar al producto. El sistema estará preparado para recibir este nuevo contenido sin necesidad de realizar grandes cambios.

6.2. Objetivo del proyecto

El objetivo principal del proyecto es desarrollar un videojuego que sirva como base para futuras aplicaciones y videojuegos para dispositivos móviles.

También se espera adquirir los conocimientos necesarios desde el punto de vista de la conceptualización, diseño arquitectónico, gestión, programación, pruebas y otros temas relacionados con el proceso de desarrollo, que sean útiles para continuar en el futuro con la realización de proyectos similares.

Por último, se espera culminar el proyecto con un producto que pueda ser publicado en la tienda de aplicaciones y sirva como punto de partida para dar a conocer el trabajo que realizó el equipo.

6.3. Objetivo del producto

El producto deberá ser un videojuego 2D para dispositivos móviles, del género simultaneous turn-based tactics, de buena calidad, que sea divertido para el público objetivo que sigue este tipo de juegos. Se entiende por buena calidad, una aplicación que tenga buena jugabilidad, esté libre de fallas y contenga recursos gráficos y de sonidos atractivos para los usuarios.

También se busca diseñar un juego que aproveche algunas de las virtudes que hacen al éxito de los dispositivos móviles, tales como la conectividad a internet en todo momento, y el hecho de que los teléfonos móviles se volvieron más personales que las computadoras.

Por último, se busca captar el mayor público objetivo posible, por lo cual se busca un juego fácil de aprender, para los jugadores “casuales”, y a su vez atrapante y desafiante, para los jugadores más “expertos”.

6.4. Descripción del equipo

Este proyecto consiste en la tesis de grado de los integrantes del equipo. Ellos son los encargados de la realización del plan de proyecto, el desarrollo del producto y la entrega final de la tesis.

Dicho equipo está conformado por Federico Brunstein, Gonzalo Cumini, Iuval Goldansky y Sebastián Leonardi, todos estudiantes de la carrera de Ingeniería de Sistemas en la Facultad de Ingeniería de la Universidad ORT, con Luis Calabria asignado como tutor por la Universidad ORT.

Finalmente este proyecto tiene como asesor técnico a Eli Barnett, dueño y fundador de la empresa de videojuegos Kef Sensei. Eli asistió al equipo en la validación de los requerimientos, alcance del proyecto / producto y brindó recomendaciones en general.

6.5. Alcance del proyecto

En base a los objetivos definidos previamente se definió el alcance del proyecto de la siguiente manera:

- Investigación acerca de los temas más importantes para el transcurso del proyecto. Los temas explorados fueron los siguientes:
 - Videojuegos exitosos de la actualidad (¿Cómo innovar?)
 - Game Engines (Realización de prototipos)
 - Herramientas de versionado

- Definir un proceso de desarrollo que se adapte a las particularidades de nuestro proyecto. En él se definen, por ejemplo, el ciclo de vida, la documentación necesaria, los roles y sus responsabilidades, entre otros.

- Desarrollar un videojuego 2D para dispositivos móviles, que pueda ser finalizado en un año de desarrollo, en base a los requerimientos definidos por el equipo. Además se desarrollara el servidor que permitirá la comunicación entre los jugadores.

6.6. Descripción del Proceso

A continuación se describirá el proceso utilizado por el equipo en el proyecto. Inicialmente se mostrará un diagrama que involucra roles, etapas, tareas y actividades específicas del proceso y luego se detallarán las cinco perspectivas del mismo para explicar en detalle las distintas áreas que éste abarca.

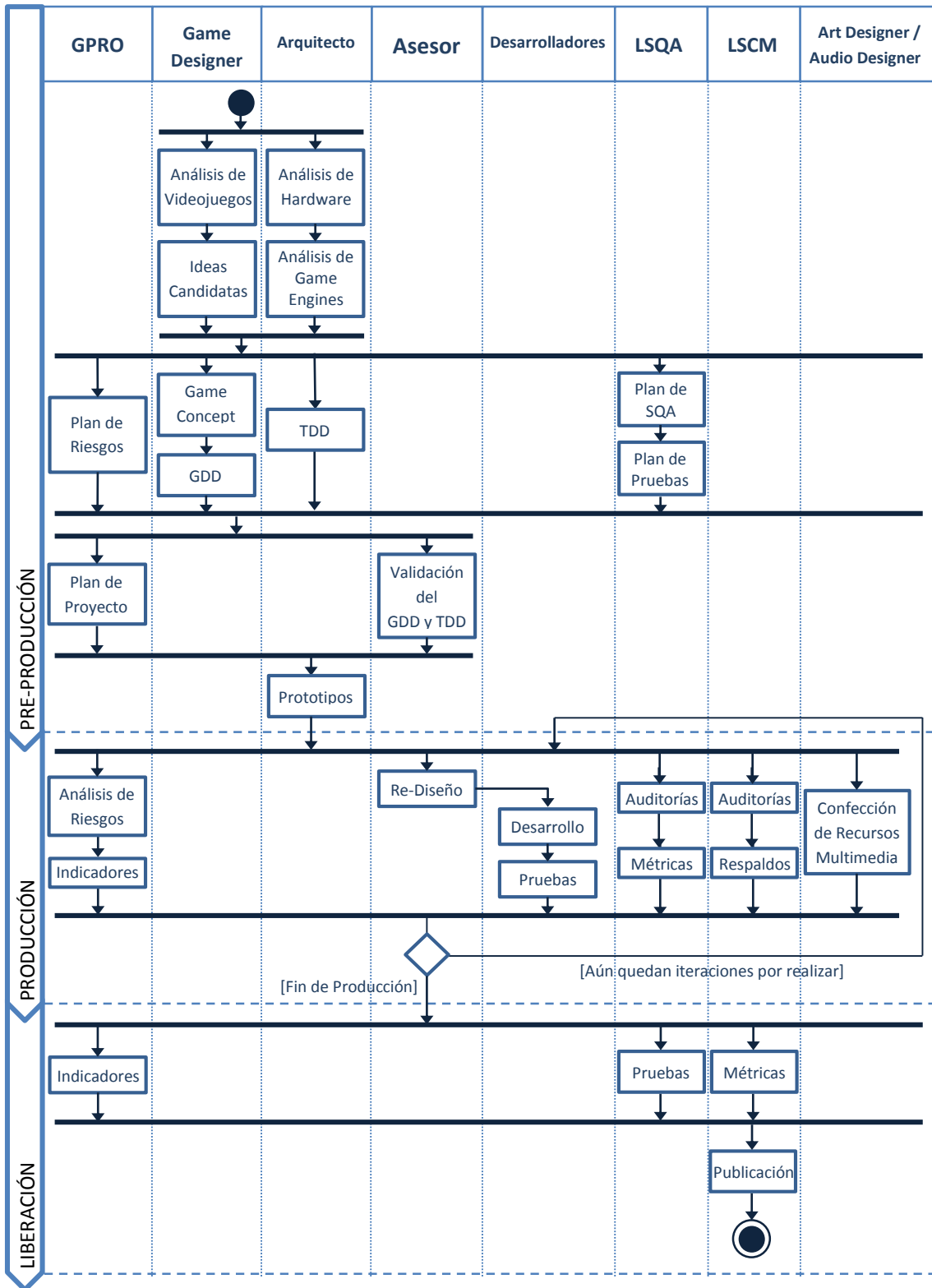


Diagrama 6-1: Actividades del Proceso

6.6.1. Funcional

Debido al poco conocimiento en el área, se decidió realizar un proceso en el cual, en una primera etapa, se realizara una fuerte investigación sobre tecnologías y tendencias de videojuegos en la actualidad. Se planificaron las actividades de las distintas áreas del proyecto como son SQA, SCM, requerimientos, entre otros y finalmente se trabajó en diversos prototipos con la finalidad de conocer las nuevas tecnologías con las que se iba a trabajar.

Luego comenzó la etapa de desarrollo de software. Ésta consistió en varias iteraciones incrementales, que alternaban tareas de re-diseño, desarrollo, pruebas, métricas, entre otras.

Finalmente hubo una etapa de testing y correcciones, acompañada por la recolección de métricas y el estudio de los indicadores planteados en las etapas más tempranas. El proceso termina cuando se sube la aplicación desarrollada a la tienda de aplicaciones.

6.6.2. Organizacional

De manera de distribuir las tareas se definieron los siguientes roles:

- Arquitecto: Encargado del diseño del sistema.
- Art Designer: Encargado del diseño y confección artístico del juego.
- Audio Designer: Encargado del diseño sonoro del juego.
- Desarrollador: Encargado de codificar el videojuego.
- Game Designer: Encargado de definir las características del videojuego.
- GPRO: Gerente de proyecto.
- LSCM: Líder de SCM.
- LSQA: Líder de SQA.
- Tester: Encargado de probar el videojuego.

6.6.3. Resultados

Se obtuvieron grandes resultados del proceso.

- Se diseñó un plan que presenta las actividades adecuadas para que el equipo pueda desarrollar un videojuego, para dispositivos móviles, en un lapso de un año. Esto incluye la definición de tareas del área de gerencia, ingeniería de requerimientos, diseño arquitectónico, desarrollo, calidad, pruebas y configuración.
- Se construyó un videojuego que puede ser ofrecido mediante la tienda de aplicaciones.
- Los integrantes del equipo obtuvieron mayores capacidades en las áreas de planificación y estimación, y conocimiento en lo que refiere a las tecnologías utilizadas.

6.6.4. Metodología

Una vez formalizado el alcance del proyecto y habiendo adquirido un conocimiento global sobre las tecnologías necesaria para llevarlo a cabo, el equipo se encontró en condiciones de decidir qué metodología de trabajo se ajustaría a las necesidades presentadas.

Se utilizó una metodología tradicional para poder definir un calendario a grandes rasgos, es decir, poder determinar las fechas claves que el equipo deberá tener en cuenta en el proceso. Se definieron las fechas de las tres etapas claves para el desarrollo de un videojuego: Pre-Producción, Producción y Liberación.

En la etapa de Producción se utilizó un ciclo de vida incremental-iterativo. Para cada iteración se utilizó un método similar a Kanban. Se definían las tareas que se iban a implementar en esa iteración, y cada integrante se auto-asignaba la tarea que iba a realizar al momento de comenzar. Esta metodología nos permitió trabajar concurrentemente en un mismo proyecto sin solaparnos.

En lo que respecta a las herramientas de apoyo que se utilizaron a lo largo del proyecto podemos mencionar:

- **Adobe Photoshop CS5:** Herramienta de diseño gráfico, utilizado para el desarrollo de todos los recursos necesarios para el juego.
- **Audacity:** Es una herramienta de edición de audio, utilizada para la edición de todos los recursos necesarios para el juego.
- **Eclipse:** Es el IDE recomendado por el creador de LibGDX.
- **Git:** Herramienta gratuita de versionado y control de código distribuido.
- **GitHub:** Alojamiento web gratuito de proyectos Git.
- **Google Drive:** Herramienta gratuita de almacenamiento de archivos en la web, que permite la edición simultánea de documentos por los integrantes del grupo.
- **LibGDX:** Game Engine elegido por el equipo para el desarrollo del juego.
- **Microsoft Office:** Se utilizaron diversas herramientas del paquete de MSOffice para elaborar la documentación del proyecto.
- **Trello:** Aplicación web para gestionar las actividades, que toma elementos de la metodología Kanban.

6.6.5. Comportamiento

El proyecto constó básicamente de tres grandes etapas: Pre-Producción, Producción y Liberación. La Pre-Producción es donde se realizó la investigación planificada, y se definió el videojuego, en la Producción se desarrolló y finalmente la Liberación se probó en profundidad y se produjo el cierre del mismo.

Se optó por utilizar un ciclo de vida incremental-iterativo, dado que para hacer un videojuego se definen primeramente los requerimientos y luego un diseño de alto nivel, para posteriormente ir avanzando en incrementos hasta llegar a la versión final. La gran ventaja de trabajar en incrementos, es que permite obtener versiones jugables de la aplicación en etapas tempranas y así recibir feedback (y posibles puntos de mejora) de diferentes usuarios. El ciclo de vida se describe más en profundidad dentro del capítulo de Gestión del Proyecto.

7. Ingeniería de Requerimientos

7.1. Introducción

Esta sección tiene como fin describir el proceso utilizado para la definición del producto. Se detalla, en primera instancia, el problema a resolver, seguido por la identificación de los potenciales usuarios del producto. También, se plantean los requerimientos funcionales y no funcionales más importantes y se establece el criterio con el cual se priorizaron.

7.2. Conceptualización del problema y su contexto

El problema planteado consiste en desarrollar un videojuego para dispositivos móviles, del género simultaneous turn-bases tactics, que sea divertido para el público objetivo que sigue este tipo de aplicaciones. También se plantea subir el producto resultante a la tienda de aplicaciones con el fin de publicarlo.

Se escogió la tecnología móvil por ser una de las tecnologías con mayor crecimiento hasta el momento y que continúa creciendo. En 2012 existían 6 billones de suscriptores de telefonía móvil (equivalente al 87% de la población mundial), se habían desarrollado más de trescientas mil aplicaciones, las cuales se habían descargado más de once billones de veces, [1].

El proceso para subir la aplicación a la tienda de aplicaciones esta detallado en el Member Center de la página de Apple y en Google Play Developer Console. Para acceder al Developer Console es necesario tener una licencia de desarrollador, la cual tiene un costo de 25 USD. A su vez, la licencia de desarrollador Apple para ingresar al Member Center, tiene un costo de 100 USD (el equipo decidió que no se compraría esta licencia hasta haber publicado el juego en Google Play, y ver la respuesta del público), [2] y [3].

7.3. Clientes y usuarios

Si bien el proyecto fue idea original de los integrantes del grupo de tesis y son dueños del mismo, el rol del cliente frente a la facultad lo ocupa Eli Barnett. Él asesoró al equipo ayudando a determinar un alcance acorde a una tesis de grado.

Los usuarios del producto Crystal Clash serán aquellas personas que descarguen el videojuego de la tienda de aplicaciones. La aplicación está orientada a todo tipo de público objetivo, tanto jugadores “casuales” como jugadores “expertos”.

7.4. Requerimientos Funcionales

7.4.1. Introducción

En esta sección plantearemos los requerimientos funcionales del videojuego “Crystal Clash”, comenzando por la historia, luego listando los requerimientos funcionales y detallando los más importantes.

7.4.2. Historia

Desde los orígenes, siempre han existido las **cuatro razas primigenias**. En el sur, vive el pueblo de la **tierra**, cuyos guerreros son los más resistentes de todos, denominados los **Toh’Chi**. En el este, viven los guerreros más peligrosos, los que dominan el **fuego**, son conocidos como los **Kah’Shin**. En el oeste se encuentran quienes dominan el **viento**, cuyos guerreros se destacan por su increíble velocidad en batalla, son llamados los **Fu’Hari**. Por último, en el norte, viven aquellos que dominan el **agua**, expertos en técnicas de curación, se hacen llamar los **Sui’Ame**.



Imagen 7-1: Imagen Historia 1

Las **leyendas** hablaban de una **quinta raza**, extremadamente poderosa, caminantes de las sombras. Muy pocos creían en estas historias, los sabios desaprobaban cualquier investigación y acusaban de herejes a quienes se interesaban en ellas. En el fondo, los sabios en verdad temían que las leyendas fueran ciertas. Un valiente investigador, al ser expulsado de las bibliotecas de los sabios, logró guardar un mapa de donde se creía que se encuentra la entrada al templo de los ancestros. El templo que guarda en sus profundidades el inmenso **poder de las tinieblas**, que alguna vez perteneció al pueblo de los **Dhen'Zetu**.



Imagen 7-2: Imagen Historia 2

Este mapa se convirtió en su obsesión durante los siguientes años, y no descansó hasta conformar el equipo de expedicionarios que lo acompañarían en **la búsqueda que cambiaría el mundo para siempre**. Finalmente, luego de muchas travesías, justo cuando comenzaban a pensar que los sabios tenían razón, descubrieron que las leyendas eran ciertas. Encontraron un gran portal, con inscripciones que claramente pertenecían a los ancestros, no había duda, **estaban frente al templo de los antiguos Dhen'Zetsu**.



Imagen 7-3: Imagen Historia 3

En la cámara principal, vieron un **gran cristal**, su color era negro, más oscuro que una noche sin luna. Y de pronto, uno de ellos escuchó una voz que lo llamaba desde el interior del cristal, al cual se acercó y trató de tomarlo. Al hacer contacto con el mismo, rompió el antiguo sello que mantenía confinado el poder de las tinieblas. Al liberarse, **el cristal se fragmentó en cientos de millones de pedazos, dispersándose por todo el mundo**.

En ese momento, personas de todos los pueblos despertaron en su interior el poder de las tinieblas, y fueron capaces de utilizarlo de la misma forma que lo hacían los antiguos Dhen'Zetsu. A partir de ese día, **una quinta raza había renacido**, durante tanto tiempo olvidada, ahora formaban parte de nuestro mundo. **Los fragmentos del cristal son extremadamente poderosos**, y combinando varios de éstos se puede obtener un poder increíble. No pasó mucho tiempo para que diferentes grupos se aliaran para comenzar a **buscar los fragmentos dispersos** alrededor del mundo.



Imagen 7-4: Imagen Historia 4

Tú, como líder de un pequeño grupo que comienza su búsqueda, debes recolectar la mayor cantidad de fragmentos posible si deseas dominar a los demás! Buena suerte!

7.4.3. Requerimientos Fundamentales

Nº	Requerimiento
1	Sign Up
2	Log In
3	Log Out
13	Listar batallas en curso
14	Pedir nueva batalla
15	Emparejar los jugadores que tienen petición de nueva partida
17	Refrescar la lista de batallas en curso
18	Ver mi avatar en la lista de batallas en curso
19	Ver el avatar de mi enemigo en la lista de batallas en curso
21	Ver mi IGN en la lista de batallas en curso
22	Ver el IGN de mi enemigo en la lista de batallas en curso
23	Ver mis estadísticas en la lista de batallas en curso
24	Ver las estadísticas de mi enemigo en la lista de batallas en curso
26	Ver el turno de la batalla en la lista de batallas en curso
27	Indicador en verde cuando la batalla se puede continuar
28	Indicador en rojo cuando la batalla no se puede continuar
29	Indicador en blanco cuando el contrincante se ha rendido
32	Mostrar menú de posicionar unidades en el primer turno
33	Posicionar una unidad en el primer turno
38	Deshacer todos los posicionamientos en el primer turno
39	Confirmar los posicionamientos en el primer turno
40	Opción para ver lo que ocurrió en el turno pasado
44	Reproducir animaciones de lo que ocurrió en el turno pasado
46	Mostrar barras de salud de los personajes en el mapa
47	Desplegar HUD de acciones al seleccionar un personaje aliado

48	Ordenar que un personaje entre en postura defensiva
49	Ordenar que un personaje ataque una celda
50	Ordenar que un personaje se mueva a una celda
51	Rendirse desde la lista de batallas en curso
59	Fin de Batalla
60	Mostrar opción para volver a la lista de batallas en curso desde el cartel de fin de batalla
61	In Game menú

Tabla 7-1: Priorización de Requerimientos Fundamentales

7.4.4. Requerimientos Importantes

Nº	Requerimiento
4	Tutorial
5	Ofrecer hacer el Tutorial la primera vez
6	Saltar el Tutorial
7	Revivir el Tutorial
20	Cambiar Avatar
25	Activar o desactivar modo silencioso en la lista de batallas en curso
30	Nubes en movimiento detrás de la lista de batallas en curso
31	Parallax scrolling en la lista de batallas en curso
36	Reorganizar unidades en el campo de batalla en el primer turno
37	Remover personajes y/o gema del campo de batalla en el primer turno
45	Ver estado de un personaje
52	Rendirse desde la pantalla de primer turno
53	Rendirse desde la pantalla de toma de decisiones
54	Ver mi avatar en el campo de batalla
55	Ver el avatar de mi enemigo en el campo de batalla
56	Ver mi IGN en el campo de batalla
57	Ver el IGN de mi enemigo en el campo de batalla
58	Activar o desactivar modo silencioso en el campo de batalla

Tabla 7-2: Priorización de Requerimientos Importantes

7.4.5. Requerimientos Deseables

Nº	Requerimiento
8	Historia
9	Mostrar la historia la primera vez
10	Saltear la historia
11	Revivir la historia
12	Single player
16	Emparejar los jugadores que tienen petición de nueva partida
34	Posicionar un héroe en el primer turno
35	Seleccionar una gema en el primer turno
41	Opción para saltarse ver las animaciones de lo que ocurrió en el turno pasado, antes de comenzar a reproducirlas
42	Opción para saltar las animaciones de lo que sucedió en el turno pasado mientras las mismas se están reproduciendo
43	Opción de repetir las animaciones de lo que sucedió en el turno pasado una vez que las mismas han concluido
62	Taberna
63	Contratar personaje o comprar gema

Tabla 7-3: Priorización de Requerimientos Deseables

7.4.6. Ingresar/Registro

Los nuevos jugadores deberán poder ingresar un usuario y una contraseña válidos para crear una nueva cuenta. El jugador que tenga una cuenta de jugador existente deberá poder ingresar sus datos para autenticarse e ingresar al juego. El juego deberá recuperar toda la información de la cuenta, incluyendo el balance actual de fragmentos de cristal, las batallas en curso, y los personajes y gemas que posee. El jugador mientras esta logueado podrá desloguearse haciendo Log Out, desvinculando su cuenta del juego para permitir que otro jugador ingrese al juego con otra cuenta.

7.4.7. Tutorial

El juego deberá contar con un tutorial que enseñe los conceptos básicos del juego. El jugador que acaba de crear su cuenta y hace Log In en el juego por primera vez, deberá ser recibido con una invitación a realizar el tutorial. Debe contar con la posibilidad de no realizar el tutorial y seguir adelante. Por otra parte, debe contar con la opción en el menú principal de volver a realizar el tutorial.

7.4.8. Ver Historia

El juego deberá contar con un video que cuente la historia de Crystal Clash. El jugador que acaba de crear su cuenta y hace Log In en el juego por primera vez, deberá poder ver este video. Deberá contar con la posibilidad de saltar la reproducción del mismo y seguir adelante. Por otra parte, deberá contar con la opción en el menú principal para revivir la historia y volver a ver el video cuantas veces desee.

7.4.9. Menú Principal

El jugador podrá ver sus batallas en curso en forma de lista en el Menú Principal. Podrá también pedirle al juego que le busque un oponente para pelear una batalla 1v1, y refrescar la lista de batallas en curso para que se actualicen según su estado más actual, así como para que aparezcan nuevas batallas que hayan comenzado producto de la petición de nueva batalla.

Cada jugador contará con un avatar en el menú principal, el cual podrá cambiar haciendo tap sobre el mismo. Aparecerá una pantalla donde se muestren todos los avatares disponibles y pueda elegir uno de su gusto como nuevo avatar.

En cada batalla, el jugador podrá ver el avatar del jugador enemigo, el número de turno actual de cada batalla y un indicador para saber si la batalla puede continuar o se está esperando por el oponente. Contará con una opción para rendirse, y se debe indicar si el jugador oponente se ha rendido.

Deberá existir una opción para activar o desactivar modo silencioso (sin sonido) de la aplicación.

7.4.10. In-Game Name

El servidor generará para cada jugador un nombre aleatorio. Cada jugador podrá ver este nombre en el menú principal, junto con su avatar. Además, en cada batalla, se mostrará el nombre del contrincante.

7.4.11. Estadísticas

El jugador podrá ver la cantidad de victorias, derrotas y empates que ha acumulado a través del tiempo, esta información estará disponible en el menú principal, junto con su avatar. En cada batalla, podrá ver las estadísticas del enemigo (vera la cantidad de victorias).

7.4.12. Posicionar Unidades

El jugador que selecciona entrar a una batalla para jugar su primer turno, accederá a la pantalla de primer turno, donde tendrá un menú con las unidades, héroes y gemas que posee para elegir las y usarlas en batalla. Al costado de este menú, el jugador podrá ver el lado del terreno que le fue asignado, y verá el área designada para posicionar sus personajes y la gema.

Deberá seleccionar una unidad del menú de unidades y luego seleccionar la celda donde quiere posicionarla dentro de la zona indicada para posicionar personajes. Podrá posicionar como máximo 5 unidades, pero deberá seleccionar al menos 3 de ellas, puede repetir unidades en el campo de batalla. Una vez ubicada, podrá cambiarla de lugar, removerla del juego.

El jugador también podrá seleccionar un héroe del menú de héroes y luego seleccionará la celda donde quiere posicionarlo dentro de la zona indicada para posicionar personajes.

Por último, podrá seleccionar una gema del menú de gemas. Tiene la opción de usar una gema, pero no es obligatorio hacerlo, puede ir a batalla sin seleccionar una gema.

7.4.13. Ver Turno

El jugador que selecciona entrar a una batalla en un turno que no sea el primero, será recibido con un menú para ver lo que ocurrió en el turno pasado en forma de animaciones. Este usuario podrá reproducir la animación, pausarla, saltarla y volver a reproducirla cuando haya terminado.

Las animaciones se reproducirán de la siguiente manera: en primer lugar, los ataques de todos los personajes en el terreno que tenían órdenes de atacar con habilidades de tipo cuerpo a cuerpo. Luego verá las animaciones de movimiento de todos los personajes en el mapa que tenían órdenes de moverse, y por último las animaciones de ataque de todos los personajes en el terreno que tenían órdenes de atacar con habilidades de tipo a distancia. Los personajes que tenían órdenes de tomar una postura defensiva, entrarán en dicha postura al comienzo del turno.

Todas las unidades del mapa atacarán cuerpo a cuerpo al mismo tiempo, se moverán al mismo tiempo y atacarán a distancia al mismo tiempo, no se turnan para hacer estos movimientos. Por último todas las unidades que hayan quedado con cero puntos de salud luego de los eventos del turno, morirán, mostrando su animación de muerte. Y todos los efectos de todas las habilidades usadas que tengan que mantenerse en el siguiente turno serán manifestados, como ser stuns, roots, silences, DoT, etc.

Luego de reproducir las animaciones, si alguno de los dos jugadores se queda sin personajes en el campo de batalla entonces pierde la batalla. Si ambos jugadores se quedan sin personajes en el mismo turno, entonces es un empate. En cualquier caso se muestra en pantalla un cartel que le indica al jugador el resultado. El servidor actualizará las victorias, derrotas y empates de ambos jugadores. Así como también el nivel y el balance de los fragmentos de cristal de ambas cuentas de jugador.

7.4.14. Toma de Decisiones

Al momento de tomar decisiones, el jugador podrá seleccionar una unidad aliada o enemiga para ver detalles de la misma, como raza, clase, salud actual, velocidad, etc. Esta información aparecerá en forma de Pop-Up, a modo de información contextual.

También se deberá mostrar barras de salud de los personajes en el mapa. Estas se verán sobre los personajes en todo momento. Estas barras de salud se irán actualizando cada vez que se pase de turno para ofrecer información precisa sobre el estado de los ejércitos.

Cuando se selecciona un personaje aliado se deberá desplegar HUD de acciones. Este HUD contará con las opciones de atacar, moverse o entrar en postura defensiva. Si el jugador hace tap en cualquier otra parte de la pantalla entonces el HUD se cierra. En caso de hacer tap sobre otro personaje aliado, entonces el HUD se vuelve a abrir en dicho personaje.

Postura defensiva: Sucede cuando el jugador seleccionó dicha opción en el HUD de acciones. Esta opción se confirma automáticamente al momento de seleccionarla.

Ordenar ataque: Al seleccionar la opción de atacar en el HUD de acciones, se iluminarán las celdas aptas para marcar como objetivo. El jugador seleccionará una de ellas y luego hará tap en cualquier parte fuera de la zona iluminada en rojo para confirmar el ataque. Si el jugador hace tap fuera de la zona iluminada antes de haber seleccionado un objetivo, entonces la orden se cancela. Cuando se haya seleccionado un objetivo de ataque, la celda tomará un aspecto especial para indicar que es el objetivo del ataque, así como también se dibujará una trayectoria de proyectil entre el personaje que está atacando, y el objetivo del ataque.

Ordenar movimiento: Al seleccionar la opción de moverse en el HUD de acciones, se iluminaran las celdas aptas para ser destino del movimiento (según la movilidad del personaje). El jugador seleccionara una de ellas y luego hará tap en cualquier parte fuera de la zona iluminada para confirmar el movimiento (el personaje tomara el camino más corto al destino). Si el jugador hace tap fuera de la zona iluminada antes de haber seleccionado un destino, entonces la orden se cancela. Cuando se haya seleccionado una celda destino para el movimiento, aparecerá un fantasma del personaje en la posición de destino con la animación de caminata, así como una trayectoria del camino que el personaje recorrerá en el siguiente turno.

Deberá existir también un menú secundario, el cual contara con las opciones de activar o desactivar el modo silencioso (sin sonido), rendirse, cancelar todas las decisiones actuales, volver al menú principal y volver a la batalla en curso.

En el campo de batalla se deberá poder ver el avatar el jugador, en el lado del campo de batalla que se le ha asignado, así como también su In-Game Name. En el lado del campo enemigo, se deberá ver el avatar e In-Game Name del contrincante.

7.4.15. Confirmar Turno

El jugador, cuando haya terminado de tomar sus decisiones, tendrá un botón para indicar que está listo y enviarlas al servidor, luego el juego volverá al Menú Principal.

7.4.16. Taberna (Tienda virtual)

El jugador podrá acceder a la taberna para adquirir nuevas unidades, héroes y gemas. Dicho personaje o gema pasara a estar disponible en el menú de unidades, héroes o gemas de la pantalla de primer turno. Al contratar una unidad o héroe, o comprar una gema, el precio se descuenta del balance actual del jugador.

7.4.17. Servidor

Existirá un servidor el cual se encargara de manejar los usuarios que desean jugar una nueva partida, e ir emparejándolos para crear nuevas batallas 1v1.

7.4.18. Actualización

Se resolvió que no se implementarían los Héroes ni las Gemas, así como tampoco la opción de saltar o volver a reproducir la animación del turno debido a falta de tiempo. Es por esto que los requerimientos relacionados con esto fueron sacados del alcance del proyecto. Se tomó esta decisión para priorizar la calidad de lo realizado, dedicándonos a arreglo de bugs, mejoras de código (refactoreo), y la experiencia de usuario, explicado a continuación:

Mejoras de código: Refactoreo, para lograr un código más limpio y fácil de entender. Además se realizaron cambios para mejorar la eficiencia, como por ejemplo, reúso de recursos, líneas de tiempo de las animaciones.

Experiencia de Usuario: Se decidió animar el Menú Principal. Se cambió el diseño de la lista de batallas, agregando Parallax Scrolling de fondo.

7.5. Requerimientos no funcionales

7.5.1. Requerimientos de eficiencia

Para dar sensación de suavidad, el juego debe correr como mínimo a 30 fps.

Al iniciar la aplicación, el tiempo de carga deberá ser menor a 5 segundos, mientras que el iniciar un round, el tiempo de carga deberá ser menor a 10 segundos.

7.5.2. Requerimientos de jugabilidad

La jugabilidad se define como la forma en que el jugador interactúa con el videojuego. Para lograr una buena interacción se deberá contar con los siguientes atributos:

- **Menús intuitivos:** La navegabilidad dentro del juego debe ser intuitiva para el usuario. Para lograr esto, se deberá utilizar palabras clave para los menús, de forma que sean representativos. Además no deberá tener que realizar más de 5 interacciones con el sistema para poder realizar alguna acción.
- **Comandos intuitivos:** La manera de realizar las acciones y movimientos dentro de una partida debe ser fácil y sencillos de aprender.
- **No linealidad:** Para que el juego no sea lineal y no aburra al jugador, se deberá asegurar de que cada partida se pueda resolver de más de una forma posible sin que haya estrategias dominantes.

7.5.3. Requerimientos de robustez

Por cada corrida de un round, se espera alcanzar un valor de 0.05 fallas. Esto se traduce a una falla cada 20 corridas.

7.5.4. Requerimientos de modificabilidad

El juego debe quedar abierto a futuros cambios. Deberá permitir la inclusión de nuevos personajes (a su vez, nuevas clases de unidades y nuevos elementos), héroes, nuevos aspectos para los personajes y nuevos mapas.

7.5.5. Requerimientos de Idioma

Todos los textos del juego se especificaran en un archivo fuera del juego para poder tener múltiples lenguajes. De esta forma, se pueden traducir todos los textos a diferentes idiomas y en el juego se especifica que archivo se debe leer según el idioma seleccionado por el usuario.

7.5.6. Restricciones de hardware

Para poder ejecutar el juego correctamente, se recomienda que el dispositivo tenga características similares o superiores a las siguientes:

- Memoria RAM: 128mb
- Procesador: 800 mhz
- GPU: 133 Mhz

7.5.7. Restricciones de software

Se requerirá de un dispositivo móvil, ya sea smartphone o tablet, el cual posea el sistema operativo Android (versión igual o superior a 2.3) o iOS (versión igual o superior a 5.1).

Se decidió que no estará disponible la versión de iOS hasta luego de la liberación para Android, momento en el cual el equipo podrá comprobar su impacto en el público objetivo.

También estará disponible la versión Stand-Alone para computadores personales, el cual deberá contar Java 6.

7.6. Prioridad de los requerimientos

De la descripción anterior se extrajo la siguiente lista. La determinación de la prioridad de los requerimientos se basó en la perspectiva del usuario. Todo el equipo de proyecto intervino en la determinación de los mismos así como también el asesor.

7.6.1. Fundamentales

Requerimiento	Grado de Dificultad
Sign Up	Intermedio
Log In	Intermedio
Log Out	Bajo
Listar batallas en curso	Alto
Pedir nueva batalla	Bajo
Emparejar los jugadores que tienen petición de nueva partida	Alto
Refrescar la lista de batallas en curso	Intermedio
Ver mi avatar en la lista de batallas en curso	Bajo
Ver el avatar de mi enemigo en la lista de batallas en curso	Bajo
Ver mi IGN en la lista de batallas en curso	Bajo
Ver el IGN de mi enemigo en la lista de batallas en curso	Bajo
Ver mis estadísticas en la lista de batallas en curso	Intermedio
Ver las estadísticas de mi enemigo en la lista de batallas en curso	Intermedio
Ver el turno de la batalla en la lista de batallas en curso	Intermedio
Indicador en verde cuando la batalla se puede continuar	Alto
Indicador en rojo cuando la batalla no se puede continuar	Alto
Indicador en blanco cuando el contrincante se ha rendido	Alto
Mostrar menú de posicionar unidades en el primer turno	Alto
Posicionar una unidad en el primer turno	Alto
Deshacer todos los posicionamientos en el primer turno	Intermedio
Confirmar los posicionamientos en el primer turno	Alto
Opción para ver lo que ocurrió en el turno pasado	Bajo
Reproducir animaciones de lo que ocurrió en el turno pasado	Alto
Mostrar barras de salud de los personajes en el mapa	Intermedio
Desplegar HUD de acciones al seleccionar un personaje aliado	Intermedio
Ordenar que un personaje entre en postura defensiva	Bajo
Ordenar que un personaje ataque una celda	Alto
Ordenar que un personaje se mueva a una celda	Alto
Rendirse desde la lista de batallas en curso	Intermedio
Fin de Batalla	Alto
Mostrar opción para volver a la lista de batallas en curso desde el	Bajo

cartel de fin de batalla	
In Game menú	Intermedio

Tabla 7-1: Tabla Requerimientos Fundamentales

7.6.2. Importantes

Requerimiento	Grado de Dificultad
Tutorial	Alto
Ofrecer hacer el Tutorial la primera vez	Bajo
Saltear el Tutorial	Bajo
Revivir el Tutorial	Bajo
Cambiar Avatar	Intermedio
Activar o desactivar modo silencioso en la lista de batallas en curso	Intermedio
Parallax scrolling en la lista de batallas en curso	Alto
Reorganizar unidades en el campo de batalla en el primer turno	Alto
Remover personajes y/o gema del campo de batalla en el primer turno	Alto
Ver estado de un personaje	Intermedio
Rendirse desde la pantalla de primer turno	Intermedio
Rendirse desde la pantalla de toma de decisiones	Intermedio
Ver mi avatar en el campo de batalla	Bajo
Ver el avatar de mi enemigo en el campo de batalla	Bajo
Ver mi IGN en el campo de batalla	Bajo
Ver el IGN de mi enemigo en el campo de batalla	Bajo
Activar o desactivar modo silencioso en el campo de batalla	Intermedio

Tabla 7-2: Tabla Requerimientos Importantes

7.6.3. Deseables

Requerimiento	Grado de Dificultad
Historia	Alto
Mostrar la historia la primera vez	Bajo
Saltear la historia	Bajo
Revivir la historia	Bajo
Single player	Alto
Emparejar los jugadores que tienen petición de nueva partida	Alto
Posicionar un héroe en el primer turno	Alto

Seleccionar una gema en el primer turno	Alto
Opción para saltarse ver las animaciones de lo que ocurrió en el turno pasado, antes de comenzar a reproducirlas	Alto
Opción para saltar las animaciones de lo que sucedió en el turno pasado mientras las mismas se están reproduciendo	Alto
Opción de repetir las animaciones de lo que sucedió en el turno pasado una vez que las mismas han concluido	Alto
Taberna	Alto
Contratar personaje o comprar gema	Intermedio

Tabla 7-3: Tabla Requerimientos Deseables

7.7. Investigación de herramientas de desarrollo

Junto a la definición de los requerimientos, el equipo de proyecto investigó las herramientas que podrían ser utilizadas para la realización de un videojuego de las características planteadas en el GDD y TDD. Posteriormente seleccionó las más adecuadas y comenzó una capacitación sobre el uso de las mismas. Esto último se llevó a cabo mediante la construcción de distintos prototipos. Los prototipos tuvieron como objetivo también chequear la viabilidad técnica del producto. De esta manera se pudo estimar mejor el esfuerzo necesario para completar cada tarea.

El punto de decisión más importante, en cuanto a herramientas de desarrollo, fue la selección del Game Engine. Luego de evaluar varias opciones, se decidió por LibGDX. Más adelante se detalla el análisis de las distintas opciones consideradas.

7.8. Estrategia de relevamiento

La idea del juego surgió dentro del equipo. Se quería desarrollar un videojuego del género “estrategia”, y dentro del mismo, se optó por el subgénero “por turnos” (Turn-based tactics). La gran mayoría de los juegos de este género se basan en “ver y reaccionar”, cosa que el equipo quería evitar. Para lograr esto, se decidió que los turnos se debían ejecutar simultáneamente. Investigando este tipo de características, descubrimos que el género al cual apuntábamos era Simultaneous Turn-Based Tactics.

El equipo además quería que el juego no requiera que ambos jugadores estén conectados simultáneamente, ni demasiado tiempo. Siguiendo el ejemplo del conocido juego “Draw Something”, se agregó el concepto de “Sesiones”, [4].

Luego de haber decidido como funcionaria el juego, se lo planteamos a nuestro asesor, el cual nos sugirió que lo probáramos. Al equipo le pareció una buena idea y decidió hacerlo. Se construyó una versión del tablero en cartulina, se hicieron varias unidades en papel y se pasó una tarde jugando para comprobar que sea divertido.

Por último, cada integrante del equipo hablo con amigos y conocidos acerca del juego, para obtener rápidamente una idea inicial sobre la viabilidad del mismo.

Se pensó la historia del juego y se realizó un Game Concept Document del mismo, el cual plasma la idea fuerza.

7.8.1. Idea Fuerza

Es un juego de estrategia para dispositivos móviles. El juego se trata de una batalla entre dos o más jugadores, donde el terreno se encuentra dividido en hexágonos. El jugador tendrá el control de ciertos personajes (sus guerreros) y podrá decidir movimientos tácticos contra su adversario. El objetivo del juego es eliminar las fuerzas enemigas, u obligarlo a rendirse.

La principal característica del juego, es que la batalla se libra por etapas, y entre cada etapa es como si se detuviera el tiempo. Durante este momento, los jugadores tendrán que tomar sus decisiones para la siguiente etapa. Una vez que ambos jugadores han terminado de planificar, se ejecutarán todas las decisiones que han tomado para esta etapa. Luego de que todas las acciones se han ejecutado, se vuelve a detener el tiempo y es momento de tomar nuevas decisiones. Las decisiones de ambos jugadores se ejecutan simultáneamente. Cada jugador se verá obligado a predecir qué hará su oponente al momento de pensar su jugada, ya que los movimientos del oponente solo serán visibles cuando el tiempo vuelva a correr.

7.9. Importancia de los recursos

Al ser un proyecto de desarrollo de videojuegos, el arte pasa a ser una parte muy importante del software. Gracias a que el equipo contaba con un integrante capaz de construir estos recursos, no se dependía de agentes externos. Por otra parte, esto provocó que dos de los integrantes se dedicaran exclusivamente a la creación de recursos (gráficos y de audio) y no al desarrollo.

7.10. Game Design Document y Technical Design Document

El proyecto tiene como principal objetivo desarrollar y publicar un videojuego. El mismo queda especificado mediante dos documentos, el Game Design Document (Véase completo en el ANEXO E – GDD) y el Technical Design Document (Véase completo en el ANEXO F – TDD).

El Game Design Document presenta las funcionalidades del videojuego, describiendo la jugabilidad del mismo. Este documento especifica los mecanismos del videojuego, especificando lo que puede hacer el usuario. Además sirve de referencia tanto para los desarrolladores y los artistas, así como el gerente del proyecto, quien podrá realizar una mejor estimación de tiempos y costos.

El Technical Design Document presenta los requerimientos técnicos referentes al videojuego. En otras palabras, describe los requerimientos no funcionales y la solución arquitectónica, a alto nivel, que se utilizará para solucionar estos problemas.

Anteriormente se escribió otro documento llamado Game Concept Document (Véase completo en el ANEXO D – GCD), el cual describe la idea fuerza del juego. Este documento se utiliza para ofrecer un primer acercamiento al juego.

7.11. Prototipos

A principios de la Pre-Producción se definió una lista de prototipos para realizar a lo largo de la misma, para mitigar incertidumbres técnicas y poder comenzar la Producción con la certeza de que íbamos poder realizar lo deseado.

La prototipos realizados son: Deploy en Android/iOS, Dibujar el mapa, Servidor, Networking (Comunicación Cliente-Servidor), Animaciones, Situar/Mover unidades, Audio y HUD.

Cada prototipo se realizó de manera independiente para probar lo específico de cada uno de ellos. Algunos de estos fueron reutilizados en la Producción, debiendo tener que integrarlos.

Gracias al desarrollo de estos prototipos, el equipo pudo familiarizarse con el Game Engine seleccionado (LibGDX) y aprendió a utilizarlo.

7.12. Validación de los requerimientos

No hubo una instancia específica de validación de requerimientos ya que no contábamos con un cliente real debido a que el videojuego es una idea original del equipo de tesis.

De todas formas, el equipo contaba con un asesor técnico, quien nos ayudó en medir el alcance de los requerimientos definidos para poder completarlos en el año de trabajo. Durante la Producción, tuvimos varias reuniones con él, donde nos brindaba feedback y oportunidades de mejora en base a su experiencia.

Las recomendaciones realizadas por Eli, referían a la jugabilidad. La mayoría de ellas implicaban cambios en los requerimientos y luego el equipo evaluaba el impacto de realizar estas modificaciones en los requerimientos originales e implementarlas.

8. Diseño Arquitectónico

8.1. Introducción

En este capítulo se describen y justifican las decisiones arquitectónicas tomadas a lo largo del proyecto.

Las tareas referidas al diseño de la arquitectura se realizaron considerando los diferentes atributos de calidad establecidos.

Se describirá, a continuación, el proceso y los resultados de cada una de estas instancias, así como los principales problemas encontrados y las lecciones aprendidas sobre el tema.

8.2. Elección de plataforma de desarrollo

El equipo no tenía ningún tipo de restricción respecto al entorno de desarrollo, por lo que la elección del mismo quedó a criterio del grupo. Se analizaron tres herramientas distintas.

- Unity3D
- Visual Studio 2010 + MonoGame
- Eclipse + LibGDX

8.2.1. Unity3D

- Está pensado para el desarrollo de juegos 3D. Si bien podríamos desarrollar el juego en 2D ignorando la tercer coordenada de todos nuestros objetos (mantenerla igual y constante para todos), los cálculos se realizan utilizando las tres coordenadas. Estos cálculos son más complejos y consumen más recursos de hardware, los cuales, son limitados en los dispositivos móviles.
- La versión gratuita no permite la exportación del juego desarrollado para dispositivos móviles, se debe comprar la licencia de cada compilador (uno para Android y uno para iOS).
- Ninguno de los integrantes tenía conocimientos de Unity3D, todos debíamos aprender a utilizar la herramienta, lo cual se podría considerar como un riesgo tecnológico.
- Por último, también se evaluó la herramienta de versionado que se utilizaría. El equipo decidió que quería utilizar Git. Investigando un poco, se descubrió que esta herramienta no funcionaba muy bien en Unity3D debido a los tipos de archivos que maneja éste.

8.2.2. Visual Studio 2010 + MonoGame

Se evaluó Visual Studio como herramienta de desarrollo para el video juego, luego de analizar MonoGame. Todos los integrantes del equipo estamos familiarizados con esta herramienta y contamos con la licencia otorgada por la universidad.

MonoGame es una implementación del framework Microsoft XNA 4. Se instala como un plug-in para Visual Studio, el cual agrega tres nuevos tipos de proyectos que se pueden crear en el mismo. Nosotros utilizaremos uno de estos nuevos proyectos para el desarrollo del videojuego.

8.2.3. Eclipse + LibGDX

Se eligió utilizar Eclipse luego de decidir utilizar LibGDX. Es el entorno de desarrollo para aplicaciones Android (java) recomendado por Google y por el creador de LibGDX.

LibGDX es un framework open-source y gratuito, que utilizamos para el desarrollo del juego. Está pensado para el desarrollo en Java, lenguaje con el cual el equipo está familiarizado, por lo que no implicó ningún tipo de riesgo. El framework facilita el manejo de imágenes, sonidos, interacción con el usuario, y la exportación a Android entre otras cosas. Además, existe una gran comunidad detrás de LibGDX, la cual ha ayudado a optimizar y mejorar este framework.

8.2.4. Decisión

En primera instancia se había elegido MonoGame. Luego se decidió cambiar de ambiente de desarrollo se debió a un problema que surgió al intentar exportar a los dispositivos móviles. Para realizar la exportación (a ambos sistemas operativos) era necesario utilizar un programa externo, Xamarin, el cual cuenta con una versión gratis (de desarrollador) complicada de utilizar haciendo el proceso demasiado engorroso. Además aplica una restricción de uso de 24 horas a la aplicación exportada. Las licencias que ofrece Xamarin para hacer el proceso de exportación más ameno, y que quitan la restricción de uso, son demasiado costosas y no entraban en el presupuesto del equipo.

El nuevo ambiente de desarrollo, LibGDX, permite trabajar sobre un proyecto, y exportar directamente a una aplicación Stand-Alone, Web y Android. Para realizar la exportación a iOS, aún es necesario utilizar Xamarin.

8.3. Descripción de la solución

8.3.1. Características de calidad

Para asegurar la calidad del juego se tomaron en cuenta los siguientes atributos de calidad:

Modificabilidad: Todo el código se escribió respetando el estándar de codificación especificado en el Plan de SQA (Véase completo en el ANEXO B – SQA). El juego fue desarrollado siguiendo el principio de abierto-cerrado y utilizando polimorfismo de manera de poder agregar nuevas funcionalidades fácilmente. Se utilizó el patrón de arquitectura MVC (Modelo Vista Controlador), con archivos de configuración, para facilitar el proceso de agregado de nuevas unidades (Véase el proceso más detallado en el ANEXO F – TDD).

Jugabilidad: Se hizo gran hincapié en este punto. La jugabilidad se definió como la forma en que el jugador interactúa con el videojuego, rubro para el cual no existen principios o patrones que aseguren los buenos resultados. Nos esforzamos en cumplir con los tres puntos definidos anteriormente a través de:

- Menús Intuitivos: Se utilizaron imágenes representativas en lugar de textos para simplificar la interpretación del usuario. Además, la cantidad de interacciones que debe realizar el jugador con el sistema para realizar cualquier acción es menor a 3.
- Comandos Intuitivos: Utilizamos imágenes fácilmente asociables a las acciones que representan (por ejemplo, la imagen de un escudo para representar la acción de cambiar a postura defensiva).
- No linealidad: Lo logramos con la gran cantidad de variables que se involucran en una partida, como lo son las distintas unidades proporcionadas, las combinaciones posibles, y el lugar disponible para colocar las unidades inicialmente.

Robustez: Según los resultados arrojados por las últimas métricas (Véase métricas completas en el ANEXO B – SQA), hemos terminado el desarrollo con 1 caída cada 25 corridas, por lo que podemos decir que hemos cumplido con este atributo.

Performance: Según las métricas tomadas de la última versión del juego a la fecha del 20 de diciembre, probados en una Tablet Toshiba Thrive (Tablet gama media, Android 4.0.4, Procesador 1 GHz y 1 GB de RAM) el tiempo de carga al iniciar la aplicación era de 4 segundos, y al iniciar un round es de 6 segundos. Ambos resultados menores a las cotas definidas en al comienzo. Respecto a los fps, en ningún momento fue un problema, nunca bajo los 30.

8.3.2. Descripción y justificación de la arquitectura seleccionada

Los componentes de nuestra aplicación se compilan en un archivo .apk, para Android. Por otro lado, el servidor no se compila ya que Ruby es un lenguaje interpretado.

A continuación se presenta un diagrama en el cual se puede ver cómo estará distribuida la aplicación. Dos dispositivos móviles tendrán instalada la aplicación, y existirá un servidor en la nube. La comunicación se realizará entre cada dispositivo y el servidor.

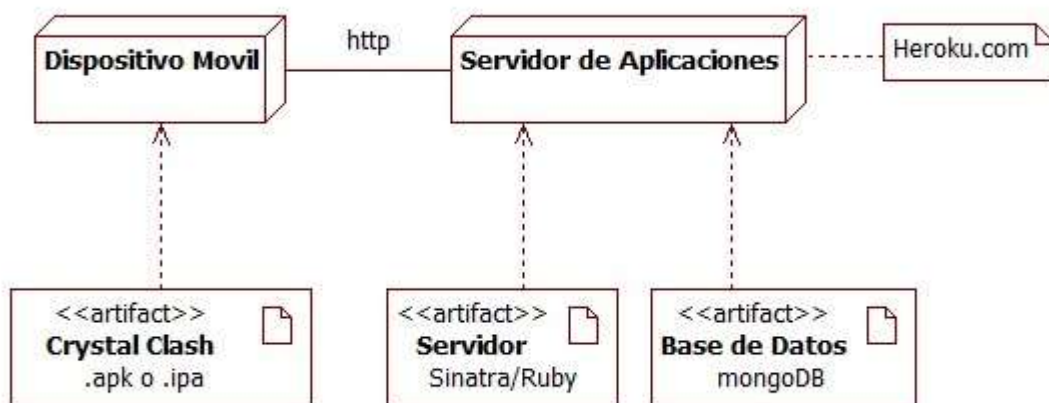


Diagrama 8-1: Diagrama de despliegue

El siguiente diagrama muestra los distintos paquetes y clases a alto nivel, especificando lo más importante de la solución.

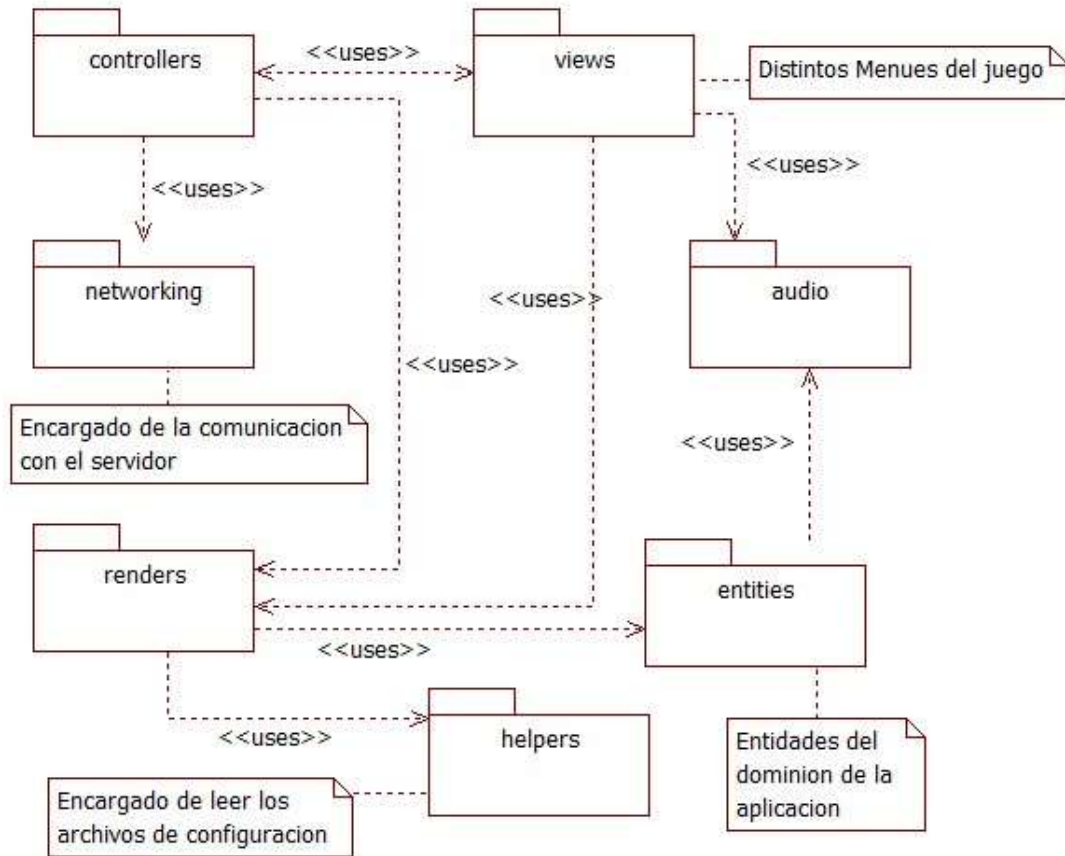


Diagrama 8-2: Diagrama de paquetes

8.3.3. Descripción de los principales mecanismos

A continuación se describen los principales mecanismos que componen el flujo y los distintos bucles en el juego.

Log in

En el diagrama siguiente se muestra el flujo de la conexión entre el dispositivo móvil y el servidor. Se ejemplifica con un método específico, pero todas las conexiones con el servidor se realizan de la misma forma. La aplicación pide datos al servidor y espera a que éste conteste.

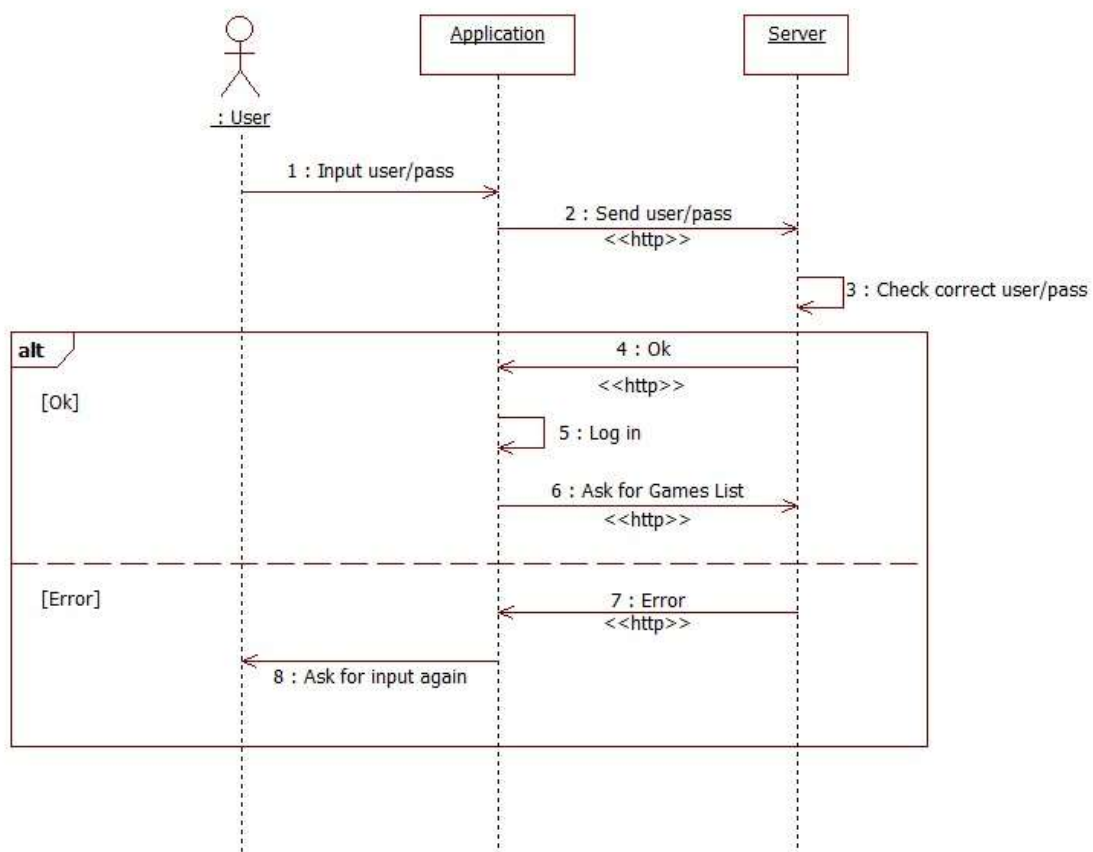


Diagrama 8-3: Diagrama de comportamiento - Log in

Open Game

A continuación se presenta un diagrama mostrando el flujo de la aplicación para abrir un juego determinado partiendo de la lista de juegos.

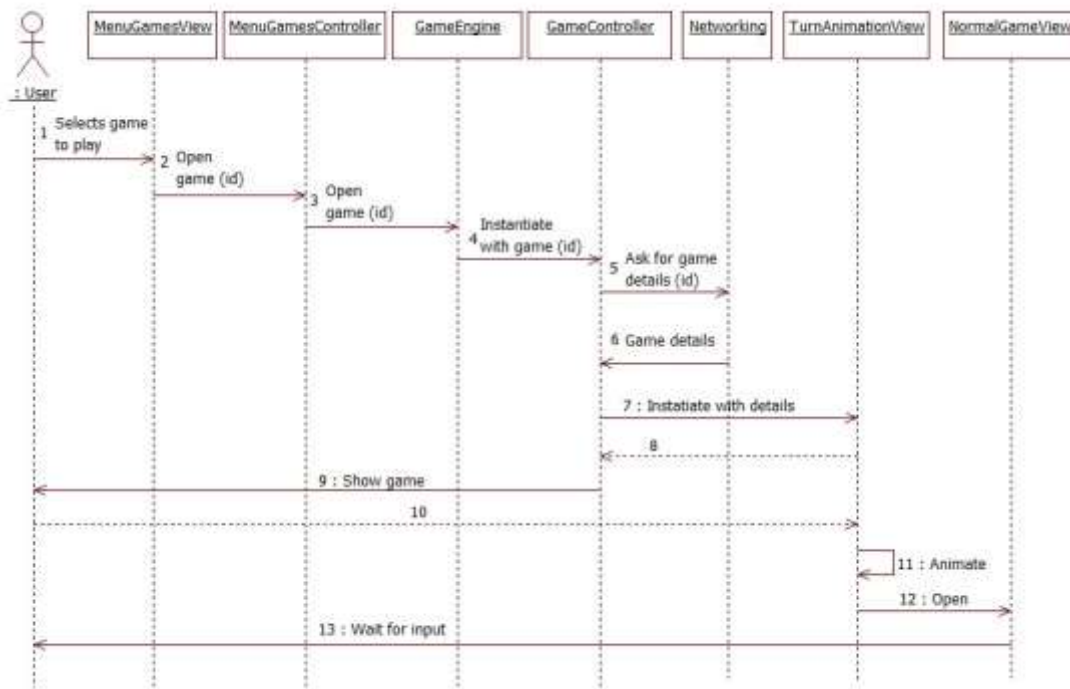


Diagrama 8-4: Diagrama de comportamiento - Open game

Send Turn

En el siguiente diagrama se podrá apreciar el flujo de actividades necesarias para el envío, al servidor, de los datos de un turno realizado por el usuario.

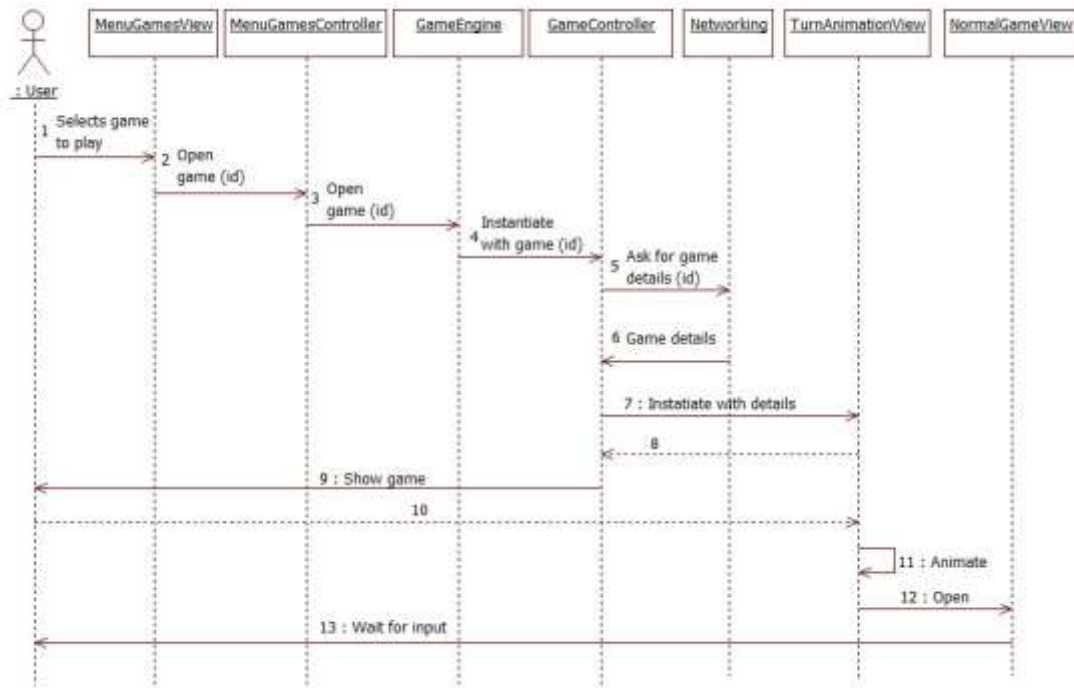


Diagrama 8-5: Diagrama de comportamiento - Send Turn

8.3.4. Restricciones

No existieron restricciones por parte del cliente para desarrollar el videojuego ya que el videojuego es una idea propia del equipo. Sin embargo, de manera de no afectar la performance del producto, fue necesario tener ciertas consideraciones en cuanto a lo que tiene que ver al manejo de recursos.

Manejo de recursos gráficos

Debido a que los recursos de hardware en los dispositivos móviles son mucho más limitados que en una computadora personal, el manejo de los recursos gráficos es una actividad que hay que realizar cuidadosamente. Un mal manejo de esta actividad puede provocar problemas de performance o robustez. Se decidió entonces, utilizar la técnica de atlas de texturas para las animaciones.

Atlas de texturas: La técnica de atlas consiste básicamente en encapsular dentro de una misma imagen, varias imágenes del personaje e ir utilizando partes de dicha imagen según las necesidades.



Imagen 8-1: Secuencia de Imágenes de ejemplo

Para utilizar un atlas para crear una animación se instancia una entidad capaz de mostrar una imagen y, dicha entidad, se encarga de mostrar sucesivamente la imagen correspondiente para cada instante de tiempo. De esta forma, se logra simular un movimiento continuo.

Manejo de recursos de sonido

Los recursos de sonido utilizados están en MP3, ya que se consideró que era el formato con los atributos adecuados para nuestro proyecto, debido a su nivel de compresión. Se utilizaron archivos con una calidad de aproximadamente 192 kbps para la música, y de 128 kbps para los efectos de sonido, de manera de mantener un equilibrio conveniente entre prestaciones y tamaño.

8.4. Estrategia de desarrollo

8.4.1. Plan de versiones

Se creó una versión intermedia demostrable (First Playable), que permitiera exponer el producto para evaluar su calidad y recolectar sugerencias de potenciales usuarios.

Luego, al entrar en la etapa de producción, se decidió que se debería obtener de cada una de las iteraciones una versión demostrable del videojuego.

- En el First Playable se implementaron todas las funcionalidades necesarias para tener una primera versión jugable. Fue muy importante alcanzar esta versión lo antes posible para poder obtener feedback de nuestro asesor técnico y atacar los puntos de mejora en etapas tempranas.
- En el Second Playable se implementaron mejoras en lo que respecta a la experiencia de usuario. Se agregó la música, los efectos de sonido, y un tutorial para aprender los conceptos básicos.
- En el Third Playable hubo un leve cambio de requerimientos y se continuó mejorando la experiencia de usuario. Se cambió el mapa, y se agregaron imágenes de las unidades.

- En el Fourth Playable, se cierra el juego, terminando de arreglar los últimos bugs que surgieron.

Por último, la versión final contempla los cambios surgidos por las pruebas de la etapa de liberación y representa la versión alcanzada del producto.

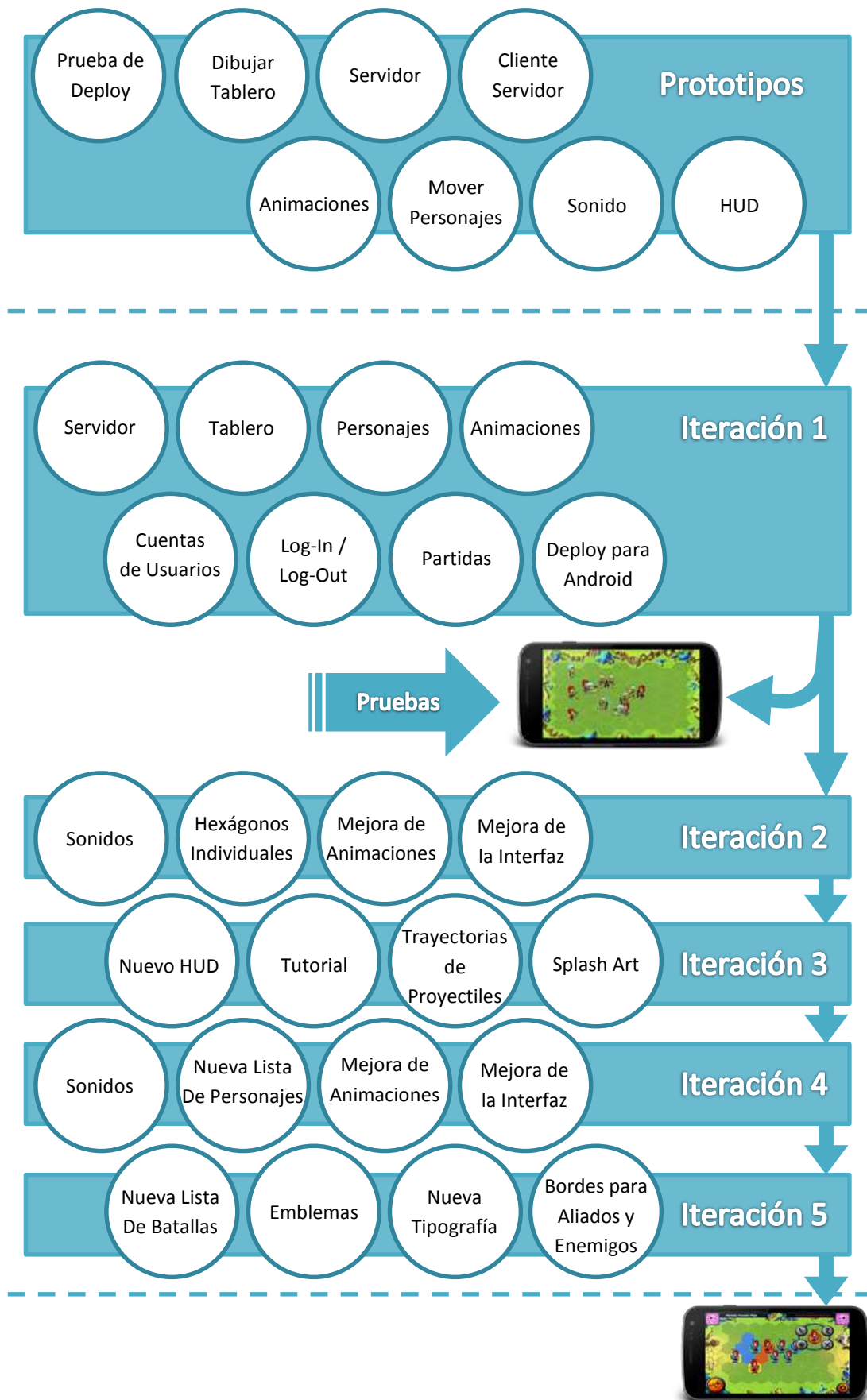


Diagrama 8-6: Plan de Versiones

8.4.2. Forma de división del desarrollo

No hubo una división estricta de las tareas de desarrollo, se detallaban las mismas en Trello, y cada integrante se auto asignaba una al momento de comenzar. Esto ayudó a que cada integrante supiera en cada momento que estaban haciendo los demás, y de esta forma se evitaron problemas debido al trabajo concurrente. Trello es una herramienta de gestión de actividades para equipos accesible vía web que destaca por su filosofía Kanban y su sencillez extrema.

Por otra parte, gracias a la utilización de Github y su sistema de Branches, en caso de que hubiera concurrencia, nos ofrecía una forma sencilla de enfrentarla.

De todas formas, el equipo tenía una gran comunicación, con reuniones semanales.

9. Gestión de la calidad

9.1. Introducción

En este capítulo se describirá el proceso de aseguramiento de calidad aplicado a lo largo del proyecto. La calidad dentro del proyecto fue definida como el cumplimiento de los estándares, criterios y atributos definidos en el Plan de SQA.

9.2. Proceso de aseguramiento de calidad del producto

9.2.1. Proceso de SQA y sus actividades

El objetivo del proceso de aseguramiento de la calidad del videojuego (SQA) consistió en desarrollar actividades que permitieran asegurar la calidad de los procesos y el producto del proyecto. Previo a definir estas actividades se definió un responsable para realizar la planificación, seguimiento y comunicación de dichas tareas; el líder de SQA (LSQA).

El LSQA coordinó e informó al resto de los integrantes del equipo, las diferentes tareas que se debían realizar para el control y aseguramiento de la calidad en las diferentes etapas del proyecto, así como también como las diferentes dificultades presentadas. Estas actividades fueron planificadas y detalladas en profundidad en un documento llamado Plan de SQA (Véase completo en el ANEXO B – SQA).

Las actividades, básicamente, consistieron en revisiones específicas para los distintos elementos de configuración, donde se evaluó el cumplimiento de los estándares y atributos de calidad. Además se validaron según criterios de aceptación, o se realizaron las pruebas correspondientes. Según cada una de las etapas del proyecto, identificamos las siguientes actividades:

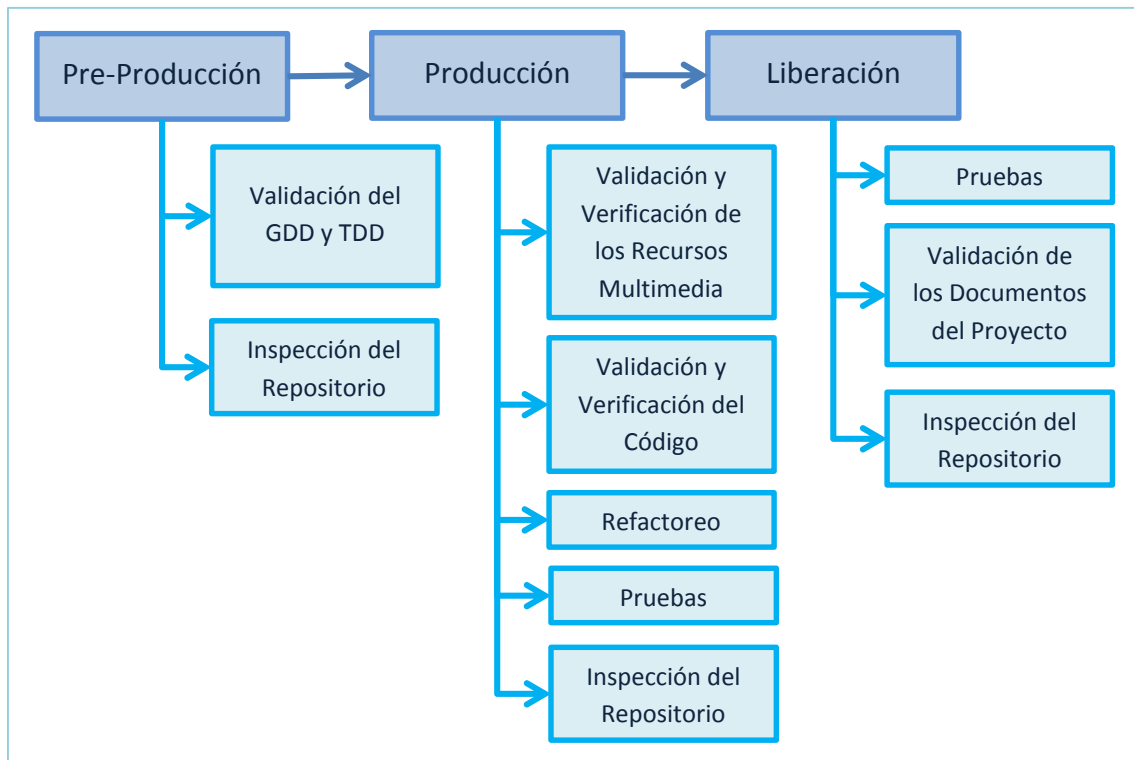


Diagrama 9-1: Actividades de SQA

9.2.2. Definición de estándares

Se definieron estándares que se utilizaron durante el desarrollo del proyecto, con el objetivo de asegurar la calidad de los distintos elementos de la configuración (EC). Los mismos abarcan distintos aspectos, como son estándares para la codificación, documentación, versionado, confección de recursos multimedia y uso de herramientas.

Documentación: Se utilizó como herramienta ofimática a Microsoft Office en sus versiones 2010 para la elaboración de cualquier documento del tipo planilla de texto (en particular la herramienta Microsoft Word). Todo documento debió respetar los estándares requeridos por la Universidad (documento 302 y 303). Con tal objetivo se definió una planilla con el formato requerido, la cual deberá ser utilizada para la creación de los documentos de texto.

Para las revisiones se utilizó la herramienta de presentaciones de Google Drive. Para realizar gráficas y almacenar sus datos se utilizó la herramienta de hojas de cálculo de Google Drive.

Codificación: El equipo de desarrollo utilizó como estándar de codificación el definido por Oracle para Java, [5].

Registro de horas: Con el objetivo de automatizar, simplificar y centralizar el reporte de horas durante el proyecto existió una hoja de cálculo (Spreadsheet) de Google Drive. Dentro de esta planilla, cada integrante tuvo una hoja propia para registrar las horas invertidas en el proyecto, especificando a que tarea correspondían. Se optó por esta modalidad ya que otras herramientas consideradas no resultaban en un balance favorable de costo beneficio, ya que introducían demasiado esfuerzo innecesario para lograr los mismos resultados que se pueden obtener con una herramienta más simple. Al ser un equipo de 4 integrantes resulta más eficiente utilizar la planilla, ya que permite visualizar rápidamente el detalle del trabajo realizado por cada integrante.

Para gestionar las tareas de las iteraciones se utilizó Trello, una herramienta que toma elementos de la metodología Kanban. De esta forma se pudo realizar un seguimiento de cada tarea, obteniendo información de su estado y su recurso asignado. Estas herramientas en conjunto permitieron:

- Respalda la información de la gestión del proyecto en un servidor remoto.
- Gestionar las actividades, manteniendo información sobre su estado.
- Obtención de información para generar reportes de esfuerzo.
- Recopilación de información estadística para mejorar las estimaciones futuras.

Repositorio: Como repositorio para los elementos de configuración no referentes al código fuente, se utilizó Google Drive, mientras que para el repositorio del código fuente se utilizó Github. En el Plan de SCM se encuentra una descripción en detalle sobre las razones por las cuales se utilizarán estos repositorios para el proyecto. (Véase en el Anexo C - Plan de SCM)

Categorías de bugs: Para el correcto informe de los distintos defectos encontrados en el videojuego se utilizó una categorización de bugs que se dividirán en 3 categorías:

- **Mayor:** Implica que la aplicación se cierre en forma abrupta, o que exista pérdida de datos o una fuga de memoria severa.

- **Normal:** Implica una falla en una funcionalidad, constituye un comportamiento no acorde a los requerimientos.
- **Menor:** Implica un problema que no afecta a la jugabilidad, no afecta el cumplimiento de los requerimientos. El caso más común de bugs menores son los defectos cosméticos.

9.2.3. Criterios de aceptación

El alcance de estos criterios se verificó y cuantificó durante las distintas evaluaciones que se llevaron a cabo durante el proyecto.

Definición de los distintos criterios

El siguiente es un listado de los distintos criterios que se utilizarán para los distintos EC.

- **Correcto:** Diremos que un EC será correcto sí y solo sí está libre de defectos. El concepto de defecto depende del tipo de elemento de configuración.
- **No ambiguo:** Diremos que un EC no es ambiguo cuando existe una sola interpretación válida del mismo, en otro caso daría motivo a dudas, incertidumbre o confusión.
- **Comprensible:** Diremos que un EC es comprensible sí y solo sí es entendible por la persona para la cual fue creado.
- **Mantenible:** Diremos que un EC es mantenible si puede ser fácilmente modificado para corregir defectos, mejorar el rendimiento u otros atributos, o adaptarse a un cambio de entorno.
- **Eficiente:** Diremos que un EC es eficiente sí y solo sí no afecta la performance del videojuego y la calidad del mismo no se ve afectada a los ojos del jugador.

A continuación se definen los distintos criterios de aceptación para los diferentes EC.

Criterios para la documentación

Para la aceptación de cualquier documento se definió los siguientes criterios:

- Correcta ortografía
- Correcto: su contenido debe ser el correspondiente al alcance del documento.
- No ambiguo
- Comprensible
- Cumplimiento del formato especificado por la Universidad ORT
- Mantenable

En caso de que un documento no cumpla con la ortografía, su aprobación era parcial. En cualquier otro caso no fue aprobado.

Criterios para la codificación

Es de esperar que cualquier sección de código cumpliera con los siguientes criterios:

- Correcto: debe cumplir con lo que especifica el requerimiento correspondiente.
- Comprensible
- Mantenable
- Cumplimientos del estándar de codificación de Java
- Eficiente: siguiendo las mejores prácticas de codificación.

El incumplimiento de cualquiera de los criterios anteriores implicó la no aceptación del código

Criterios para los recursos del videojuego

Cualquier recurso del videojuego debió cumplir con lo siguiente:

- Correcto: debe cumplir con lo que especifica el requerimiento correspondiente.
- No ambiguo: debe transmitir en forma clara el concepto que representa.
- Eficiente: Buena relación tamaño calidad
- Mantenable

El incumplimiento de cualquiera de los criterios anteriores implicó la no aceptación del recurso.

9.2.4. Validación GDD y TDD

Estos documentos fueron validados en conjunto con el asesor antes de comenzar con la etapa de Producción. Internamente se verificó el cumplimiento de los criterios de aceptación para documentos.

9.2.5. Inspecciones

Documentos: Todo documento se revisó antes de comenzar con la etapa de producción y durante la liberación. Pudiendo existir nuevas revisiones en caso de la aparición de cambios o nuevos requerimientos que impacten en algún documento específico. Se verificó el cumplimiento de los criterios de aceptación para documentos.

Código: Previo a cada impacto en la Branch Master, se validó el apego a los requerimientos establecidos en el GDD, así como su correctitud mediante pruebas realizadas por el desarrollador. Además se inspeccionó el apego a los estándares de codificación establecidos por Oracle para Java.

Recursos: Previo a incorporar un recurso multimedia al videojuego, se validó el apego a los requerimientos establecidos en el GDD. Además se verificó el cumplimiento de los criterios de aceptación definidos para los recursos multimedia. Especialmente se verificó que no afecten desde el punto de vista de la performance en el videojuego en la etapa de Producción.

Repositorio: Inicialmente se inspeccionó que la estructura del repositorio corresponda con la definida en el plan de SCM. Luego, periódicamente se revisó el cumplimiento del Plan de SCM y que cada uno de los elementos del repositorio esté validado por el LSQA.

9.2.6. Pruebas

Existieron diferentes instancias de pruebas al culminar cada iteración durante la etapa de Producción, y por último, otra instancia de pruebas durante la Liberación. Entre las principales pruebas que se realizaron se encuentran:

Pruebas modulares

Es una forma de verificar el correcto funcionamiento de un módulo de código. Esto sirvió para asegurar que cada uno de los módulos funcionara correctamente por separado. A diferencia de las pruebas unitarias convencionales, estas no fueron automatizadas ya que la herramienta no lo permite.

Pruebas de integración

En este tipo de pruebas varios módulos de software fueron combinados y probados como un grupo. Las pruebas de integración fueron realizadas inmediatamente después de integrar dos o más módulos.

Pruebas de sistema

Es donde se verificó cada uno de los requerimientos establecidos en el GDD y TDD.

Pruebas de aceptación

Esta fue una instancia en la que se validaron los requerimientos y se midió el nivel de satisfacción de los usuarios objetivo.

Pruebas alpha y beta

Las pruebas alpha del videojuego se realizaron dentro de un ambiente controlado por el equipo de desarrollo, con usuarios externos al equipo. En cambio las pruebas beta se realizaron en un ambiente no controlado, el videojuego fue distribuido a un grupo reducido de personas ajenas al equipo para que lo utilizaran en sus dispositivos móviles personales.

9.2.7. Registro de errores

Los errores detectados en la sesiones de pruebas, se ingresaron en la planilla de las tareas a realizar en la próxima iteración con el objetivo monitorear y corregir los bugs del videojuego, registrándose para cada uno de estos la información referente al mismo para poder reproducirlo.

Esta se le asigna un responsable, un estado, un nivel de prioridad y una breve descripción.

9.2.8. Refactoreo

Se entiende como refactoreo a la técnica de modificación del código fuente sin cambiar su comportamiento con el fin de mejorar atributos de calidad como mantenibilidad y performance. Esta tarea se realizó especialmente luego de finalizada cada iteración de la Producción, y también existió una instancia de refactoreo durante la Liberación.

9.3. Métricas del producto y proceso

Se definió un conjunto de métricas con el fin de obtener información cuantitativa de los principales atributos de calidad del videojuego y del proceso de elaboración del mismo. Estas sirvieron para obtener feedback continuo y detectar fallas del proceso para tomar acciones correctivas lo antes posible.

Para que un videojuego tenga éxito es indispensable que se defina una serie de métricas para determinar si se han alcanzado los objetivos de calidad, jugabilidad y rendimiento propuestos.

Se realizaron mediciones al terminar cada iteración a lo largo de las etapas del proyecto, de manera de detectar tempranamente obstáculos con los que tuvo que enfrentarse el equipo.

A continuación se describen todas las métricas utilizadas durante el proyecto:

9.3.1. Métricas de Proceso

Esfuerzo real vs esfuerzo previsto

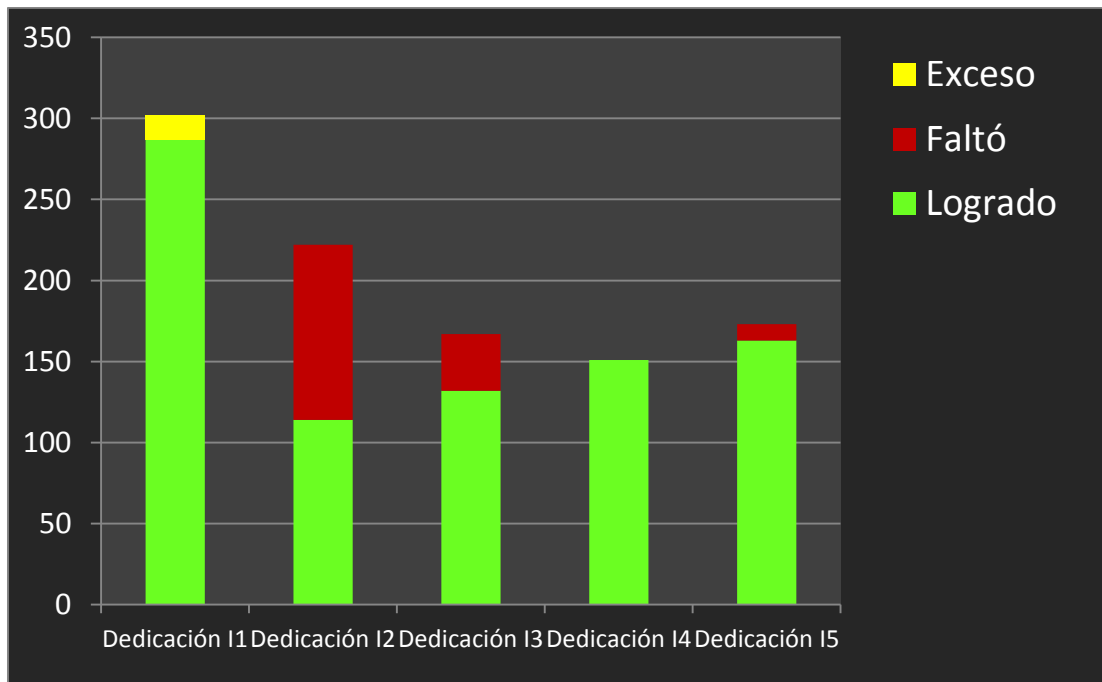
Se tomaron los registros de esfuerzo planificado para cada actividad y se los comparó con el esfuerzo real que tomó cada actividad. Este tipo de métrica le fue especialmente útil para el equipo en las primeras iteraciones, ya que permitió ver que tan acertadas fueron las estimaciones y ver qué tan eficiente era el equipo. De esta forma se pudo mejorar la estimación durante la producción.

Datos relevados y análisis

Se tomaron medidas para todas las iteraciones de la producción, a continuación se podrá ver la comparación entre los estimativos y el esfuerzo real.

	Logrado (h)	Faltó(h)	Exceso(h)	Estimado(h)
Dedicación Iteración 1	287		15	287
Dedicación Iteración 2	114	108		222
Dedicación Iteración 3	132	35		167
Dedicación Iteración 4	151			151
Dedicación Iteración 5	163	10		173
Total	847	153	15	1000

Tabla 9-1: Tabla General Dedicación



Gráfica 9-1: Grafica General Dedicación

Observaciones:

- La estimación fue mejorando a medida que avanzaban las iteraciones, con excepción de la iteración 2 donde fuimos muy optimistas tratando de aumentar el tiempo dedicado por semana.

9.3.2. Métricas de Producto

Funcionalidad: Grado de completitud

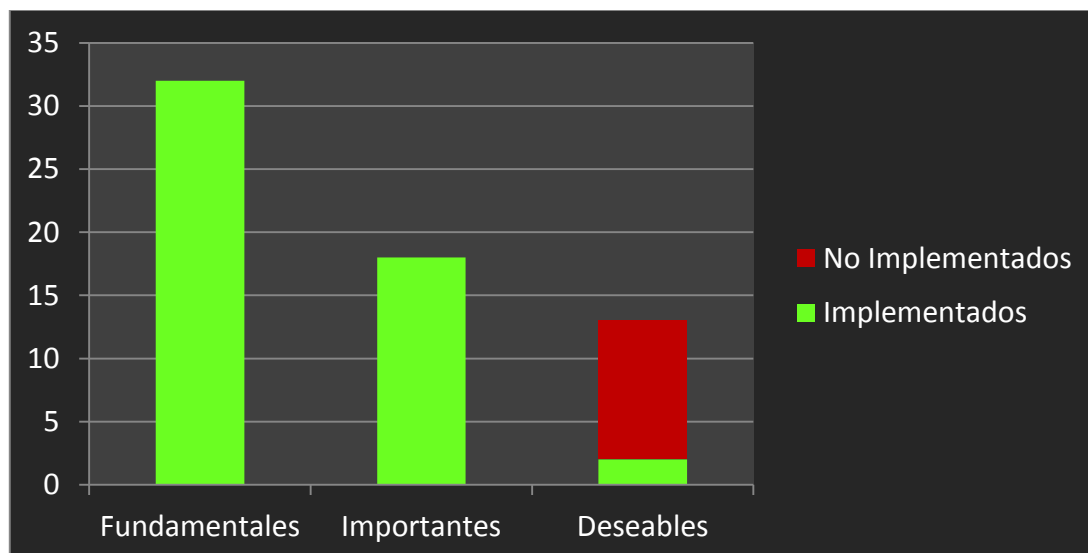
Se calculó el porcentaje de requerimientos completados, esto se realizó para cada una de las categorías de requerimientos: fundamentales, importantes y deseables. Esta métrica se calculó al final de la Producción.

Datos relevados y análisis

Se definieron en total 63 requerimientos funcionales, y se categorizaron según su dificultad de implementación y también según su importancia. A continuación se muestran los datos y las gráficas de los requerimientos.

Dificultad	Definidos	Implementados	No Implementados
Fundamentales	32	32	0
Importantes	18	18	0
Deseables	13	2	11

Tabla 9-2: Tabla de completitud / Dificultad



Gráfica 9-2: Gráfica de cumplimiento de requerimientos agrupados por dificultad

Como se observa, se cumplió con la totalidad de los requerimientos de alta importancia, pero no los requerimientos deseables. Esto se debió a que hubo que acotar el alcance del proyecto y se optó por no realizar los requerimientos deseables.

Robustez: Cantidad de caídas por corridas (Estabilidad)

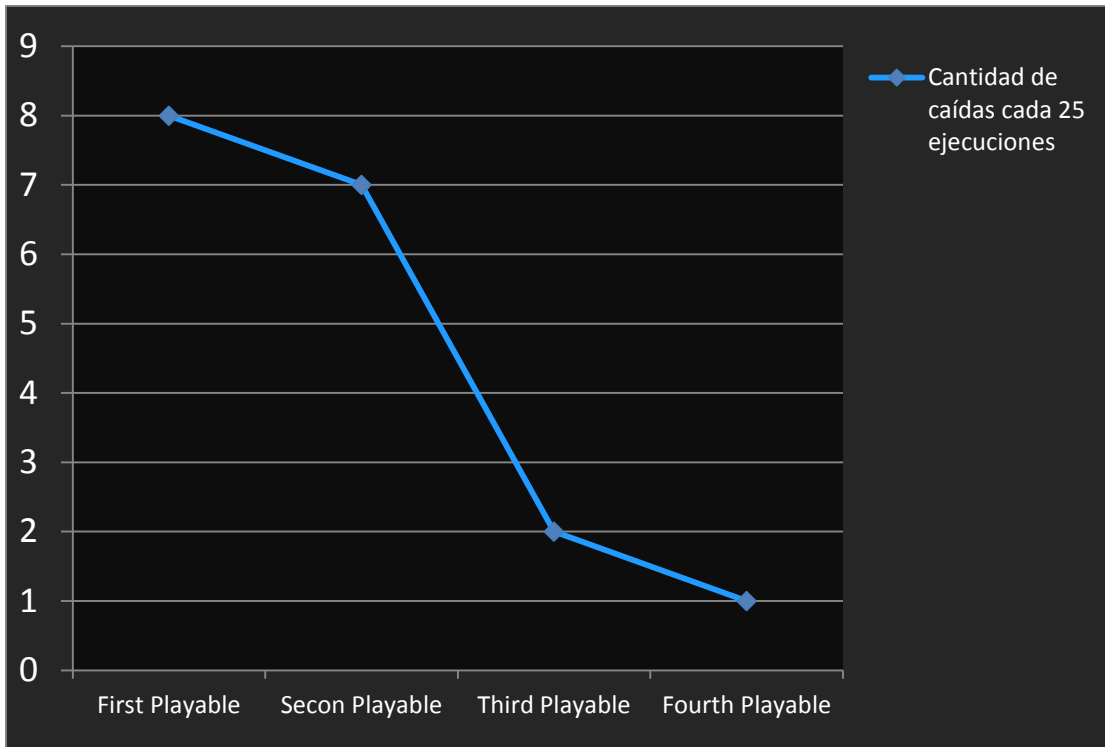
El objetivo de esta métrica fue determinar el número de cierres abruptos en relación a la cantidad de ejecuciones. Era deseable que el valor tienda a cero a medida que se completa el desarrollo, pero consideramos aceptable que en el peor caso el videojuego se cerrara abruptamente debido a un error una de cada veinte veces que se ejecutara. Esta métrica se calculó durante la Producción y la Liberación.

Datos relevados y análisis

Una de los factores más importantes a la hora de jugar a Crystal Clash, es el dispositivo en el cual se juega. En las primeras versiones del juego cuando se corrió en dispositivos de alta gama el juego tendía a no caerse, sin embargo en dispositivos de baja gama se jugaba un turno y la aplicación se caía. Al mejorar las versiones las caídas fueron disminuyendo, sobre todo cuando se corrió el juego en dispositivos de baja gama. Las pruebas realizadas eran para recrear el uso normal de la aplicación, para lo que se jugaban varios turnos en diferentes partidas.

	Cantidad de caídas cada 25 ejecuciones
First Playable	8
Second Playable	7
Third Playable	2
Fourth Playable	1

Tabla 9-3: Tabla de completitud / Importancia



Gráfica 9-3: Gráfica de Errores por corridas

Robustez: Defectos corregidos sobre defectos encontrados

El objetivo es determinar el grado de corrección de los defectos encontrados. Es deseable que este valor tienda a 1.

Para calcular la métrica fue necesario tener la lista de defectos encontrados y la lista de defectos corregidos. Los defectos fueron categorizados en las categorías antes mencionadas (Véase Anexo B - Plan de SQA, Categoría de Bugs) para así lograr un mayor detalle de los mismos. Esta métrica fue calculada durante la Liberación.

Datos relevados y análisis

En los primeros tres Playables la tasa de corrección de errores fue del 100%, ya que en los casos puntuales donde no se llegara a corregir todos los errores encontrados, éstos se trasladaban para corregirse en la siguiente iteración.

Del Fourth Playable se corrigieron el 90% de los defectos encontrados. Del total de éstos, aproximadamente 10% representaban errores de la aplicación, el resto representan mejoras que habían sido aplazadas al priorizar otras actividades. Entre estas mejoras, se encontraban muchos detalles estéticos que encontró el equipo al probarlo o los testers durante las pruebas de la Liberación. Los errores que quedaron sin corregir son completamente cosméticos y no de funcionalidad, serán corregidos en próximas versiones.

Satisfacción: Nivel de satisfacción

Se tenía pensado medir el nivel de satisfacción cada vez que se tenga terminado un nuevo Playable, al final de la Producción y en la Liberación. La métrica tomaría como entrada los resultados de las encuestas realizadas a los jugadores, se calcularía como el promedio de puntuación de las respuestas. Lo deseable es que este valor sea mayor o igual a 70%.

Datos relevados y análisis

Si bien el equipo tenía planeado presentar el producto a los alumnos de una clase de videojuegos de la Universidad ORT, esto no fue posible debido a retrasos en el cronograma y a otras tareas que recibieron mayor prioridad. Sin embargo todos los usuarios que participaron de las pruebas alpha y beta otorgaron un feedback positivo, criticando aquellos aspectos que les resultaban confusos y reconociendo las grandes virtudes del videojuego.

Jugabilidad: Tiempo promedio de aprendizaje

Se deseaba que el videojuego sea auto explicativo, por lo tanto el jugador deberá por sí solo poder aprender a jugar en un tiempo no mayor a 5 minutos. El tutorial ayudaría a reducir el tiempo de aprendizaje, es por esto que esta métrica se midió luego de implementar el tutorial del videojuego.

Datos relevados y análisis

Se realizó un tutorial para el videojuego que explica todas las funcionalidades básicas. Los testers dedicaron un promedio de 4 minutos para realizarlo, por lo tanto se cumplió la métrica satisfactoriamente.

Jugabilidad: Cantidad de turnos promedio por partida

Se decidió medir la cantidad de turnos promedio de las partidas para poder controlar la duración de las mismas. Debido a las características del videojuego, si una partida lleva más de 15 turnos, podría suceder que los jugadores tarden más de dos semanas en terminarla, lo cual es extremadamente extenso. Esta métrica se medirá cada vez que se construyó un nuevo Playable y otra vez durante la Liberación.

Datos relevados y análisis

Al realizar las medidas en el First Playable el promedio de turnos para terminar una partida era de 14. Este se mantuvo en el Second Playable, ya que no se hicieron cambios que afectaran la jugabilidad. Al final de la producción al disminuir el tamaño del tablero, el promedio de turnos por partida bajó a 10. Este resultado nos pareció aceptable y decidimos mantenerlo. En la última medición al final de la liberación, comprobamos que se pudo mantener.

Jugabilidad: Cantidad de Estrategias Dominantes

Es ideal que no exista una estrategia dominante en el videojuego ya que resulta frustrante para los jugadores que tienen que enfrentarse a la misma. Se buscó que para cada estrategia exista otra para contrarrestarla. Esto se midió cada vez que se construya un nuevo Playable y otra vez durante la Liberación.

Datos relevados y análisis

En las primeras versiones del videojuego, el personaje denominado Pesadilla estaba desbalanceado, siendo ella bastante más poderosa que el resto de los personajes. Para la versión final la Pesadilla recibió un cambio para reducir su poder desmesurado, eliminando todo tipo de estrategia dominante y así teniendo una forma de contrarrestar a cada una de las estrategias seleccionadas.

Jugabilidad: Cantidad de Pantallas necesarias para comenzar a Jugar

Se midió la cantidad de pantallas intermedias necesarias para que un jugador pueda comenzar a jugar, partiendo desde el menú principal. Debido a que el videojuego tiene como objetivo atraer a jugadores casuales que dispongan de poco tiempo para jugar, se buscó tener como máximo 5 pantallas para poder comenzar. Esto se midió cada vez que se construyó un nuevo Playable y otra vez durante la Liberación.

Datos relevados y análisis

El número de menús fue descendiendo con el pasar del tiempo. Teniendo 4 menús para llegar a jugar en la primer versión y solamente 2 en la versión final.

Rendimiento: Frame rate

Para que un videojuego tenga animaciones fluidas, el valor promedio de los FPS debe oscilar entre 30 y 60 FPS. Esto se debe a que la ilusión de movimiento se pierde al bajar de 25 FPS, y por encima de 60 FPS el ojo humano ya no detecta la diferencia. Debido a las características del videojuego, con gráficos 2D y con efectos gráficos simples, no necesitó más de 30 FPS. Además tener como tope 30 FPS redujo el procesamiento requerido y permitió que el dispositivo ahorre en batería. Esto se midió cada vez que se construyó un nuevo Playable y otra vez durante la Liberación.

Datos relevados y análisis

El frame rate nunca bajó de los 30 cuadros por segundo.

Rendimiento: Tiempo de carga

Se midió el tiempo transcurrido entre el instante en el que se solicita la apertura de la aplicación y el instante que se muestra el menú principal. Debido a que el videojuego tiene como objetivo atraer a jugadores casuales que dispongan de poco tiempo para jugar, se buscó tener como máximo 5 segundos de carga. Esto se midió cada vez que se construyó un nuevo Playable y otra vez durante la Liberación.

Datos relevados y análisis

El tiempo de carga fue aumentando a medida que el juego fue acumulando contenido, pero nunca superó los 4 segundos en ninguno de los dispositivos de prueba. En las primeras versiones el tiempo de carga era menor a 2 segundos, mientras que en las últimas versiones se encontraba apenas por debajo de los 4 segundos.

9.3.3. Post-Liberación

Al no haber publicado el videojuego aún, al día de hoy no se tienen datos para medir ninguno de los puntos establecidos. De todas formas, se pueden ver las métricas para esa etapa en el Anexo B – Plan de SQA.

9.4. Resultados de las actividades de SQA

En la fase de preproducción se realizaron la mayoría de los planes y documentos necesarios para llevar a cabo las actividades de SQA. En primer lugar, se definió el Plan de SQA (Véase completo Anexo B - Plan de SQA), donde se establecieron las actividades a llevar a cabo para el aseguramiento de calidad.

En dicho documento se definieron los estándares, criterios y atributos que deberían cumplir los distintos elementos de configuración. También se especificaron las diferentes métricas para medir la evolución del trabajo realizado por el equipo, para poder tomar las decisiones adecuadas a lo largo del proyecto.

Durante la etapa de producción, se midieron los atributos fijados en el Plan de SQA y se registraron las métricas y los errores. Se realizó, además, una primera instancia de prueba con diversos voluntarios, donde se probó el First Playable. En esta, se obtuvo puntos de mejora en lo que tiene que ver con la jugabilidad.

Finalmente, se realizó la de evaluación final con los beta testers durante las pruebas en sus dispositivos personales. Si bien no hubo un registro estricto, los beta testers siempre fueron acompañados por un integrante del equipo quien obtuvo las medidas necesarias y feedback.

Tener un Plan de calidad sirvió para que los programadores, que al tener distintos estilos de programación, respetaran un estándar y pudieran realizar las revisiones sin mayor dificultad. Además, al tener las pruebas y métricas definidas, permitió a los desarrolladores realizar pequeñas pruebas rápidas para minimizar el impacto.

10. Gestión de los elementos de configuración

10.1. Introducción

El objetivo de esta sección consiste en explicar el proceso que se definió para controlar cada uno de los elementos de configuración utilizados durante el desarrollo del videojuego y los resultados del mismo.

10.2. Proceso de SCM

Se designó un responsable para el desarrollo de las actividades asociadas al área de apoyo de SCM; el líder de SCM (LSCM). Dentro de sus principales compromisos está presente el desarrollo de un Plan de SCM, gestión del repositorio, respaldos y versionado, de manera de garantizar en todo momento la correcta obtención de cualquier elemento de configuración (EC) durante las distintas etapas del proyecto.

Trabajó junto al LSQA y los delegados en cada área, para la validación de cada elemento de configuración antes de que éste formara parte de la línea base, y luego a modo de integración, la realización de las auditorías internas.

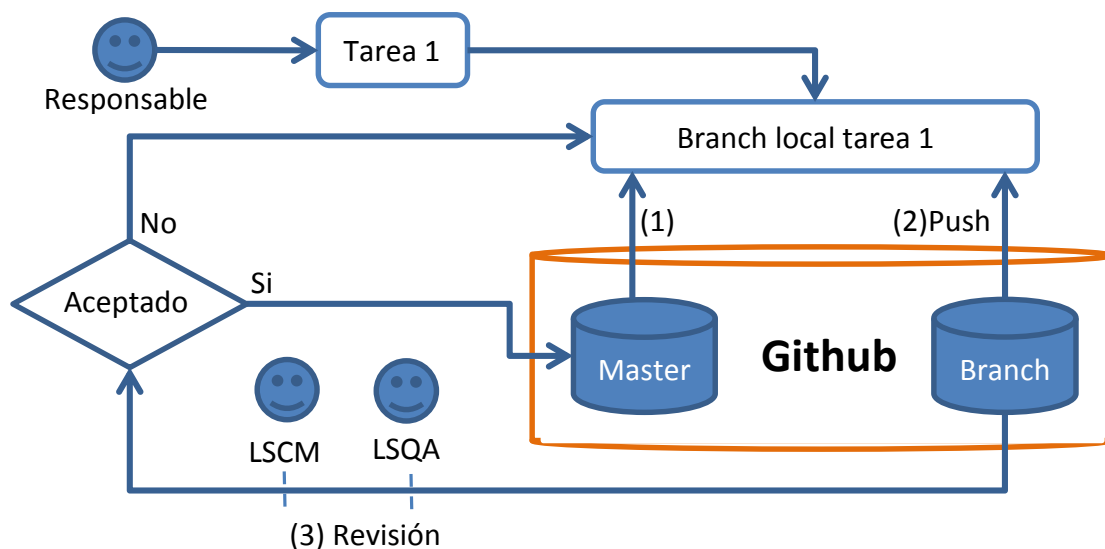


Diagrama 10-1: Ciclo de SCM

10.2.1. Conceptos

- **Branch Master:** Representa la línea base en el sistema Git.
- **Branch:** Copia de la línea base sobre la cual un integrante trabaja sin afectar al resto. Una Branch puede ser enviada al servidor remoto para que otros vean y colaboren en el trabajo.
- **Commit:** Se refiere a la idea de hacer que un conjunto de cambios tentativos se conviertan en permanentes.
- **EC (Elementos de Configuración):** Componentes creados durante el desarrollo. Pueden ser programas (código fuente y ejecutables), recursos (gráficos, audio, video), documentos (técnicos, administrativos y de usuario) y estructura de datos (base de datos y archivos)
- **Línea Base:** Un punto de referencia en el desarrollo del software que queda marcado por la aprobación del elemento de configuración, mediante una revisión técnica formal.
- **Pull Request:** El pedido que un integrante hace al LSCM para integrar un elemento de configuración a la línea base.
- **Push:** Enviar los cambios realizados localmente al servidor de desarrollo.
- **Repositorio development:** Proyecto almacenado en Github.

10.2.2. Tecnologías

- **Git:** Herramienta gratuita de versionado y control de código distribuido.
- **Github:** Alojamiento web gratuito de proyectos Git.
- **Google Drive:** Herramienta gratuita de almacenamiento de archivos en la web, que permite la edición simultánea de documentos por los integrantes del grupo.
- **Trello:** Gestor de actividades para equipos accesible vía web que destaca por su filosofía Kanban y su sencillez extrema.

10.2.3. Código

Asignación de tarea

Cada integrante accedía al Trello, donde podía ver las tareas que se debían completar en la presente iteración, y elegía una a criterio. De esta manera se otorgaba cierta libertad al integrante de lograr el objetivo de la iteración de la mejor manera que vea conveniente.

Development

Nueva funcionalidad / Arreglo de errores: El integrante asignado creaba una nueva Branch local a partir de la Branch master del Repositorio Development, con un nombre nemotécnico a la funcionalidad a implementar.

Cada Branch se debía subir al servidor cuando era creada, así el equipo podría ver en que se estaba trabajando, y debía ser actualizada cada vez que el desarrollador identificaba que había terminado completamente una sub-tarea.

Cuando una funcionalidad o tarea estaba terminada, se debía crear un Pull Request a Master para pedir que la Branch en la que el cambio fue implementado formara parte del código en la línea base. Cada Pull Request fue revisado por el LARQ, LSCM y LSQA, cada uno dejaba comentarios en los puntos donde identifican que no se cumplía con los estándares definidos.

Cuando cada revisor creía que el código cumplía con los estándares, lo aprobaban dando una notificación de aceptación, y el LSCM unía la Branch en Master para que el cambio pasara a formar parte de la línea base y daba por cerrado el Issue que dio inicio al cambio. La funcionalidad fue evaluada desde la perspectiva del usuario por parte de todos los miembros del equipo, y todos tuvieron voto en cuanto a si la funcionalidad se adecuaba a lo planeado. De no ser aprobada la funcionalidad, el miembro del equipo asignado debía corregir lo que haya sido identificado por los revisores, y reiterar el Pull Request para que sea sujeto nuevamente a revisión. El proceso se repitió hasta que la funcionalidad fue incluida en el juego.

10.2.4. Recursos artísticos

Asignación de tarea

El Art Designer se auto asignaba una tarea a la vez, del grupo de tareas elegidas por el equipo para la presente iteración. Estas tareas estaban definidas en la herramienta Trello.

Confección

Nuevo recurso / Cambio de recurso existente: El Art Designer trabajaba en el recurso asignado, y lo almacenaba dentro la carpeta correspondiente de Google Drive.

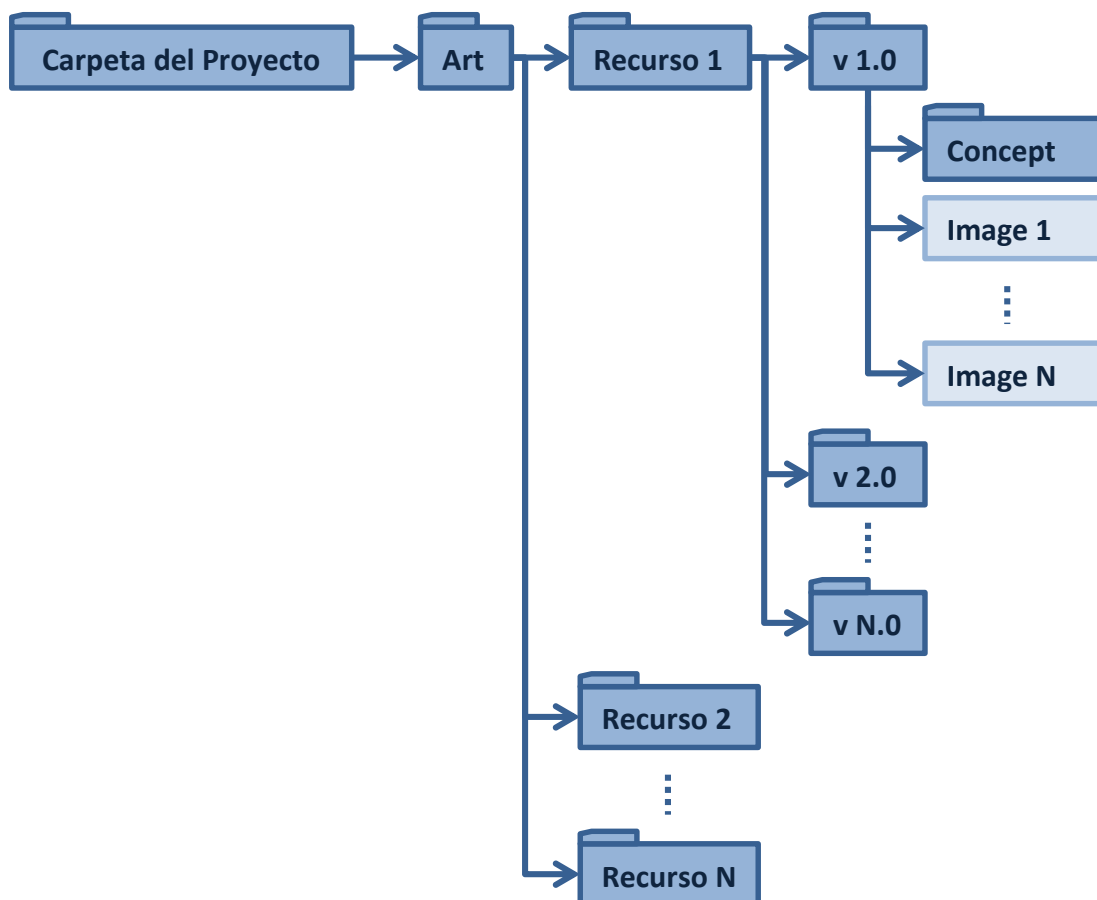


Diagrama 10-2: Estructura del repositorio de Arte

Cuando el recurso estaba finalizado se sometía a una revisión por parte de LSCM y LSQA, cada uno dejaba comentarios en los puntos donde identificaba que no se cumplía con los estándares definidos. Cuando cada revisor creía que el recurso cumplía con los estándares, lo aprobaban dando una notificación de aceptación y el recurso se daba por finalizado.

El recurso fue evaluado desde la perspectiva del usuario por parte de todos los miembros del equipo, y todos tuvieron voto en cuanto a si éste se adecuó a lo planeado. De no ser aprobado el recurso, el miembro del equipo asignado debía corregir lo que había sido identificado por los revisores, y notificar a los miembros del equipo para reiterar la revisión. El proceso se repitió hasta que el recurso fue incluido en el juego.

10.2.5. Documentos

El desarrollo de cada documento fue delegado al responsable del área que ese documento describe. Cada integrante versionó sus documentos por día de trabajo en Google Drive con el nombre: <Nombre> (201x-mm-dd).docx. El LSCM fue responsable de controlar semanalmente que cada documento fuera correctamente versionado, de no ser correcto notificaba al autor para corregir la falta.

10.2.6. Ciclo de Control de Cambios

Al detectar un posible cambio, cada integrante lo ingresará con detalle en la lista de cambios / bugs a resolver en la siguiente iteración. En cada reunión de planificación, las propuestas de cambio serán analizadas por el equipo, se aceptarán o rechazarán, y luego se priorizarán los cambios aceptados. De esta manera, formarán parte de la siguiente semana de trabajo, priorizados junto con las tareas programadas. De esta forma los cambios de mayor impacto serán resueltos antes de agregar nuevas características.

10.3. Repositorio

Los elementos de configuración fueron almacenados en dos servicios remotos. El código fuente del juego fue versionado en Github como un proyecto libre, y el resto de los elementos incluyendo recursos artísticos y documentos fueron versionados de manera privada en Google Drive. De esta manera todo aquel que quiera consultar el código del juego pudo hacerlo a través del sitio Github, pero solo el equipo y el tutor tuvo acceso a los recursos, documentos, y liberaciones del juego.

10.3.1. Estructura

- Repositorio Github para código del juego(nombre clave: fuzzy-adventure)
 - <https://github.com/iuval/fuzzy-adventure>
- Repositorio Github para código del servidor(nombre clave: furry-octo-avenger)
 - <https://github.com/iuval/furry-octo-avenger>
- Repositorio en Google Drive para documentos, arte, y contenido compartido para el desarrollo

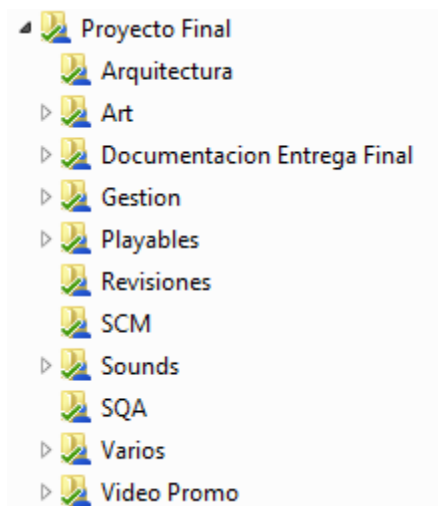


Imagen 10-1: Estructura del repositorio Google Drive

10.3.2. Respaldos

Considerando que la posibilidad de pérdida de los datos en los servicios de Github y Google Drive es remota, todos los integrantes del equipo tienen de cualquier manera la copia de todos los elementos de configuración.

10.4. Resultados de las actividades de SCM

Durante el proyecto no surgieron grandes dificultades en lo que respecta al proceso de SCM, por lo cual podemos señalar que el proceso funcionó correctamente y la selección de herramientas fue adecuada. No hubo necesidad de realizar ajustes en el plan.

Por diversas razones, en muchas instancias no se trabajó en conjunto, debiendo repartirse las tareas. Gracias a la buena comunicación dentro del equipo, se evitó el riesgo de trabajo concurrente y el hecho de perder cambios, tanto en el código fuente así como en el resto de los EC.

Los resultados de las actividades de SCM fueron los siguientes:

- El documento de Plan de SCM, el cual fue desarrollado en la preproducción y especifica todas las actividades que se realizaron para la gestión de los EC.
- Un repositorio web en Github, conteniendo todos los elementos de configuración de software: código, recursos y liberaciones.
- Un repositorio web en Google Drive, conteniendo todos los elementos de configuración que no son software: documentos, recursos, planillas, entre otros.
- Un respaldo de todos los EC (tanto de software, como no) en la computadora personal de cada integrante.

11. Gestión de Proyecto

11.1. Introducción

En este capítulo se resumirá todo lo referido a la gestión del proyecto. Se presentará la planificación de actividades y esfuerzo para alcanzar los objetivos del mismo, para posteriormente contrastar éstos valores contra los resultados reales obtenidos. Además, se comentarán los indicadores del proyecto, cuáles fueron sus lecturas y las decisiones tomadas a partir de sus resultados.

Se realizará un análisis de riesgos del proyecto, el cual constará de la identificación y ponderación de los mismos, así como también de los planes de mitigación y contingencia implementados.

Finalmente se concluirá con una sección que presentará las principales desviaciones del proyecto y sus causas.

11.2. Ciclo de vida

Para la realización de este proyecto se utilizó un ciclo de vida iterativo-incremental debido a que en el desarrollo de un videojuego, se definen al principio los requerimientos y luego con un diseño de alto nivel del mismo se avanza mediante incrementos hasta llegar a la versión final. El proyecto constó de tres grandes etapas: Pre-Producción, Producción y Liberación.

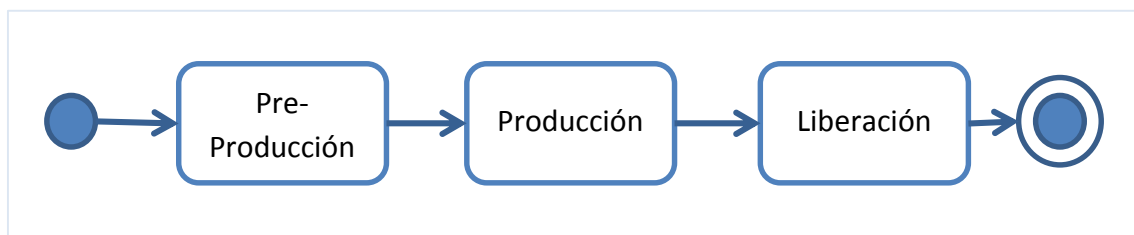


Diagrama 11-1: Macro proceso del proyecto.

11.2.1. Pre-Producción

El proceso de Pre-Producción se compuso de dos subprocesos, definición de requerimientos y diseño de alto nivel del sistema. En esta etapa, el equipo investigó diferentes opciones de tecnología para los diferentes componentes del producto. Se realizaron prototipos para evacuar dudas tecnológicas, y así minimizar riesgos. Los prototipos además sirvieron para definir el alcance del producto con más detalle y con la seguridad de que todo fuera posible de realizar.

Además se elaboró el Game Design Document (GDD) en el que se especificaron los requerimientos funcionales del videojuego, el Technical Design Document (TDD) y los planes necesarios para la ejecución del proyecto.

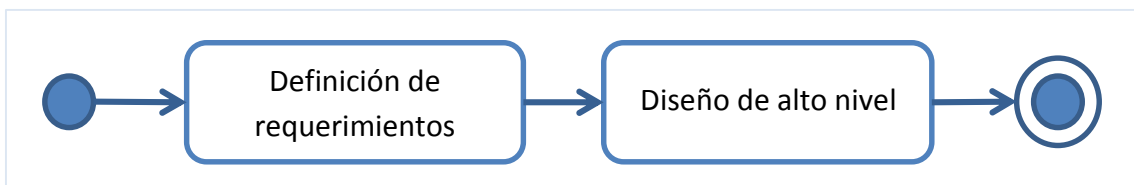


Diagrama 11-2: Subprocesos de la Pre-Producción.

La definición de requerimientos se comportó como un espiral, donde las fases del mismo fueron las siguientes:



Diagrama 11-3: Fases de la definición de los requerimientos.

Durante la fase de investigación se recolectó información acerca de diferentes herramientas y tecnologías que puedan ser de utilidad para la construcción del videojuego. Se investigó sobre los siguientes temas:

- Entornos y herramientas de desarrollo (Engine).
- Librerías o plug-ins para crear efectos de animación.
- Herramientas para la gestión del proyecto.
- Hardware necesario para ejecutar un juego de tales características.

Luego de la investigación, le siguió la fase de prototipación y análisis. Aquí se buscó poner en práctica las diferentes opciones resultantes de la fase anterior. Se construyeron prototipos para determinar el costo beneficio de cada tecnología o herramienta analizada, para luego seleccionar la más adecuada para las necesidades. Con las herramientas y tecnologías ya seleccionadas, y teniendo el conocimiento de qué se puede realizar y qué no, se especificaron nuevos requerimientos que fueron formando el alcance final del juego.

Por último, se validaron los requerimientos con el asesor. De estos, los que fueron aprobados fueron incluidos en el GDD como nuevos requerimientos. Se iteró en este ciclo hasta que se consideró, junto con el asesor y el tutor, que el conjunto de requerimientos seleccionados fueron adecuados y determinaban un alcance viable para un proyecto de fin de carrera de la Universidad. Luego de completar la definición de requerimientos se pasó al diseño de alto nivel del sistema.

Este proceso tuvo como entrada los requerimientos especificados en el proceso anterior y su resultado fue una primera versión del TDD. La culminación del TDD se realizó en la etapa de Producción.

11.2.2. Producción

Esta etapa se compuso de iterar sobre las actividades de diseño detallado, desarrollo y pruebas, como se muestra en el Diagrama 11-4. En esta etapa se procuró la confección de los recursos gráficos, de sonido y la construcción del videojuego. Para la misma se utilizó un ciclo de vida incremental-iterativo como se describe más adelante.

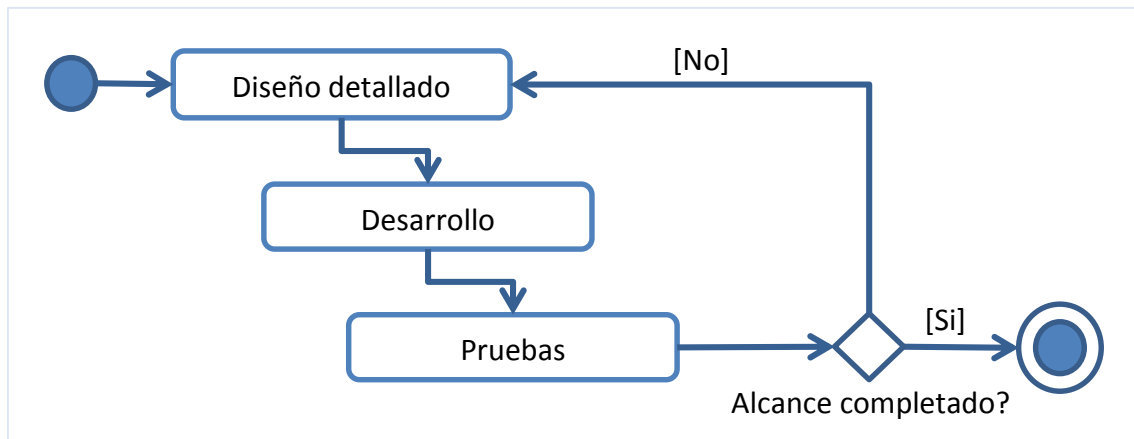


Diagrama 11-4: Actividades de la Producción.

Se realizaron 5 iteraciones. La primera iteración tuvo como resultado el first playable y tuvo una duración de 7 semanas. Esta fue la primera versión jugable del videojuego. Lo que se buscó es tener feedback rápido por parte del asesor y también del público objetivo.

Del feedback obtenido, se extrajeron ideas para mejorar el juego. Con estas recomendaciones, el equipo se reunió con el asesor y se priorizaron los nuevos requerimientos para determinar cuáles de ellos se implementarían y cuáles no.

Las siguientes iteraciones fueron de 4 semanas cada una, se continuó con el desarrollo del videojuego, agregando algunos pequeños nuevos requerimientos. Al final de cada iteración, se obtuvo una nueva versión que fue expuesta al tutor y al asesor. Nuevamente se analizó en busca de obtener nuevas ideas para seguir mejorándolo, evaluando constantemente que requerimientos incorporar y cuales descartar.

En la última iteración, se cerró el desarrollo del videojuego.

Diseño detallado: En esta actividad se realizó un diseño detallado de los requerimientos definidos para el entregable a desarrollar en esa iteración. Este diseño fue la guía a seguir para los desarrolladores en la actividad de desarrollo. También se aprovechó esta actividad para rediseñar soluciones de iteraciones anteriores, en caso de que se encontraran fallas en las pruebas o hubiera cambios en los requerimientos. Al concluir esta actividad se reflejó el diseño resultante en el TDD.

Desarrollo: Durante esta actividad se desarrolló lo especificado en el diseño detallado. Además se realizaron pruebas básicas por parte de los desarrolladores. Estas pruebas fueron a modo de asegurarse que lo desarrollado tuviera una funcionalidad mínima y aceptable como para pasar a la etapa de pruebas.

Pruebas: En esta actividad se ejecutaron pruebas, y se registraron las fallas detectadas. El resultado de las pruebas será utilizado como input para el diseño detallado de la siguiente iteración a modo de corregir los bugs encontrados.

11.2.3. Liberación

En la etapa de Liberación se realizaron las actividades de testing (alpha y beta test), actividades correctivas y actividades del cierre del proyecto, como la documentación.

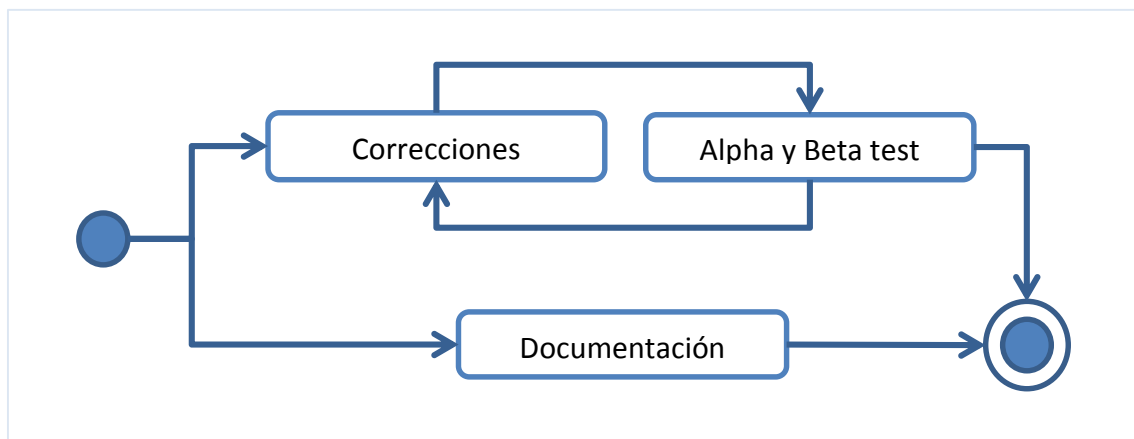


Diagrama 11-5: Actividades de la Liberación.

Las actividades correctivas implicaron solucionar algunos problemas que el videojuego había acarreado luego de la Producción.

Para el beta test se suministró una versión de la aplicación un grupo de personas externas al proyecto para testear el juego en sus dispositivos móviles. Se registraron los bugs encontrados por los testers, se priorizaron y se corrigieron los que tenían más influencia en la jugabilidad y la satisfacción del usuario final.

Se buscó la forma de distribución más adecuada para hacer llegar el producto a los usuarios objetivos. Estos usuarios estuvieron en un entorno conocido y accesible debido a que se necesitaba tener contacto para obtener el feedback correspondiente.

11.3. Cronograma

La etapa de Pre-Producción fue planificada para comenzar el jueves 25 de abril de 2013 y finalizar el miércoles 3 de julio de 2013. Luego, se planificó comenzar con la Producción el jueves 4 de julio de 2013 finalizando el jueves 5 de diciembre de 2013. Por último, la Liberación se planificó para realizarse entre el jueves 5 de diciembre de 2013 y el jueves 20 de febrero del año 2014.

Task Name	24 Mar '13		29 Apr '13		10 Jun '13		22 Jul '13		02 Sep '13		14 Oct '13		25 Nov '13		06 Jan '14		17 Feb '14			
	24	11	29	17	04	22	10	28	15	02	20	08	26	13	01	19	06	24	11	01
1 Crystal Clash	[Barra de progreso que cubre desde el 24 de marzo hasta el 11 de febrero de 2014]																			
2 Preproducción	[Barra de progreso que cubre desde el 29 de abril hasta el 10 de junio de 2013]																			
11 Producción	[Barra de progreso que cubre desde el 4 de julio hasta el 5 de diciembre de 2013]																			
27 Liberación	[Barra de progreso que cubre desde el 5 de diciembre de 2013 hasta el 19 de febrero de 2014]																			

Imagen 11-1: Captura de pantalla de la herramienta Microsoft Project.

Las macro etapas del proyecto se dividieron en iteraciones que agruparon las actividades. Al finalizar cada iteración, se evaluaron las tareas realizadas y se analizaron los resultados, luego se planificaron las actividades para la siguiente iteración, y así sucesivamente.

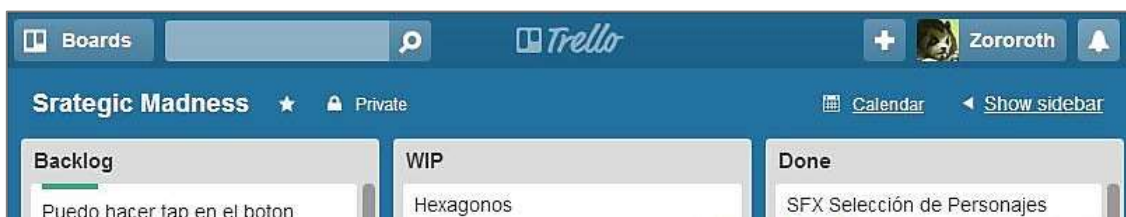


Imagen 11-2: Captura de pantalla de la herramienta Trello.

En la Imagen 11-1 se puede observar la herramienta que el equipo utilizó para mantener un cronograma a grandes rasgos del proyecto. Mientras que en la Imagen 11-2 se muestra como se ve la herramienta Trello, utilizada para gestionar las iteraciones con la metodología Kanban.

En la etapa de la Producción, se dividió el producto en playables, al finalizar cada iteración se produjo un playable, es decir una versión jugable del videojuego. Al terminar una iteración el equipo analizó el trabajo realizado y planificó el trabajo a realizar en la siguiente iteración.

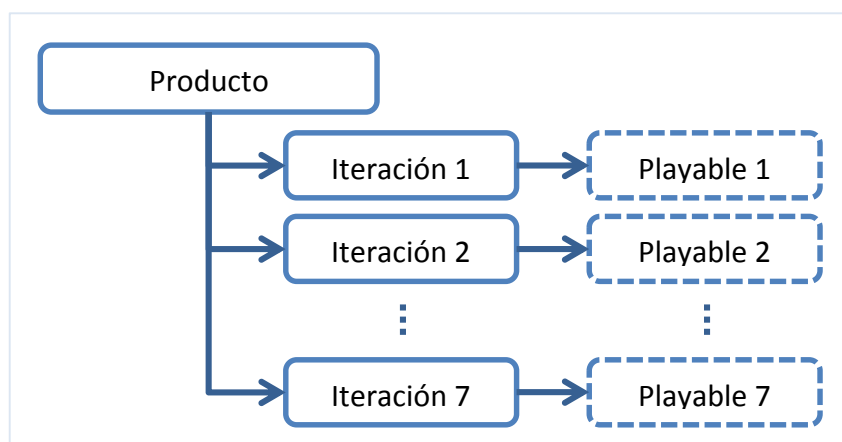


Diagrama 11-6: Planificación realizada por el equipo.

Al trabajar de esta forma, el equipo pudo mejorar cada vez las estimaciones e identificar qué tipo de tareas pudieron ser realizadas en una iteración y cuáles necesitaron ser divididas en actividades de más bajo nivel.

11.3.1. Cronograma Recursos Gráficos

El foco del proyecto no solamente estuvo en la funcionalidad del videojuego sino también en el diseño gráfico del mismo. Es por esto que se buscó que el “look and feel” del juego sea muy atractivo. El equipo mismo fue responsable de generar los recursos gráficos. A medida que avanzó el desarrollo, se le solicitó al art designer los diferentes recursos gráficos para incorporar al videojuego. En los momentos en que el diseñador sufrió algún retraso, no fue un impedimento para el avance del proyecto ya que se utilizaron recursos más simples a modo de sustituto momentáneo para continuar mientras los recursos definitivos no estuvieron disponibles.

11.3.2. Cronograma Recursos Sonido

Los sonidos del juego fueron diseñados y obtenidos por parte del equipo y fueron incluidos en el mismo a medida que fueron necesarios. Las voces de los personajes fueron grabadas por personas externas al equipo, por lo que implicaron una tarea de organización para ir a grabar las mismas, algunas voces fueron grabadas por los integrantes del equipo. En cuanto a la música, también fue producida por personas ajenas al equipo.

11.4. Entregables por Fase

Los productos planificados para cada etapa fueron los siguientes:

11.4.1. Entregables de la Pre-Producción

Pre-Producción

- Game Concept
- GDD
- TDD
- Plan de proyecto
- Plan de SQA
- Prototipos

Los primeros tres documentos de la lista, son los que definen el videojuego en sí. Primero se definió la idea fuerza del videojuego, luego se especificaron los requerimientos funcionales y por último se realizó un documento técnico describiendo los atributos de calidad y la estructura del mismo.

Los otros documentos sirvieron para apoyar la gestión del proyecto, mientras que los prototipos fueron realizados para ayudar al equipo a definir con mayor precisión el alcance. Dentro de las últimas iteraciones se definieron distintos prototipos a realizar.

Iteración 4

- **Prototipo prueba de Deploy:** Este prototipo sirvió para comprobar que se podía hacer deploy del videojuego en Android sin ningún problema.
- **Prototipo dibujar Grilla:** Este prototipo sirvió para comprobar la forma de dibujo que maneja el Game Engine, averiguar rápidamente si era viable para los requerimientos que estábamos manejando.
- **Prototipo de Servidor:** Este prototipo sirvió para evacuar las principales dudas que pudieran surgir en lo que refería al servidor.
- **Prototipo Cliente-Servidor:** Este prototipo sirvió para comprobar que se podía comunicar con el servidor desde dentro del videojuego, y comprobar que el Game Engine lo permitía.

Iteración 5

- **Prototipo prueba de Animaciones:** Este prototipo sirvió para comprobar que se podían implementar las animaciones básicas para los personajes.
- **Prototipo Situar, Mover personajes:** Este prototipo sirvió para comprobar las posibilidades que existen en el Game Engine para implementar estas funcionalidades.
- **Prototipo prueba de Sonidos:** Este prototipo sirvió para comprobar las posibilidades que ofrece el Game Engine para reproducir los sonidos y la música.
- **Prototipo de HUD:** Este prototipo sirvió para comprobar las opciones disponibles para realizar un HUD según como fue conceptualizado en el GDD.

11.4.2. Entregables de la Producción

- Primera versión del juego o first playable: El objetivo de este entregable fue obtener feedback rápido por parte del asesor y por parte de los potenciales usuarios. De esta primera versión del juego se obtuvieron ideas para mejoras en lo referente a la jugabilidad, las cuales fueron analizadas en conjunto con el asesor y el tutor.
- Las siguientes versiones del juego: En base a los resultados obtenidos en la iteración anterior, se trabajó en los nuevos requerimientos para mejorar el juego. Además se continuó con la construcción del juego que estaba planeado.
- La última entrega de la producción fue el videojuego con todas las funcionalidades implementadas.

11.4.3. Entregables de la Liberación

- Versión final del juego: En la etapa de la Liberación se realizaron correcciones de algunos pequeños defectos encontrados.
- Alpha test: Se realizó esta etapa de testing de la cual se obtuvo una lista de defectos encontrados.
- Beta test: Al igual que en el alpha test, se obtuvo una lista de defectos encontrados y oportunidades de mejora.

11.5. Esfuerzo

Para la realización del proyecto, se decidió medir el esfuerzo en base a las horas de las distintas tareas planificadas. De esta forma, el equipo pudo tener un mayor control sobre las actividades del proyecto. Estas mediciones se tomaron al finalizar cada una de las tres etapas del proyecto y se incluyeron en el documento del plan de proyecto a medida que se obtuvieron.

Para definir las tareas de cada iteración, se utilizó la herramienta Trello, y para registrar el esfuerzo, se utilizó un spreadsheet de Google Docs. Dentro de Trello, el equipo ingresó las actividades a realizar, y mantuvo actualizado su estado, utilizando las diferentes columnas de Kanban así como comentarios.

Existió un spreadsheet para cada iteración donde se indicaron las actividades a realizar en dicha iteración, el período en el cual debían desarrollarse y el esfuerzo estimado. En otro spreadsheet, cada integrante debió registrar las horas de esfuerzo, indicando a que tarea correspondieron y una descripción de la misma. Este sistema ayudó a generar registros del esfuerzo real que se le dedicó a las distintas tareas.

11.6. Pre-Producción

Para la etapa de la Pre-Producción se decidió dedicar 10 semanas, 5 iteraciones de 2 semanas cada uno. En esta fase se llevaron adelante actividades de investigación, capacitación, prototipos, reuniones, gestión, diseño y preparación de documentos. En la siguiente tabla se muestra el esfuerzo real versus el esfuerzo estimado de cada una de estas actividades.

Actividades	Esfuerzo Estimado (h)	Esfuerzo Real (h)	Esfuerzo (%)
Investigación	85	52	61
Prototipos	170	128	75
Documentación	85	53	62
Diseño	120	96	80
Gestión y Reuniones	140	132	94
Total	600	461	76

Tabla 11-1: Tabla comparativa de las horas reales versus las horas estimadas de la Pre-Producción.



Gráfica 11-1: Distribución de las horas dedicadas a la etapa de la Pre-Producción.

Como se puede observar en la Gráfica 11-1, las horas dedicadas en esta etapa fueron distribuidas de forma similar en las tres principales actividades de esta etapa (Diseño, Prototipación y Reuniones). De todos modos, se observa una mayor dedicación a la realización de prototipos, actividad principal de la Pre-Producción.

En cuanto a las estimaciones, se puede observar que en las actividades principales la estimación estuvo por encima del 75%, a excepción de los prototipos, donde se pensó que requeriría más horas de las que en realidad llevó.

El total de horas estimadas de la preproducción fue de 15 horas hombre por semana. El equipo en esta etapa pudo dedicarle alrededor de 11 horas hombres por semana, sin embargo se cumplió con todas las actividades planificadas. Cerca del final de la Pre-Producción fue necesario tomar una semana extra, ya que fue necesario dedicar una semana para preparar la primera revisión, por lo tanto la Pre-Producción llevó 11 semanas en lugar de 10.

11.7. Producción

Para la etapa de la Producción se decidió dedicar 23 semanas, 7 iteraciones, siendo la primera de 5 semanas de duración y el resto de 3 semanas cada una. Al llegar al final de esta instancia, el equipo decidió agregar 1 semana más a la Producción. Esto debido a que el equipo notó que requerirían de más tiempo para corregir ciertos bugs persistentes en el juego.

El porqué de este atraso se puede explicar por dos razones. Una de ellas es la dedicación del equipo y la otra los constantes cambios de los requerimientos que surgieron de las diferentes instancias en las que el equipo intercambiaba ideas con el asesor.

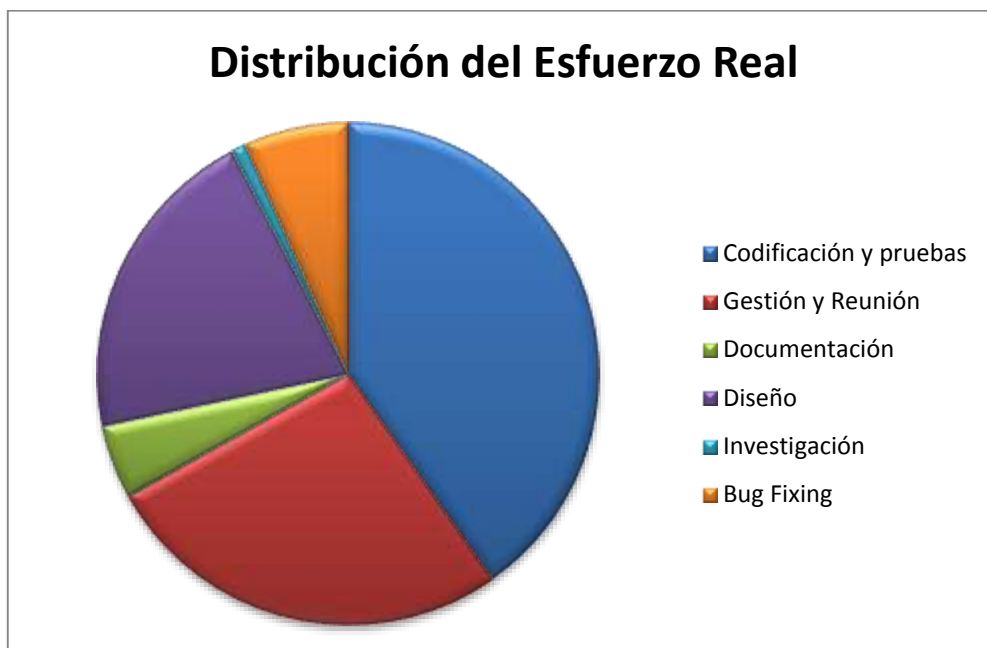
Por el lado de la dedicación, uno de los integrantes del equipo tuvo un viaje de 2 semanas, mientras que otro integrante tuvo tres viajes, uno de 2 semanas y dos de 1 semana de duración. Además un integrante del equipo se encontraba trabajando full-time durante la producción, y otro integrante trabajó part-time. Esto se ve reflejado en los datos recolectados, se desprende de la Tabla 11-2 que cada integrante pudo dedicarle en promedio 9 horas-hombre por semana mientras que la estimación era de 11.

En esta etapa se llevaron adelante actividades de investigación, codificación y testing, diseño, reuniones, gestión, preparación de documentos y corrección de errores.

En la siguiente tabla se muestra el esfuerzo real versus el esfuerzo estimado de cada una de estas actividades:

Actividades	Esfuerzo Estimado (h)	Esfuerzo Real (h)	Esfuerzo (%)
Codificación y pruebas	400	347	87
Gestión y Reunión	230	230	100
Documentación	50	41	82
Diseño	250	178	71
Investigación	20	8	40
Bug Fixing	50	58	116
Total	1000	862	86

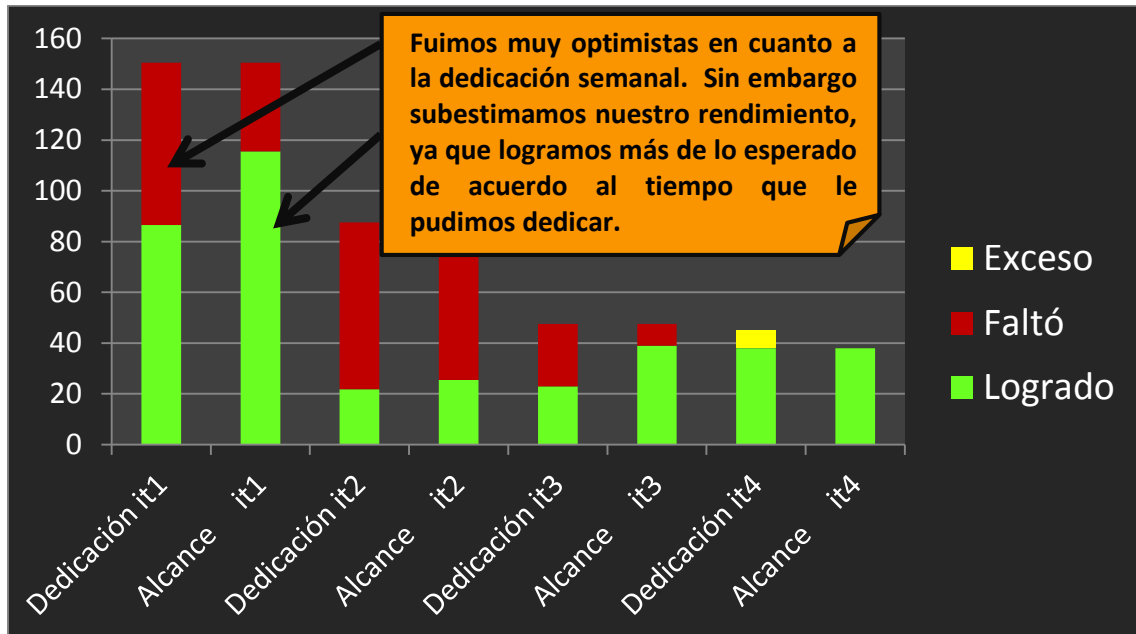
Tabla 11-2: Tabla comparativa de las horas reales versus las horas estimadas de la Producción.



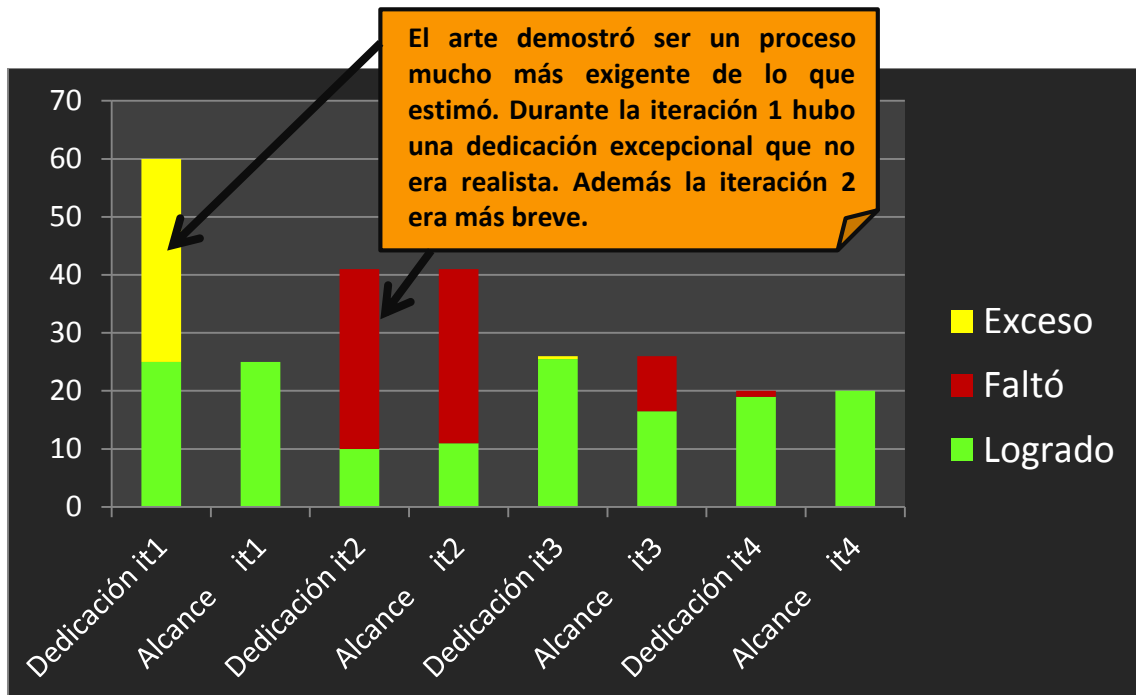
Gráfica 11-2: Distribución de las horas dedicadas a la etapa de la Producción.

Se desprende de la Gráfica 11-2 que la actividad a la cual se le dedicó más horas en la Producción fue la codificación y pruebas, como era de esperar. La actividad de bug-fixing llevó más de lo planificado ya que hubo ciertos errores persistentes al final de la etapa que hicieron que las horas dedicadas a esta actividad fuesen mucho mayores a las planeadas.

A pesar de estos errores, las estimaciones en general fueron más acertadas que en la etapa de la preproducción. En esta etapa se logró un 86% en contraste del 76% de la etapa anterior. Durante la Producción el equipo fue ajustando su capacidad de estimación, minimizando los errores de estimación.



Gráfica 11-3: Lección aprendida en las estimaciones referidas a Codificación y pruebas (al tiempo de culminar la iteración 4).



Gráfica 11-4: Lección aprendida en las estimaciones referidas al Arte (al tiempo de culminar la iteración 4).

Si bien se planificaron 7 iteraciones, el proyecto culminó realizando 5 iteraciones, ya que algunas iteraciones se extendieron un poco más en el tiempo debido retrasos y decisiones del equipo (referentes a fechas clave como reuniones con el asesor), y algunos requerimientos resultaron ser más rápidos de hacer que lo planificado.

Como leer estas gráficas: Cada iteración cuenta con dos barras; dedicación y alcance. El alcance indica la cantidad de trabajo que había para hacer en esa iteración, representa el porcentaje del total que se logró cumplir, indicando en verde lo que se logró y en rojo lo que faltó hacer. La dedicación indica el esfuerzo real puesto para realizar dichas tareas, se indica en verde el tiempo dedicado, si se dedicó menos tiempo del que se planeaba dedicar, entonces aparecerá una barra roja para indicar el tiempo no dedicado. En el caso de que se haya dedicado más tiempo del planeado, se indicara el exceso con una barra amarilla que comienza una vez se ha consumido todo el tiempo originalmente planificado. Como se puede observar en las gráficas 11-3 y 11-4, a medida que se iban realizando las iteraciones, la diferencia entre lo estimado y lo real fue bajando considerablemente, hasta que en la última iteración en la gráfica, la diferencia es casi despreciable.

11.8. Liberación

Como se comentó en la sección 5.2, la Liberación se realizará entre el jueves 5 de diciembre y el jueves 20 de febrero del año 2014. Debido al atraso de una semana en la Pre-Producción, y el retraso de otra semana en la Producción, se decidió modificar el comienzo de la liberación para el jueves 19 de diciembre.

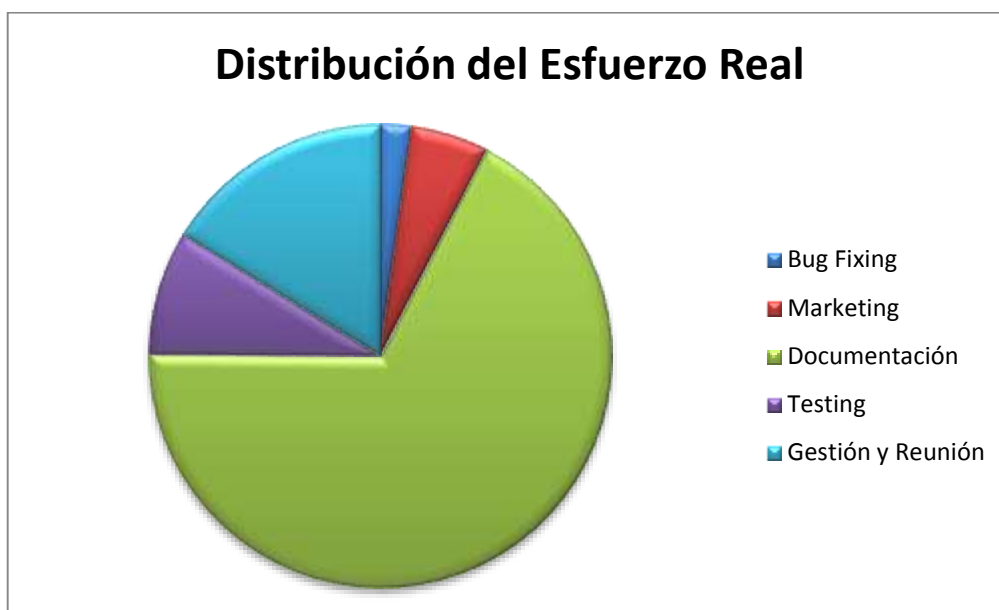
Esto quiere decir que el periodo de resguardo que el equipo había previsto para el final del proyecto fue consumido. Durante las 11 semanas que corresponden al período de Liberación, el equipo se tomara las 2 semanas de vacaciones en lugar de las 3 originalmente planificadas, para poder dejar una semana de resguardo antes de la entrega. Por lo que la nueva fecha de finalización de la Liberación será el jueves 27 de febrero de 2014.

En las 8 semanas de trabajo de la última etapa, el equipo le dedicó 370 horas. Esto se traduce a una dedicación de poco más de 12 horas semanales por integrante mientras que las estimaciones fueron alrededor de 11 horas semanales. En esta etapa se llevaron a cabo actividades de marketing, pruebas, documentación, gestión y reuniones. En la siguiente tabla se muestra el esfuerzo real versus el esfuerzo estimado de cada una de estas actividades:

Actividades	Esfuerzo Estimado (h)	Esfuerzo Real (h)	Esfuerzo (%)
Bug Fixing	10	8	80
Marketing	20	20	100
Documentación	220	250	114
Testing	40	32	80
Gestión y Reunión	60	60	100
Total	350	370	106

Tabla 11-3: Esfuerzo real versus estimado de la Liberación.

A diferencia de las otras dos etapas, la estimación total de la Liberación estuvo subestimada. Sin embargo, el error fue de solamente 20 horas, lo que demuestra la continua mejoría en las estimaciones de las actividades. En las actividades de menor cantidad de horas (bug fixing, marketing, testing) los valores de esfuerzo real se encuentran muy cercanos a las estimaciones. La actividad de documentación estuvo subestimada, pero el error es de muy pocas horas. El esfuerzo real sobre el estimado de esta etapa fue de un 106%, siendo esta etapa la que se estimó mejor.



Gráfica 11-5: Distribución de las horas dedicadas a la etapa de la Producción.

Como se puede observar en la Gráfica 11-5, la actividad a la que se le dedicó mayor tiempo en la liberación fue la documentación. Esta actividad incluyó la culminación de los documentos del proyecto, así como también su corrección y confección de acuerdo a los formatos que exige la Universidad ORT. Las horas de bug fixing, codificación y testing son las relacionadas al alpha y beta test que realizó el equipo. La tarea de marketing refiere a la elaboración de un tráiler para promocionar el videojuego.

En cuanto a las estimaciones, se puede observar que la mejoría continuó en esta etapa. Hay que aclarar también que fue la actividad a la cual se le dedicó menos horas en comparación con las dos anteriores lo que disminuye el margen de error.

11.9. Total del proyecto

La etapa de Pre-Producción comenzó el jueves 25 de abril y finalizó el miércoles 10 de julio. Luego, se comenzó con la Producción el jueves 11 de julio finalizando el jueves 12 de diciembre. Por último, la Liberación se realizó entre el jueves 12 de diciembre y el jueves 6 de marzo del año 2014.

El proyecto comenzó el jueves 25 de abril del año 2013 y terminará el jueves 6 de marzo del año 2014 cuando sea entregado en la Universidad ORT. Para el cálculo del esfuerzo, se tomaron en cuenta las actividades realizadas hasta el jueves 27 de febrero de 2014. Cuando se planificó, se pensaba terminar el 20 de febrero, dejando 2 semanas de holgura antes de la entrega final.

Debido al atraso de una semana en la Pre-Producción y una semana de atraso en la Producción, la Liberación comenzó dos semanas más tarde (consumiendo así las dos semanas de holgura) y se decidió utilizar una de las semanas de las vacaciones del equipo para tener una semana de holgura antes de la entrega.

El total del proyecto fueron 42 semanas, 11 de la Pre-Producción, 23 de la Producción y finalmente 8 semanas de la Liberación.

En la Tabla 11-4 se puede observar el esfuerzo real versus el esfuerzo estimado de las distintas actividades de todo el proyecto. En la Gráfica 11-6 se puede ver la distribución de las horas dedicadas a cada actividad.

Actividades	Esfuerzo Estimado (h)	Esfuerzo Real (h)	Esfuerzo (%)
Investigación	105	60	57
Prototipos	170	128	75
Codificación y pruebas	400	347	86
Diseño	370	274	74
Bug Fixing	60	66	110
Marketing	20	20	100
Documentación	335	344	102
Testing	40	32	80
Gestión y Reunión	430	422	98
Total	1930	1693	87

Tabla 11-4: Esfuerzo real versus esfuerzo estimado de las actividades del proyecto.

En la Tabla anterior se refleja el esfuerzo del equipo en el proyecto: 1693 horas hombre. Siendo la duración del proyecto de 42 semanas, esto refleja que la dedicación semanal fue de 10 horas hombre. Las estimaciones a lo largo del proyecto fueron mejorando etapa a etapa. En la preproducción el porcentaje de las horas reales sobre las horas estimadas fue de un 76 %, en la producción de un 86 % y en la liberación un 106 %. El global de todo el proyecto fue 87 %, valor satisfactorio teniendo en cuenta que el equipo nunca había realizado un proyecto de estas características.



Gráfica 11-6: Distribución de las horas por actividad.

La actividad a la cual se le dedicó más horas a lo largo del proyecto fue la codificación y pruebas. Esta actividad requirió un valor cercano a un 20% de las horas del proyecto. Otro 20 % de las horas fueron dedicadas a la documentación del proyecto, siendo estas horas distribuidas entre la Pre-Producción y la Liberación.

Las horas registradas como diseño de la solución son un 16 % del total del proyecto, ya que incluye las tareas de ilustración y animaciones, así como producción de sonidos y música. Además incluye todo el tiempo dedicado a tomar decisiones sobre los requerimientos del videojuego para mejorar su jugabilidad. Este valor es un tanto engañoso ya que en muchas jornadas de desarrollo el equipo se juntaba a pensar o diseñar las soluciones de los requerimientos a desarrollar y estas horas eran registradas dentro de la categoría de codificación y pruebas. Por lo que algunas horas de diseño están dentro de la actividad de esta categoría.

Las actividades de investigación y prototipos requirieron de 3 % y 7,5 % respectivamente. La mayor parte de estas horas fueron realizadas en la Pre-Producción. Por último, la corrección de errores o bug fixing, tomó un 3,8 % del total del proyecto y estas horas se realizaron durante el alpha y beta testing.

Teniendo en cuenta el total del proyecto podemos observar que la dedicación del equipo se mantuvo constante a lo largo de todas las etapas. En la preproducción fueron 11 horas hombre por semana, en la producción se alcanzó una dedicación de 9 horas semanales (causado por los viajes de dos integrantes del equipo) y finalmente en la liberación se registraron 12 horas hombre por semana.

Como ya se mencionó, el promedio fue de 10 horas, valor que se encuentra dentro de las expectativas que el equipo se fijó a principio de año, entre 10 y 15 horas hombre por semana.

11.10. Análisis de Riesgos

11.10.1. Identificación de Riesgos

Errores de Planificación y Estimación

Por ser la primera vez que el equipo se enfrentó a un proyecto de estas características, era inevitable que se cometieran errores de planificación y estimación. Pero era fundamental que estos errores se abordaran lo antes posible para ir disminuyendo su impacto en el proyecto.

A lo largo de la carrera, los estudiantes son constantemente expuestos al trabajo en conjunto, por lo que no es algo nuevo para los integrantes del proyecto. Sin embargo nunca antes habían formado parte de un equipo con tantas personas, y varios miembros del proyecto nunca antes habían trabajado entre sí, por lo que no se conocían los hábitos de trabajo de cada uno.

Todos estos factores influirían en las estimaciones que se iban a realizar, pudiendo generar diferencias entre el tiempo estimado para realizar una tarea, y el esfuerzo real que eventualmente llevaría a realizarla. Era importante que se registraran los resultados del desempeño lo antes posible para utilizar esta información en futuras estimaciones.

Obtención de recursos de Arte y Audio

Para la construcción de un videojuego serían necesarios recursos multimedia como imágenes 2D, sonidos y música. El proceso de creación de estos recursos era importante para cumplir con las exigentes restricciones de memoria que imponen los dispositivos móviles. Se debía lograr un equilibrio entre la calidad visual y gráfica del juego, y la calidad de los recursos que se utilizarían, ya que cuanto mayor calidad de imágenes y sonidos, se necesitaría más memoria para cargarlos. La idea era optimizar los recursos sin que esto impacte de manera negativa en la parte visual.

El videojuego requeriría una gran cantidad de recursos gráficos y de audio ya que cada personaje tendría su propio conjunto único de imágenes y sonidos. El equipo decidió generar sus propios recursos, ya que un integrante del equipo está capacitado para crear recursos gráficos, y además otro integrante del equipo tiene conocimientos de edición de audio para crear recursos sonoros. De todas maneras ninguno de ellos había trabajado en un proyecto de estas magnitudes con anterioridad, por lo que existía el riesgo de que se cometieran errores de estimación y hasta en el peor de los casos, que no se lograra llegar a tiempo con el producto deseado.

Incertidumbre tecnológica

Dados los objetivos del proyecto, el equipo debía buscar un Game Engine adecuado y también necesitaba implementar un servidor. Si bien un integrante del equipo tenía experiencia previa desarrollando servidores de las características que necesitábamos, el alcance relativamente ambicioso del proyecto podía acarrear riesgos tecnológicos. Sobre todo al no saber si todos los aspectos del videojuego podían ser implementados en el Game Engine que se escogiera para desarrollar el mismo. Este factor podía impactar de varias maneras durante el desarrollo del proyecto.

Uno de los principales problemas serían las estimaciones. Es probable que hubiera que estimar tareas que requirieran conocimientos que el equipo no poseía, entonces se requeriría una pequeña investigación para tener una idea un poco más acertada de cuánto esfuerzo podrían llegar a consumir. Podría suceder que el alcance propuesto en la etapa inicial no se pudiera lograr dentro de los límites de tiempo establecidos en el proyecto, dados los diferentes recursos y el talento humano con el que se contaba. Es importante que el equipo pudiera darse cuenta lo antes posible si se podría cumplir con el alcance propuesto o si se necesitaría que hacer modificaciones al mismo durante el transcurso del proyecto. Para ello eran importantes los datos empíricos que se fueran recolectando durante el proyecto, para poder realizar evaluaciones e ir madurando la habilidad para estimar actividades futuras.

También podía suceder que hubiera actividades que presenten inconvenientes durante su ejecución. Dado que se desconocían cómo funcionan las herramientas internamente, o que simplemente el equipo se estaba interiorizando en áreas de conocimiento nuevas, podía haber fallas cuyos errores sean difíciles de detectar y corregir. En estos casos, se podría terminar consumiendo una cantidad de esfuerzo valiosa que probablemente estuviera planificada para otras actividades. Esto alteraría el cronograma del proyecto, ya que se atrasan actividades, y podría impactar en el alcance final del producto.

Introducción de errores al trabajar concurrentemente

El equipo contaba con cuatro desarrolladores, que trabajarían concurrentemente sobre el mismo proyecto, por lo cual era posible que en algún momento se solapasen y se generaran conflictos.

Se debería utilizar una herramienta adecuada y definir un plan específico para evitar estos conflictos, ya que de presentarse podrían significar horas de esfuerzo invertidas en solucionar el problema. Sin mencionar que durante estas resoluciones de conflictos podría suceder que se pierda trabajo realizado por algún integrante del equipo, lo que significaría más horas hombre perdidas en re trabajo, horas que serían mejor invertidas en crear nuevas funcionalidades y que seguramente estarían destinadas a otra tarea en la planificación.

Era crucial la comunicación entre los integrantes del equipo de desarrollo, así como el seguimiento estricto de las pautas estipuladas para evitar cualquier consecuencia negativa del trabajo concurrente.

11.10.2. Escalas para probabilidad e impacto

Las siguientes tablas muestran las escalas para probabilidad e impacto que se utilizaron en el análisis de riesgos:

Escala para la probabilidad		Escala para el impacto	
Muy alta	1,0	Muy alto	5
Alta	0,8	Alto	4
Moderada	0,6	Moderado	3
Baja	0,3	Bajo	2
Muy baja	0,1	Muy bajo	1

Tabla 11-5: Tabla de escalas para probabilidad e impacto.

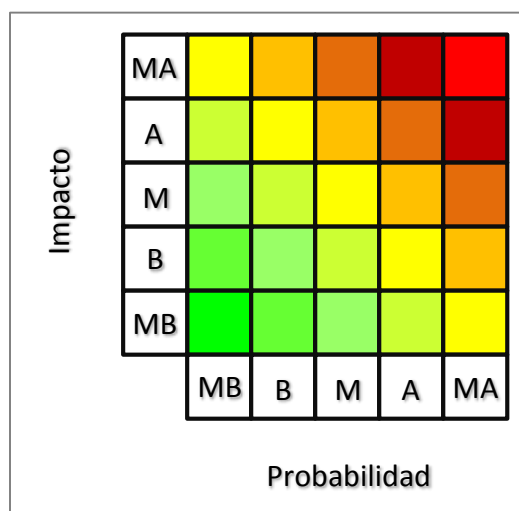


Imagen 11-3: Tabla cromática para la magnitud.

11.10.3. Análisis cualitativo

Para el análisis cualitativo de los riesgos se utilizó una matriz priorizada de probabilidad e impacto de los mismos. La magnitud de los riesgos se calculó como el producto de la probabilidad de ocurrencia y el impacto sobre el proyecto. Según la magnitud de los riesgos se seleccionó una estrategia para tratar el riesgo, y se definió cuáles son los riesgos sobre los cuales habrá que hacer un seguimiento más cercano para evitar consecuencias negativas.

Id de Riesgo	Probabilidad de ocurrencia	Impacto en el proyecto	Área de impacto	Magnitud (0-5)
Errores de Planificación y Estimación	Muy Alta	Muy Alto	Cronograma Alcance	5
Incertidumbre tecnológica	Alta	Muy Alto	Cronograma Alcance	4
Obtención de recursos de Arte y Audio	Alta	Alto	Cronograma	3,2
Introducción de errores al trabajar concurrentemente	Moderada	Alto	Cronograma	2,4

Tabla 11-6: Análisis cualitativo de los riesgos.

Al final de cada iteración se reevaluaron los riesgos, teniendo en cuenta la experiencia adquirida por el equipo en el transcurso del proyecto.

11.10.4. Selección de estrategias

Para cada uno de los riesgos más importantes anteriormente identificados, se seleccionó una de las siguientes estrategias para manejarlo.

Estrategias

- **Evitar** – Eliminar la amenaza, eliminando la causa.
- **Mitigar** –Reducir la probabilidad o el impacto del riesgo.
- **Aceptación**
 - **Activa** – Acción a tomar cuando el riesgo se manifieste.
 - **Pasiva** – No se hará nada.
- **Transferir** – Transferir la responsabilidad a un tercero.

Para cada riesgo mitigado mediante una estrategia, el equipo del proyecto identificó una manera de prevenir el riesgo o reducir su impacto o probabilidad de ocurrencia.

Para cada riesgo, se describió un curso de acción dado el evento de que el riesgo se materialice, de esta forma se reducirá el impacto.

Las estrategias que se tomaron para los diferentes riesgos fueron escogidas dependiendo de la magnitud calculada en la sección anterior, y se muestran en la siguiente tabla:

Riesgo	Estrategia
Errores de Planificación y Estimación	Mitigar
Incertidumbre tecnológica	Mitigar
Obtención de recursos de Arte y Audio	Mitigar
Introducción de errores al trabajar concurrentemente	Mitigar

Tabla 11-7: Estrategia a tomar.

Dadas las características del proyecto, no se puede seleccionar estrategias como evitar o transferir un riesgo, por lo que en general se decidió mitigarlos. A continuación se describen en detalle los planes de mitigación y de contingencia definidos para manejar los riesgos descritos.

11.10.5. Plan de respuesta

Errores de Planificación y Estimación

Estrategia: Mitigar, el proyecto fue dividido en tres macro tareas, Pre-Producción, Producción y Liberación. Se establecieron fechas para comenzar y finalizar las mismas, y las tareas se planificaron y estimaron para poder realizarse dentro de estos límites. Semanalmente se llevó a cabo una reunión con el tutor para evaluar el estado del proyecto, donde se analizaron desvíos y se realizaron correcciones para apegarse al plan. De ser necesario se priorizarían tareas en detrimento de otras, sufriendo modificaciones el plan para lograr cumplir con la fecha límite del proyecto.

Al terminar una iteración, se tomaron datos del desempeño de los integrantes del equipo a modo de información estadística. Esto fue utilizado como input al realizar futuras estimaciones, con el fin de ir mejorando los resultados hasta alcanzar un margen mínimo de error entre las estimaciones y el esfuerzo real que conllevaron para realizarse.

Plan de Contingencia: Aceptación Activa, dada la situación en la que no se lograra cumplir con la fecha límite del cronograma, se extendería la duración de la macro tarea lo necesario para culminar con las tareas imprescindibles en una nueva iteración, se descartarían las tareas restantes que puedan ser ignoradas. Dichas iteraciones extra se planificarían para realizar las tareas de cierre y planificación para la siguiente macro tarea. Esto no sería posible si se atrasara la Liberación, en este caso se deberían sacrificar las tareas pendientes y se ejecutaría el proceso de cierre con el progreso que se haya logrado. Esta situación sería la menos deseable y debería evitarse a toda costa.

Capital humano involucrado: todo el equipo.

Disparador: Sí al culminar la última iteración de una macro tarea aún quedaran tareas imprescindibles sin culminar, se ejecutaría el plan de contingencia.

Control y Comunicación: El estado del cronograma se evaluó en la reunión semanal con el tutor del proyecto.

Incertidumbre tecnológica

Estrategia: Mitigar, Durante la Pre-Producción, se confeccionaron prototipos que abarcaron las más importantes áreas del videojuego. De esta manera se logró evacuar las dudas de forma temprana y se pudo decidir qué acción tomar. Quizás la manera de resolver cierta funcionalidad podría llevar mucho más tiempo del previsto, por lo tanto se debería decidir si se asumirá la nueva duración o sí se modificaría el requerimiento para poder cumplir con los plazos.

Plan de Contingencia: Aceptación Activa, dado el peor escenario, donde el equipo descubriera que determinada funcionalidad no puede realizarse con la tecnología con la que se está trabajando, se debería tomar una decisión. Se modificaría el requerimiento, de manera que se pudiera realizar en tiempo y forma, tratando de no alejarse demasiado del concepto original.

Capital humano involucrado: arquitecto y desarrolladores.

Disparador: Si el equipo se viera atascado ante la incapacidad de implementar una funcionalidad.

Control y Comunicación: En la reunión semanal con el tutor se probaron los prototipos realizados y se actualizaron la lista de prototipos sin realizar, durante la Pre-Producción. Durante la Producción, al manifestarse el riesgo, se comunicaría al equipo la situación y se convocaría una reunión, física o a través de Skype, para tomar una decisión.

Obtención de recursos de Arte y Audio

Estrategia: Mitigar, para cada iteración se definieron los recursos gráficos y sonoros que se necesitaban. Se establecieron fechas para comenzar y finalizar la creación de los mismos, y las tareas se planificaron y estimaron para poder realizarse dentro de estos límites. Semanalmente se llevó a cabo una reunión con el tutor para evaluar el estado del proyecto, allí se revisaron los recursos gráficos y sonoros creados y se analizaron desvíos y se realizaron correcciones para apegarse al plan. De ser necesario se priorizarían recursos en detrimento de otros, el plan podría sufrir modificaciones para lograr cumplir con la fecha límite del proyecto.

Plan de Contingencia: Aceptación Activa, dada la situación en la que no se lograra cumplir con la fecha límite del cronograma, se intentaría transferir la creación de dicho recurso a la siguiente iteración, utilizándose mientras tanto un sustituto de baja calidad para no interrumpir el desarrollo.

Si el recurso no fuese creado al acercarse el fin de la Producción, se buscaría un sustituto definitivo para el recurso faltante, en primer lugar se buscaría obtener en forma gratuita el sonido o el recurso gráfico necesario de internet, bajo licencia creative commons. Si se tratara de un recurso muy específico que no pudiese ser obtenido en forma genérica, se contrataría un recurso externo al proyecto para que cree el recurso grafico o sonoro para nosotros.

Capital humano involucrado: GPRO, art designer y audio designer.

Disparador: Sí al culminar una iteración de la producción aún quedaran recursos de la misma sin crear, se ejecuta el plan de contingencia.

Control y Comunicación: En la reunión semanal con el tutor se revisaron los recursos gráficos y sonoros creados y se actualizó la lista de recursos gráficos y sonoros sin realizar. Al manifestarse el riesgo, se comunicaría al equipo la situación y se convocaría una reunión, física o a través de Skype, para tomar una decisión.

Introducción de errores al trabajar concurrentemente

Estrategia: Mitigar, se utilizó una herramienta de versionado (Git) para manejar la concurrencia, y el encargado de SCM elaboró un plan de SCM para evitar este riesgo. Se crearon nuevas branches a partir de la principal, para cada nueva funcionalidad. Se asignaron recursos diferentes a tareas diferentes, y se buscó que diferentes recursos trabajen en diferentes secciones del sistema. Una vez que se completaba el desarrollo de una funcionalidad, el desarrollador debía realizar el rebase con la branch principal, y luego hacer push de sus cambios. Finalmente hacía un pull-request para informar al encargado de SCM que su branch estaba lista para incorporarse a la branch principal. Todo el equipo de desarrolladores participó de la revisión del código antes de su aprobación, el encargado de SCM se encargó de realizar el último merge que una funcionalidad estaba lista.

Plan de Contingencia: Aceptación Activa, en caso de que se presentara un cambio concurrente, el encargado de SCM decidiría cuál branch se incorporará primero (generalmente el que implique menos cambios, así se evita que alguien tenga que modificar cambios que no realizó). El autor del código que quedó para realizar merge en segundo lugar, sería el responsable de repetir todo el proceso nuevamente para adaptarse al estado de la branch principal de ese momento.

Disparador: Si el encargado de SCM recibiera dos o más pull request que tienen cambios sobre el mismo lugar.

Control y Comunicación: El control fue realizado en cada branch, al realizar un pull request. La comunicación con el encargado de SCM se realizó a través de la herramienta Github.

11.10.6. Reevaluación de riesgos

Se realizó una evaluación cualitativa de riesgos al final de cada iteración de Producción.

Etapa \ Riesgo	Errores de Planificación y Estimación	Incertidumbre tecnológica	Obtención de recursos de Arte y Audio	Introducción de errores al trabajar concurrentemente
Pre-Producción	Muy alta	Alta	Muy alta	Baja
Fin iteración 1	Alta	Moderada	Alta	Baja
Fin iteración 2	Moderada	Baja	Alta	Baja
Fin iteración 3	Baja	Muy Baja	Moderada	Baja
Fin iteración 4	Muy Baja	Muy Baja	Baja	Baja

Tabla 11-8: Reevaluación cualitativa de la probabilidad de los riesgos.

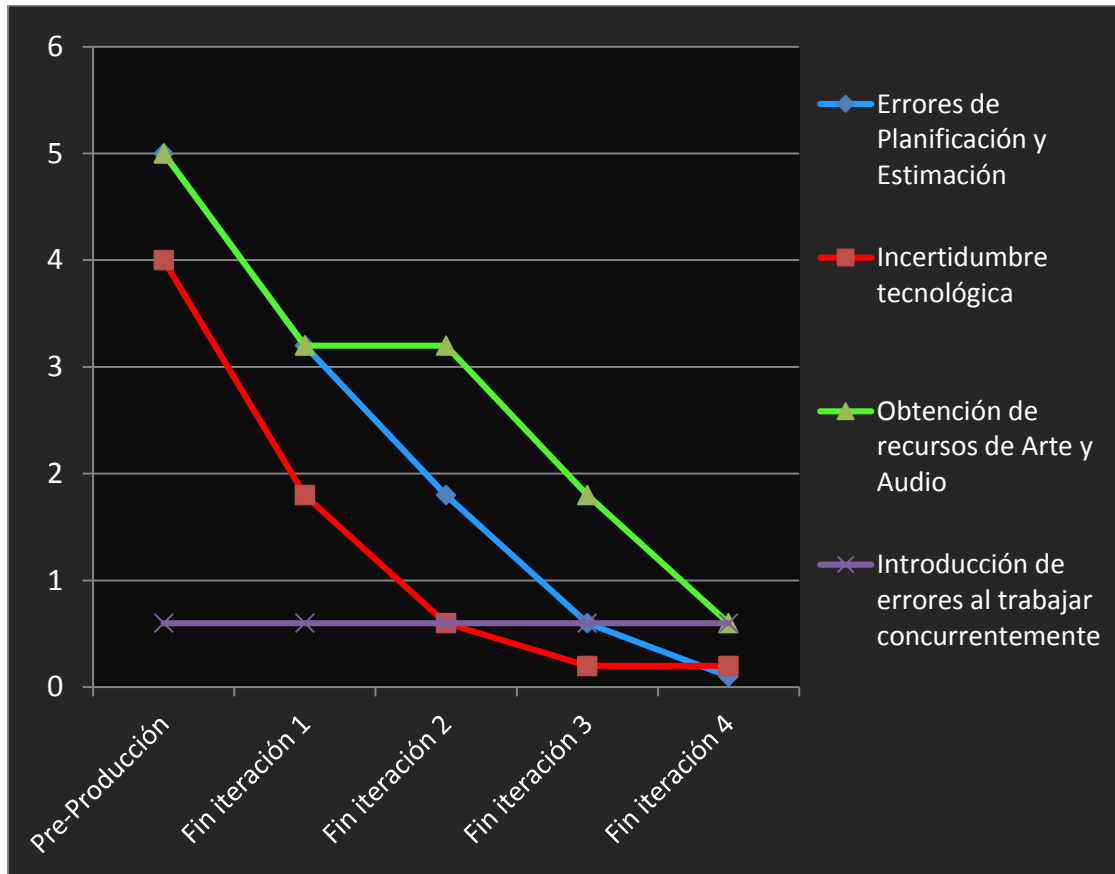
Etapa \ Riesgo	Errores de Planificación y Estimación	Incertidumbre tecnológica	Obtención de recursos de Arte y Audio	Introducción de errores al trabajar concurrentemente
Pre-Producción	Muy alto	Muy alto	Muy alto	Bajo
Fin iteración 1	Alto	Moderado	Alto	Bajo
Fin iteración 2	Moderado	Bajo	Alto	Bajo
Fin iteración 3	Bajo	Bajo	Moderado	Bajo
Fin iteración 4	Muy Bajo	Bajo	Bajo	Bajo

Tabla 11-9: Reevaluación cualitativa del impacto de los riesgos.

Etapa \ Riesgo	Errores de Planificación y Estimación	Incertidumbre tecnológica	Obtención de recursos de Arte y Audio	Introducción de errores al trabajar concurrentemente
Pre-Producción	5	4	5	0,6
Fin iteración 1	3,2	1,8	3,2	0,6
Fin iteración 2	1,8	0,6	3,2	0,6
Fin iteración 3	0,6	0,2	1,8	0,6
Fin iteración 4	0,1	0,2	0,6	0,6

Tabla 11-10: Magnitud de los riesgos.

Como se puede apreciar en las tablas anteriores, la magnitud de los riesgos fue disminuyendo a medida de que el proyecto avanzaba. Esto fue a causa de la experiencia que el equipo iba adquiriendo tanto en lo técnico como en lo administrativo. A continuación se describe para cada riesgo como fue la evolución a lo largo del proyecto.



Gráfica 11-7: Evolución de los riesgos.

Errores de Planificación y Estimación

Si bien el equipo ya tenía experiencia de trabajar en conjunto ya que varios integrantes del equipo ya habían trabajado entre sí durante la carrera, nunca nos habíamos enfrentado a un proyecto de estas magnitudes. Durante la Pre-Producción el equipo tenía mucha incertidumbre respecto al rendimiento de los diferentes recursos del equipo, por lo que el riesgo de cometer errores de planificación y estimación era muy alto.

A partir de la segunda iteración, el equipo fue conociendo su forma de trabajo, y conociendo el rendimiento de los diferentes recursos. Además los desarrolladores fueron obteniendo mayor práctica al utilizar la herramienta de desarrollo, todo esto fue influyendo para disminuir la probabilidad de cometer errores en las estimaciones.

La información obtenida luego de finalizar cada iteración fue fundamental para mejorar la precisión de las estimaciones, usando esta información histórica como referencia fue fundamental para realizar las siguientes estimaciones. Así también se puede ver en las gráficas del esfuerzo dedicado en las diferentes iteraciones, donde el esfuerzo real se fue acercando cada vez más al estimativo.

Es por todo esto que la probabilidad de cometer errores en las estimaciones fue disminuyendo conforme avanzábamos en el proyecto, y el impacto que podrían tener sobre el producto fue disminuyendo también ya que las funcionalidades más esenciales fueron priorizadas para realizarse en las primeras etapas de la producción. Por lo tanto la magnitud del riesgo fue disminuyendo a través de la Producción.

Incertidumbre tecnológica

Durante la Pre-Producción el equipo tenía mucha incertidumbre respecto a las capacidades que ofrecían las herramientas disponibles para desarrollar el videojuego. Fue durante esta etapa que se escogió un Game Engine para utilizar, por lo que fue necesario ir haciendo prototipos de las funcionalidades más importantes cada vez que considerábamos un nuevo Game Engine. En un principio el equipo tenía mucha incertidumbre en las estimaciones y en el alcance que propuso para el producto. Gracias a los prototipos y la capacitación en la etapa de Pre-Producción, se obtuvieron conocimientos muy importantes para tomar decisiones y estimar las tareas de las próximas etapas. Aun así, todavía no había comenzado la construcción del producto y existían incertidumbres tanto en las estimaciones como en el alcance, por lo tanto durante la primera iteración la incertidumbre tecnológica tenía una probabilidad moderada y un impacto moderado en el proyecto. Luego de la primera iteración, que fue una etapa muy fuerte de desarrollo del producto, el equipo se familiarizó con la herramienta y los problemas que se debían resolver.

A partir de la segunda iteración, dado un requerimiento se podía afirmar si era posible de lograr o no con las herramientas con las que se contaba, por lo que el impacto pasó

a ser bajo. En estas etapas de la producción, cualquier funcionalidad esencial ya fue testeada en un prototipo con anterioridad, por lo cual si algún nuevo requerimiento no podía hacerse con la tecnología, seguramente se trataría de un requerimiento no esencial, por lo tanto el impacto sobre el proyecto es bajo. Por esta razón fue que en las últimas etapas del proyecto este riesgo fue de magnitud baja

Obtención de recursos de Arte y Audio

La creación de los recursos sonoros y gráficos fue un riesgo muy importante hasta la tercera iteración, ya que eran un componente esencial del videojuego e iban a ser confeccionados por los integrantes del equipo, con excepción de la música que iba a ser producida por un integrante externo al equipo.

Los recursos gráficos fueron priorizados para realizarse con más anterioridad según la importancia que tuvieran, para evitar así llegar a etapas tardías del proyecto con recursos esenciales todavía en espera de su confección. Esto permitió que el riesgo relacionado a los recursos visuales fuera descendiendo conforme avanzaba el proyecto, en las etapas más tardías del proyecto. Si llegara a suceder que un recurso se encuentra muy retrasado en su producción, seguramente se trate de un recurso poco importante por lo cual no es algo que impacte mucho en el producto.

Lo mismo se hizo con las voces de los personajes, el encargado de confeccionar los sonidos de las voces, debería seleccionar actores de voz para grabar las voces y coordinar sesiones de grabación para obtener los recursos sonoros. Todo esto es un proceso que puede acarrear mucho riesgo, sobre todo por la cantidad de agentes externos al equipo que se ven involucrados. De la misma forma que con los recursos gráficos, se priorizó todo para culminar con la creación de los recursos más importantes al principio de la Producción. La creación de la música del videojuego fue encargada al compositor (agente externo) lo más antes posible, por las mismas razones que se mencionan anteriormente.

Es por esto que al llegar a las últimas iteraciones de la Producción, si se tuviera que obtener un recurso sustituto genérico, obtenido a través de internet bajo licencia creative commons, seguramente serán un recurso de baja importancia y no se va a notar en el conjunto. Es por todo esto que al finalizar la Producción, el riesgo relacionado a los recursos fue de magnitud baja.

Introducción de errores al trabajar concurrentemente

Este riesgo mantuvo una baja magnitud en forma constante durante todo el proyecto debido a que si bien siempre existió la posibilidad de que sucediera, el equipo tomó muchas medidas para minimizar su probabilidad e impacto. El equipo de desarrollo mantuvo constante comunicación durante todo el proceso, y los desarrolladores fueron auto asignándose tareas para que no requirieran que dos personas trabajaran sobre la misma sección a la vez. En el caso de que sucediera que existieran dos pull request sobre el mismo sector del videojuego, entonces se seguiría el proceso estipulado en el plan de SCM para minimizar el impacto sobre el producto y el proyecto. Es por esto que la magnitud del riesgo se mantuvo baja durante toda la producción.

11.11. Desviaciones y Conclusiones

Como se pudo notar en la sección de esfuerzo de este capítulo, existieron desviaciones varias en distintas etapas y tareas del proyecto. En esta sección se analizarán los motivos de dichos desvíos, las acciones tomadas frente a éstos y las conclusiones de la gestión del proyecto.

A continuación se puede ver el cronograma al culminar el proyecto, con las modificaciones resultantes de los desvíos:

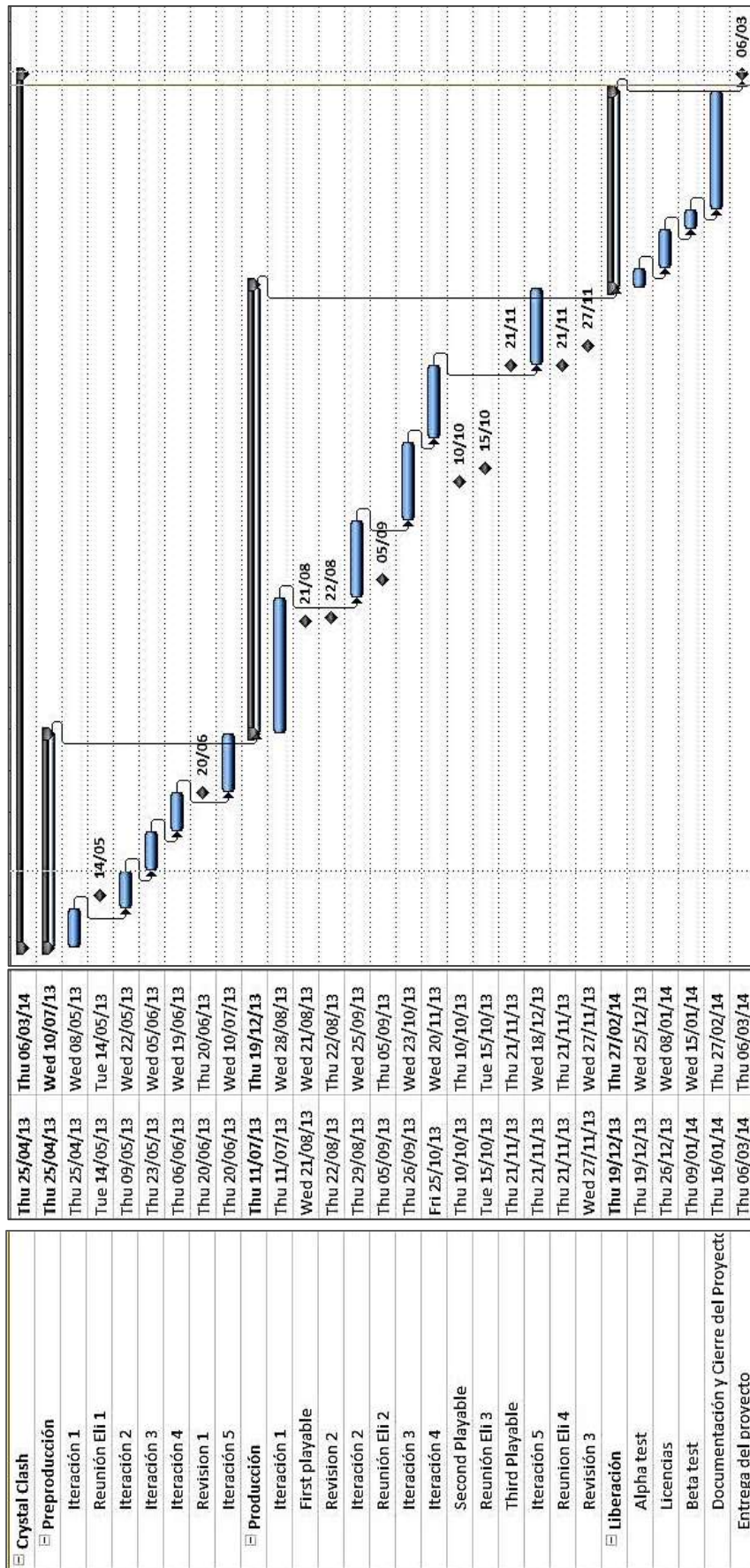


Imagen 11-4: Cronograma del proyecto al finalizar el mismo.

Consideramos importante realizar comentarios respecto a las fechas de los diferentes Playables a lo largo de la Producción. Como se había planeado en un principio, existiría un Playable al final de cada iteración, pero como se aprecia en la Imagen 11-4, solamente existieron 3 Playables intermedios durante la Producción, y uno más al finalizar la misma.

El First Playable no fue liberado al finalizar la primera iteración, sino una semana antes de lo previsto. Esto se debió a que el equipo decidió tener lista una versión jugable del producto para la segunda revisión, ver Imagen 11-4.

El Second Playable no fue liberado al finalizar la segunda iteración, esto se debió a que en la segunda reunión con Eli surgieron nuevos requerimientos que atrasaron el proceso de desarrollo. El Second Playable finalmente fue liberado a mediados de la tercera iteración, ya que necesitábamos tener una versión jugable para la tercera reunión con el asesor, ver Imagen 11-4.

11.11.1. Pre-Producción

Esta etapa del proyecto fue clave, ya que fue aquí donde se decidió que herramienta utilizar para desarrollar el videojuego. Se necesitó hacer prototipos en forma rápida para verificar todas las funcionalidades contra la herramienta seleccionada.

Fue durante la primera revisión que surgieron algunos puntos interesantes respecto a la herramienta que habíamos escogido inicialmente, ya que tenía muchas limitaciones para hacer deploy en Android. Luego de la revisión, el equipo tuvo una reunión para decidir si quedarnos con la herramienta actual o buscar una mejor opción, y decidimos buscar otra herramienta.

Como consecuencia, decidimos tomar una semana extra de Pre-Producción para poder repetir todos los prototipos utilizando LibGDX que fue el nuevo game engine definitivo para el proyecto. Si bien la Producción se vio retrasada, este retraso no se debe a una ineficiencia por parte del equipo, sino que se debe a una decisión que creemos fue para mejor, ya que este nuevo game engine nos permitió hacer deploy de forma muy sencilla. Esto nos permitió probar el videojuego en nuestros dispositivos móviles.

11.11.2. Producción

Desarrollo

El desarrollo del videojuego fue realizado dentro de los tiempos estimados, y logramos cumplir con todas las funcionalidades esenciales que nos planteamos al principio del proyecto.

Sin embargo decidimos tomar una semana extra de Producción ya que quedaban algunos retoques que deseábamos incluir en el videojuego. Estos últimos requerimientos surgieron luego de analizar las últimas versiones jugables y decidimos que eran importantes para mejorar la experiencia de juego.

Vale la pena realizar ciertas aclaraciones respecto a las primeras dos iteraciones de la Producción en lo que refiere al desarrollo, ver Gráfica 11-3. En la primera iteración se estimaron 150 horas de desarrollo a lo largo de 5 semanas, lo que implicaría 30 horas semanales distribuidas entre 4 desarrolladores.

La realidad fue que el Art Designer se dedicó exclusivamente a la confección de los recursos gráficos, y otro integrante del equipo tuvo un viaje inesperado fuera del país, lo que resultó en que el equipo de desarrollo se redujera a 2 integrantes.

Esto resultó en que finalmente se terminaron dedicando 90 horas de desarrollo a lo largo de 7 semanas, pero únicamente con 2 desarrollares. Cabe destacar que el promedio de dedicación semanal por desarrollador fue apenas una hora menos del originalmente estimado, la razón del desvío se debió exclusivamente a la reducción del equipo de desarrollo.

Al realizar la estimación para la segunda iteración, el equipo de desarrollo sintió que estaba capacitado para dedicar más de 7,5 horas semanales de desarrollo por persona, por lo que se estimó con la idea de lograr dedicar en 4 semanas lo mismo que se había logrado dedicar en las 7 semanas de la iteración anterior. Esto significaba duplicar la dedicación semanal por desarrollador, ya que el equipo de desarrolladores seguiría siendo de 2 personas.

Se decidió que el equipo de desarrollo se mantuviera en 2 personas ya que el costo de capacitar a los otros dos integrantes era mayor al beneficio obtenido, además resultaba más conveniente que el Art Designer continuara dedicándose exclusivamente a la creación del arte del videojuego. De la misma forma el Audio Designer dedicó su tiempo exclusivamente a la planificación y confección de los recursos de sonido.

Como se puede ver en la Gráfica 11-3, el objetivo de duplicar el esfuerzo en desarrollo no fue alcanzado durante la segunda iteración. Esto se debió a que el equipo tuvo que invertir horas de esfuerzo en el diseño del videojuego, ya que luego de la segunda reunión con Eli el equipo decidió realizar cambios sustanciales en el diseño del mapa. Luego, en el resto de las iteraciones de la Producción se puede ver que el equipo fue realizando estimaciones cada vez más precisas.

Recursos Gráficos

La creación del arte del videojuego fue realizada dentro de los tiempos estimados, y no generó retrasos en el calendario, pero sí requirió más esfuerzo del estimado en la primera iteración. Esto se debe a que si bien el diseñador estaba acostumbrado a crear recursos gráficos del estilo, nunca se había enfrentado a un proyecto de estas características, y por naturaleza se cometieron errores en las estimaciones.

Hay que tener en cuenta que la primera iteración sucedieron dos cosas simultáneamente. En primer lugar, debido a retrasos causados por malas estimaciones, la iteración se extendió dos semanas, terminando en un total de 7. Además, el diseñador gráfico logró una dedicación semanal mayor a la estimada. Basándose en los resultados de esta iteración, se realizaron las estimaciones para la segunda, siguiendo el mismo índice de dedicación. Luego resultó ser que en la primera iteración el diseñador había realizado una dedicación irreal, y no se pudo continuar con la misma.

Además luego de una de las reuniones con el asesor, se decidió incluir arte más detallado de los personajes dentro del juego, para generar un vínculo más estrecho entre el jugador y los mismos. Estos recursos gráficos requirieron mayor dedicación además de que no estaban previstas en el plan inicial.

12. Conclusiones

12.1. Conclusiones generales

Al comienzo del proyecto, principios de 2013, cuando el equipo discutía las posibles ideas para desarrollar, se planteó lo siguiente: debe ser algo que nos mantenga motivados, y que el proyecto no se convierta simplemente en un último requisito para poder graduarse. A partir de esto, surgió la idea de desarrollar un videojuego. Luego, el equipo decidió que debía enfocarse en los dispositivos móviles ya que era el mercado que venía en mayor crecimiento y con tendencia a continuar creciendo. Además, esto resultaría en un desafío para el equipo ya que no tenía experiencia en el desarrollo de videojuegos, pero estaba dispuesto a aprender para poder publicar un juego propio en las tiendas de aplicaciones.

Gracias a que en el equipo había un integrante con capacidades artísticas, y uno con grandes conocimientos en el área de la música, el equipo decidió que quería hacerse cargo del todo el producto. Esto quiere decir, el equipo sería el responsable tanto del desarrollo del videojuego así como de la creación de los recursos gráficos y sonoros necesarios. Estamos muy contentos y orgullosos de poder decir que logramos cumplir con este objetivo, a pesar del desafío que esto implicó.

Vale la pena comentar que cuando se comenzó con el proyecto, las ideas que tenía pensadas el equipo sobrepasaban el alcance del proyecto. Gracias a una temprana reunión con nuestro asesor, quien nos hizo darnos cuenta de esto, el equipo tuvo que priorizar las características del juego, dejando de lado algunas, para que entrara dentro de los límites de un proyecto académico. La idea del equipo es poder continuar con el juego luego de la entrega, agregando las características que, lamentablemente, quedaron afuera para esta versión.

Nos gustaría destacar como el equipo enfrento los problemas que se le presentaron, respondiendo rápidamente al cambio, pudiendo salir adelante. El principal problema que tuvo que enfrentar fue el cambio de Game Engine a pocas semanas de comenzar la Producción.

Por último, si bien el equipo aún no ha publicado el juego en la tienda de aplicaciones, es algo que seguro va realizar en los próximos meses. Además, se pudo crear un tráiler para promocionar el juego, también una producción original del equipo.

12.2. Productos resultantes

Al finalizar el proyecto, se obtuvo un videojuego 2D, del género Simultaneous Turn-Based Tactics para Android, según lo especificado en los documentos Game Design Document y Technical Design Document. Se logró implementar todos los requerimientos fundamentales e importantes y varios de los deseables, quedando algunos de estos aún por realizar. Si bien el equipo tenía como idea que el juego estuviera disponible para iOS también, por un tema de facilidad del proceso de prueba se generó solo la aplicación para Android. El equipo quiere terminar de implementar los requerimientos deseables que quedaron fuera antes de entrar en el proceso de despliegue en iOS.

También surgieron como productos del proyecto los distintos documentos que el equipo tuvo que escribir durante este proceso, tanto los de la ingeniería tradicional como los de la industria de los videojuegos. Estos documentos incluyen la definición de las tareas del área de gestión, ingeniería de requerimientos, diseño de arquitectura, desarrollo y calidad.

12.3. Logros obtenidos

Se logró completar el proceso de desarrollo de videojuego desde el comienzo, donde se pensó la idea del mismo y se lo especificó en un documento. A partir de este, se pudo desarrollar el videojuego de forma exitosa, logrando implementar casi en su totalidad los requerimientos.

Por otra parte, podemos contar con un logro la experiencia obtenida en lo que refiere al desarrollo de videojuegos, lo que abre nuevas puertas a los integrantes del equipo.

12.4. Problemas encontrados

Uno de los problemas más importantes con los que se tuvo que enfrentar el equipo de desarrollo fue el cambio de Game Engine. En un principio, luego de investigar y analizar las distintas posibilidades, el equipo decidió utilizar MonoGame. Rápidamente se comenzó a desarrollar prototipos para probar el Game Engine, y ganar experiencia para luego poder estimar adecuadamente los requerimientos. Al momento de desarrollar el prototipo de “Deploy en Android/iOS” se encontraron varios problemas (detallados más adelante), por lo que el equipo decidió continuar investigando por otras alternativas para el desarrollo y se encontró LibGDX. Nuevamente se desarrollaron todos los prototipos, esta vez sin ningún tipo de problemas.

Por otra parte, si bien la idea fuerza del juego, y las principales características del mismo se mantuvieron estables, algunos de los requerimientos enfocados en la jugabilidad cambiaron. Esto provocó cierto re-trabajo, pero el equipo de desarrollo respondió rápidamente evitando un retraso en el cronograma.

Otro problema al que se enfrentó el equipo fue a la mala estimación de los requerimientos relacionados al arte en las primeras etapas de la Producción. En algunas oportunidades, el equipo de desarrollo tuvo que esperar a que los recursos gráficos estuvieran prontos para poder continuar. Sin embargo, este problema tampoco provocó retrasos en el cronograma gracias a que se utilizaban recursos provisionales.

12.5. Lecciones aprendidas

La principal lección aprendida durante el proyecto respecta a la gestión de proyectos, principalmente el relevamiento y la validación de los requerimientos, el aseguramiento de la calidad, y la gestión en general. Fue muy importante llevar un buen registro de horas trabajadas para lograr ver el rendimiento del equipo.

Otra gran lección aprendida concierne exclusivamente al desarrollo de videojuegos. Aprendimos que antes de comenzar con el proceso de desarrollo, es muy importante comprobar que el videojuego será atractivo. Una forma de hacer esto es probando el juego en papel.

13. Bibliografía

- [1] Digital Buzz Blog. “Infographic: 2012 Mobile Growth Statistics | Digital Buzz Blog” (2013, Jun 13) [Online]. Disponible: www.digitalbuzzblog.com/infographic-2012-mobile-growth-statistics
- [2] Apple Inc. “Apple Developer” (2013, Jun 10) [Online]. Disponible: <https://developer.apple.com>
- [3] Google. “Google Play Developer Console” (2013, Jun 10) [Online]. Disponible: <https://play.google.com/apps/publish>
- [4] Google. “Draw Something - Android Apps on Google Play” (2013, Jun 9) [Online]. Disponible: <https://play.google.com/store/apps/details?id=com.omgpop.dstpaid>
- [5] Oracle. “Code Conventions for the Java Programming Language” (2013, Jun 9) [Online]. Disponible: <http://www.oracle.com/technetwork/java/index-135089.html>

14. Estructura de la documentación CD

Directorio Documentación:

Proyecto Crystal Clash.PDF

- Gestión de Proyecto
 - Plan de Proyecto.PDF

- Game Design
 - Game Concept Document.PDF
 - Game Design Document.PDF

- Arquitectura
 - Technical Design Document.PDF

- Gestión de la Calidad
 - Plan de SQA.PDF

- Gestión de la Configuración
 - Plan de SCM.PDF

15. Anexos

ANEXO A	PLAN DE PROYECTO
ANEXO B	PLAN DE SQA
ANEXO C.....	PLAN DE SCM
ANEXO D	GAME CONCEPT DOCUMENT
ANEXO E.....	GAME DESIGN DOCUMENT
ANEXO F	TECHNICAL DESIGN DOCUMENT