



Universidad
Carlos III de Madrid

Departamento de Informática

TRABAJO FIN DE GRADO

Análisis, Diseño y Desarrollo de un Juego de Estrategia para Dispositivos Móviles

Autor: Iván Lim Blanco

Tutor: José Luis López Cuadrado

Leganés, julio de 2014

Título: Análisis, Diseño y Desarrollo de un Juego de Estrategia para Dispositivos Móviles.

Autor: Iván Lim Blanco.

Director: José Luis López Cuadrado.

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

A mi padre y a mi hermano. Sin ellos, este proyecto habría terminado hace dos años.

Resumen

Este proyecto documenta de principio a fin el desarrollo de un juego estilo Tower Defense para tabletas, comenzado con el análisis en el mercado actual de productos similares y continuando con su diseño y codificación.

El juego se elabora para un cliente que desea publicar su primer juego para tabletas en el mercado, aportando una propuesta nunca antes vista en el género Tower Defense, un modo para dos jugadores en el mismo dispositivo.

El cliente desea que el juego disponga de una base sencilla, cuyo desarrollo no tenga riesgo de alargarse excesivamente, y sobre la que se pueda añadir la característica que propone, garantizando un correcto funcionamiento y una calidad y acabado al nivel de los juegos similares que existen en el mercado.

De desde el estudio de la situación actual hasta la implementación del código del juego, se explica en detalle cada paso acometido en su desarrollo, haciendo especial hincapié en las fases de análisis y diseño, que resultan imprescindibles para garantizar que en el proyecto se cumplan los objetivos sin desperdiciar tiempo ni recursos. Asimismo, se ofrece la documentación adecuada para la implantación de los servicios y el software necesarios para ejecutar el juego en un ordenador corriente (en un simulador), y una guía de usuario apropiada para que cualquier usuario pueda comprender y manejar el producto.

Este proyecto es analizado en su mayor medida desde la perspectiva del diseño e implementación, estudiando sus probabilidades de éxito en el mercado pero sin abarcar las fases de publicación y venta. Si bien el producto final no puede garantizar un número elevado de descargas en los mercados en los que se ofrecerá, sí que se ha trabajado para que cumpla todos los requisitos de estabilidad y funcionalidad para ofrecer un nivel alto de calidad al usuario final.

Palabras clave: Juego, Tower Defense, Estrategia, Tableta, Dispositivo móvil, Táctil, Multijugador, 2 jugadores.

Abstract

This project documents the entire development of a Tower Defense style game for tablets, focusing on design and coding, and analyzing the current similar products on the market.

The game is made for a client who wants to publish his first game for tablets on the market, providing a proposal never seen before in the Tower Defense genre, a two player mode in the same device.

The client wants the game to be built with a simple base on which to add the feature he proposes, preventing the risk of over-extend the development time, and ensuring correct performance and quality finish, like similar games for tablets on the market.

From the study of the current situation to the implementation of the game code, every step undertaken in the game development is explained in detail, with particular emphasis on the analysis and design phases, which are essential to ensure that the project meets the objectives without wasting time or resources. Also, the proper documentation for the implementation of services and software needed to run the game on an average computer (in a simulator) is shown, and a user guide appropriate for any user to understand and manage the product is offered at the end of the document.

This project is analyzed from the design and implementation perspective, studying their chances of success in the market but without covering the publication and sale phases. While the end product may not guarantee a high number of downloads in the markets in which it will be offered, it's developed to meet all the requirements of stability and functionality to deliver high quality product to the end-user.

Keywords: Game, Tower Defense, Strategy, Tablet, Mobile Device, Touch, Multiplayer, 2 players.

Tabla de Contenidos

Índice de Figuras	14
Índice de Tablas	18
1 Introducción y Objetivos	22
1.1 Introducción	22
1.2 Objetivos	24
1.2.1 Objetivo final	24
1.2.2 Enumeración de objetivos principales	24
1.3 Origen del Proyecto	25
1.4 Medios Empleados.....	26
1.5 Estructura de la memoria	27
1.6 Definiciones.....	29
2 Estudio de viabilidad del sistema y Análisis socioeconómico	35
2.1 Establecimiento del alcance del sistema	35
2.1.1 Estudio de la solicitud	35
2.1.2 Identificación del alcance del sistema	36
2.2 Migración a dispositivos móviles	37
2.2.1 Aspectos fundamentales de la migración	38
2.2.2 Migración específica para este proyecto	44
2.3 Estado del Arte y Estudio de la situación actual	46
2.3.1 Situación del mercado.....	46
2.3.2 Situación del género Tower Defense	53
2.3.3 Estudio de la competencia	54
2.3.4 Perspectivas de impacto	61
2.3.5 Marco regulador.....	63
2.3.6 Diagnóstico del estudio de la situación actual.....	64
2.4 Entorno socio-económico	65

2.4.1	Búsqueda de Publisher.....	65
2.4.2	Publicación independiente alternativa	67
2.5	Alternativas de diseño	68
2.5.1	Alternativa 1. Aplicaciones nativas	69
2.5.2	Alternativa 2. Flash CS6.....	70
2.5.3	Alternativa 3. Unity3d	71
2.5.4	Selección de la mejor alternativa de diseño	73
3	Gestión del proyecto.....	75
3.1	Ciclo de vida	75
3.1.1	Modelo en cascada retroalimentado.....	75
3.1.2	Definición de fases del modelo	76
3.2	Organización del proyecto	77
3.2.1	RBS.....	77
3.2.2	Responsabilidades de los recursos humanos	78
3.3	Planificación	79
3.4	Estimación de costes.....	80
4	Desarrollo de la aplicación	82
4.1	Análisis	82
4.1.1	Sobre los requisitos de Usuario y de Software	82
4.1.2	Requisitos de Usuario.....	86
4.1.3	Requisitos de Software	96
4.1.4	Trazabilidad de requisitos	110
4.1.5	Casos de uso.....	112
4.1.6	Diagramas de casos de uso	133
4.2	Diseño	141
4.2.1	Gestión de la migración a dispositivos de pantalla multitáctil	141
4.2.2	Arquitectura de la aplicación	145
4.2.3	Diagramas de flujo	147
4.2.4	Diseño de clases	153
4.3	Implementación	160

4.3.1	Entorno.....	160
4.3.2	Gráficos y sonidos	161
5	Guía de implantación	163
5.1	Instalación de Flash Professional CC.....	163
5.1.1	Nota previa sobre Adobe CS6 y CC	163
5.1.2	Descarga e instalación.....	163
5.2	Instalación de Adobe AIR	166
5.2.1	Adobe AIR.....	166
5.2.2	Adobe AIR sdk	166
5.3	Configuración de publicación	169
5.3.1	Configuración de parámetros	169
5.3.2	Configuración del simulador	172
5.4	Publicación en mercados	173
5.4.1	Cuentas de desarrollador	173
5.4.2	Creación de las cuentas.....	174
5.4.3	Creación y subida de la aplicación	174
6	Manual de usuario	177
6.1	Introducción.....	177
6.1.1	Género.....	177
6.1.2	Pantalla.....	178
6.1.3	Elementos del escenario	179
6.2	Manejo básico	182
6.2.1	Construir torre.....	182
6.2.2	Mejorar y vender torre	183
6.2.3	Enviar enemigo.....	183
6.2.4	Comprar habilidad especial.....	185
6.2.5	Mejorar habilidad especial.....	185
6.2.6	Manejo de habilidades.....	186
6.2.7	Opciones y guía de batalla	187
6.3	Elementos de juego	188

6.3.1	Energía.....	188
6.3.2	Estrellas	189
6.3.3	Unidades.....	189
6.3.4	Unidades Extra	192
6.3.5	Torres	192
6.3.6	Atributos de torres	194
6.3.7	Habilidades especiales	194
6.4	Menú de opciones y mute	196
6.4.1	Menú de opciones.....	196
6.4.2	Mute	196
6.4.3	Guía de juego	196
6.5	Guía en el dispositivo	197
7	Conclusiones	202
7.1	Conclusiones del Proyecto.....	202
7.1.1	Análisis de objetivos.....	202
7.1.2	Adaptaciones de juegos para dispositivos móviles.....	203
7.1.3	Mercado	204
7.2	Conclusiones personales.....	204
7.3	Líneas futuras	205
8	Referencias	207

Índice de Figuras

Figura 1. Pantallazo de adaptación a móvil de Earthworm Jim [a.1]	22
Figura 2. Dispositivo de pantalla multitáctil [a.2]	26
Figura 3. Diferencias PC/Móvil Kingdom Rush [a.3].....	38
Figura 4. Modelos en baja poligonización [a.4].....	39
Figura 5. Sprite Sheet de New Super Mario Bros Wii [a.5]	40
Figura 6. Diferentes tipos de eventos táctiles [a.6].....	41
Figura 7. Emulación de mando de consola en dispositivo móvil [a.7]	42
Figura 8. Aproximación de cámara en versión móvil de Fieldrunners [a.8]	43
Figura 9. Ejemplo de juego estilo Tower Defense [a.9]	46
Figura 10. Descargas globales de aplicaciones y juegos [1]	47
Figura 11. Distribución de descargas entre hombres y mujeres [1]	48
Figura 12. Descargas por País [1].....	49
Figura 13. Beneficios según el tipo de aplicación [1]	49
Figura 14. Distribución de beneficios por plataforma [1]	50
Figura 15. Estudio demográfico de usuarios de iOS y Android [1].....	50
Figura 16. Distribución de ingresos por tipo de dispositivo [1]	51
Figura 17. Porcentajes por tipo de dispositivo [1]	52
Figura 18. Predicción de ventas de Portátiles, Smartphones y Tabletás [2]	52
Figura 19. Editor de niveles de Warcraft III [a.10]	54
Figura 20. Kingdom Rush [a.11].....	55
Figura 21. Plants vs Zombies [a.12]	56
Figura 22. Jelly Defense [a.13].....	57
Figura 23. Field Runners [a.14].....	58
Figura 24. Bloons Tower Defense 5 [a.15]	59
Figura 25. Pirate Legends TD [a.16].....	60
Figura 26. Expectativas de descargas estándar [3]	61
Figura 27. Lealtad de usuarios según el género del juego [4].....	62

Figura 28. Página principal de www.fgl.com	66
Figura 29. Subir juego a www.fgl.com	67
Figura 30. Xcode para Mac OS X [a.17]	69
Figura 31. Flash CS6 [a.18].....	70
Figura 32. Unity3d [a.19]	72
Figura 33. Modelo de ciclo de vida en cascada retroalimentado [11].....	76
Figura 34. RBS del proyecto.....	77
Figura 35. Diagrama de Gantt 1.....	79
Figura 36. Diagrama de Gantt 2.....	79
Figura 37. Presupuesto	80
Figura 38. Diagrama de caso de uso: Menú Principal	133
Figura 39. Diagrama de caso de uso: En partida	134
Figura 40. Diagrama de caso de uso: Interactuar con Interfaz de Unidades	135
Figura 41. Diagrama de caso de uso: Interactuar con Interfaz de Habilidades Especiales.....	136
Figura 42. Diagrama de caso de uso: Menú de Interacción con Torre	137
Figura 43. Diagrama de caso de uso: Menú de Ajustes	138
Figura 44. Diagrama de caso de uso: Guía de Juego	139
Figura 45. Diagrama de caso de uso: Interactuar con Habilidad Especial	140
Figura 46. Diagrama de caso de uso: Menú fin de partida	140
Figura 47. Pre-visualización de filtro de sombreado en Flash Professional [a.20]	142
Figura 48. Diferencia de pérdida de calidad en el escalado en bitmaps con respecto al escalado en imágenes en formato vector [a.21].....	143
Figura 49. Evento estándar "Zoom" que requiere capacidad multitáctil del dispositivo [a.22]	144
Figura 50. Esquema del patrón MVC [11]	146
Figura 51. MVC del Sistema	147
Figura 52. Simbología en diagramas de flujo	147
Figura 53. Diagrama de flujo general de la aplicación	148
Figura 54. Diagrama de flujo de proceso de compra de torre	149

Figura 55. Diagramas de flujo de Habilidades Especiales	150
Figura 56 . Diagrama de flujo de envío de enemigos	151
Figura 57. Diagrama de flujo de eliminación de enemigos	152
Figura 58. Diagrama de flujo de recolección de estrellas	152
Figura 59. Diagrama de clases	154
Figura 60. Definición de Clase: Partida.....	155
Figura 61. Definición de Clase: Avisporro.....	156
Figura 62. Definición de Clase: Helicoptero	156
Figura 63. Definición de Clase: Torre	157
Figura 64. Definición de Clase: Bomba.....	158
Figura 65. Definición de Clase: Enemigo	158
Figura 66. Definición de Clase: Bala	159
Figura 67. Layout de Flash	160
Figura 68. Librería de Flash para el proyecto	161
Figura 69. Web de descarga de Adobe Flash CC	164
Figura 70. Centro de descargas de Adobe Creative Cloud	164
Figura 71. Descarga de Flash Professional CC	165
Figura 72. Ventana de Creative Cloud.....	165
Figura 73. Descarga de Adobe AIR	166
Figura 74. Descarga de Adobe AIR sdk	167
Figura 75. Cuadro de ayuda de Flash Pro CC.....	168
Figura 76. Administrador de sdk de AIR de Flash CC.....	168
Figura 77. Configuración de publicación de Flash Pro CC	169
Figura 78. Configuración de Air para aplicación Android: General.....	170
Figura 79. Configuración de Air para aplicación Android: Implementación	171
Figura 80. Configuración del Simulador de aplicaciones Android de AIR	172
Figura 81. Vista general de la pantalla de batalla	177
Figura 82. Posición de los jugadores	178
Figura 83. Interfaz de construcción de torre.....	182
Figura 84. Interfaz de mejora de torre	183

Figura 85. Interfaz de envío de enemigos	184
Figura 86. Tiempo de espera entre enemigos.....	184
Figura 87. Habilidades especiales.....	185
Figura 88. Mejora de habilidades especiales	185
Figura 89. Botones de configuración.....	187
Figura 90. Energía	188
Figura 91. Menú Overview	197
Figura 92. Menú Troops	198
Figura 93. Menú Defenses.....	199
Figura 94. Menú Graphics	200

Índice de Tablas

Tabla 4. Estudio de alternativas de solución	73
Tabla 5. Ejemplo de Requisito de Usuario/Software	83
Tabla 6. RU.R:001. Idioma del juego	86
Tabla 7. RU.R:002. Compatibilidad con tabletas Android	86
Tabla 8. RU.R:003. Compatibilidad con tabletas iPad	87
Tabla 9. RU.R:004. Visualización en dispositivos.....	87
Tabla 10. RU.R:005. Disposición multijugador	88
Tabla 11. RU.R:006. Lenguaje apropiado	88
Tabla 12. RU.C:001. Pantalla inicial	89
Tabla 13. RU.C:002. Menú Principal	89
Tabla 14. RU.C:003. Área de juego	90
Tabla 15. RU.C:004. Victoria/Derrota	90
Tabla 16. RU.C:005. Menús en batalla	91
Tabla 17. RU.C:006. Envío de enemigos	91
Tabla 18. RU.C:007. Torres defensivas	92
Tabla 19. RU.C:008. Elementos especiales.....	92
Tabla 20. RU.C:009. Recursos	93
Tabla 21. RU.C:010. Interfaz integrada	93
Tabla 22. RU.C:011. Diversión táctil	94
Tabla 23. RU.C:012. Daño específico.....	94
Tabla 24. RU.C:013. Nivel de enemigos.....	95
Tabla 25. RS.N:001. Idioma del juego.....	96
Tabla 26. RS.N:002. Compatibilidad con sistemas operativos	96
Tabla 27. RS.N:003. Rendimiento	97
Tabla 28. RS.N:004. Visualización en dispositivos.....	97
Tabla 29. RS.N:005. Disposición multijugador	98
Tabla 30. RS.N:006. Lenguaje apropiado.....	98

Tabla 31. RS.F:001. Pantalla inicial	99
Tabla 32. RS.F:002. Menú Principal	99
Tabla 33. RS.F:003. Área de juego	100
Tabla 34. RS.F:004. Color de equipo.....	100
Tabla 35. RS.F:005. Área táctil	101
Tabla 36. RS.F:006. Victoria/Derrota	101
Tabla 37. RS.F:007. Menús en batalla	102
Tabla 38. RS.F:008. Menú inGame de Opciones	102
Tabla 39. RS.F:009. Menú inGame de Guía de jugador.....	103
Tabla 40. RS.F:010. Envío de enemigos	103
Tabla 41. RS.F:011. Enemigo especial	104
Tabla 42. RS.F:012. Torres defensivas	104
Tabla 43. RS.F:013. Construcción de torres	105
Tabla 44. RS.F:014. Elementos especiales.....	105
Tabla 45. RS.F:015. Evolución de defensas	106
Tabla 46. RS.F:016. Recurso principal	106
Tabla 47. RS.F:017. Matriz de precios	107
Tabla 48. RS.F:018. Interfaz integrada	107
Tabla 49. RS.F:019. Diversión táctil	108
Tabla 50. RS.F:020. Daño específico	108
Tabla 51. RS.F:021. Nivel de enemigos.....	109
Tabla 52. Matriz de Trazabilidad	111
Tabla 53. Esquema de Casos de uso	112
Tabla 54. CU:000. Cerrar la aplicación	113
Tabla 55. CU:001. Iniciar una partida	114
Tabla 56. CU:002. Construir una torre	115
Tabla 57. CU:003. Mejorar una torre	116
Tabla 58. CU:004. Vender una torre.....	117
Tabla 59. CU:005. Enviar un enemigo	118
Tabla 60. CU:006. Comprar habilidad especial	119

Tabla 61. CU:007. Mejorar habilidad especial.....	120
Tabla 62. CU:008. Cambiar calidad gráfica.....	121
Tabla 63. CU:009. Habilitar tipos de sonido	122
Tabla 64. CU:010. Terminar batalla	123
Tabla 65. CU:011. Retomar batalla.....	124
Tabla 66. CU:012. Victoria/Derrota	125
Tabla 67. CU:013. Acceder a la Guía de juego.....	125
Tabla 68. CU:014. Activar/desactivar sonido	126
Tabla 69. CU:015. Lanzar Bomba.....	128
Tabla 70. CU:016. Mover Helicóptero	129
Tabla 71. CU:017. Mover Avispa	130
Tabla 72. CU:018. Recoger Estrella.....	131
Tabla 73. CU:019. Enviar Enemigo.....	132

1 Introducción y Objetivos

1.1 Introducción

En el desarrollo de software, la creación de un videojuego supone un reto único. No se trata de una herramienta con una utilidad práctica determinada, sino de un producto de entretenimiento, en el que al margen de la funcionalidad y eficiencia del sistema, la finalidad no es más que entretener y divertir al usuario.

Durante estos últimos años, el desarrollo a pasos agigantados de la telefonía móvil, junto con el crecimiento exponencial de los dispositivos móviles tanto en ventas como en prestaciones, ha dado lugar a una nueva rama de explotación en la industria de los videojuegos.

Los nuevos dispositivos multitáctiles ofrecen en la actualidad nuevas opciones a la hora de plantear la comunicación entre el usuario y el juego. La sustitución del teclado y el ratón por los sensores táctiles permite plantear experiencias de juego nunca antes vistas y acercarlas a un público casual que hasta ahora no había tenido interés en este tipo de entretenimiento.



Figura 1. Pantallazo de adaptación a móvil de Earthworm Jim [a.1]

Es evidente que en el mundo de los videojuegos, la oferta en cuanto a temáticas y dinámicas de juego era muy amplia ya antes de que aparecieran los dispositivos multitáctiles, por lo que más que crear juegos completamente nuevos, la mayoría de empresas han optado por plantear portabilidades o adaptaciones de juegos ya existentes a este nuevo formato.

Aunque partir de géneros que ya han tenido éxito en PC, parecía ofrecer ciertas garantías, muchas de estas empresas están encontrando complicaciones que hacen que el proceso pueda no resultar rentable en muchos casos.

Por un lado, las diferencias entre el formato original y la nueva configuración de los dispositivos móviles son tan grandes que es difícil definir en qué medida es viable en términos de diseño un cambio de enfoque tan radical, y por otro, los mercados de aplicaciones para dispositivos móviles son tan recientes, que es difícil valorar el éxito que puede tener un juego en concreto basándose en experiencias o estadísticas previas, ya que no existen o no son fiables más allá de 3-5 años atrás.

Conociendo las dificultades, este proyecto plantea la adaptación de un juego existente en interfaz convencional (PC/web) a plataformas multitáctiles, empleando las características de estos nuevos dispositivos en favor de una mejora en la experiencia de juego y aportando una visión nueva y moderna a un género de juego aún por explotar en el mercado de dispositivos móviles.

Se estudian y tratan todas las complicaciones que surgen en esta difícil tarea y se ofrecen soluciones eficientes para garantizar que el producto final es competente y de calidad.

1.2 Objetivos

1.2.1 Objetivo final

El objetivo de este proyecto es desarrollar para un hipotético cliente, un videojuego estilo Tower Defense (en adelante, el proyecto), desde su fase de diseño y planificación, hasta la fase previa a la publicación del producto. Asimismo, se estudiarán los diferentes mercados de difusión y los posibles métodos de publicación y marketing para evaluar las implicaciones, riesgos y beneficios que supondría el lanzamiento del producto en el futuro.

1.2.2 Enumeración de objetivos principales

Teniendo en cuenta la propuesta del cliente sobre la que se trabaja en este proyecto, se pueden diferenciar varios aspectos principales en los que se ha de trabajar para la consecución del objetivo principal.

Dispositivos

Crear el videojuego de forma que sea compatible tanto en sistemas Android como iOS, en dispositivos genéricos del mercado.

Migración

Como se detalla en el próximo apartado, en este proyecto se reutilizará un software creado anteriormente para navegadores web de ordenadores de escritorio. Se ha de trabajar para conseguir una correcta adaptación de un sistema manejado con ratón al nuevo sistema multitáctil.

Entorno

Emplear el entorno Adobe Flash Professional CS6 y la última versión disponible del motor para plataformas móviles Adobe AIR.

Compatibilidad

Compilar a partir de una única versión del juego compatible con Android e iOS, con un tamaño de pantalla que se adapte fácilmente a las diferentes dimensiones de los dispositivos en los que se jugará.

Jugabilidad

Ofrecer un modo batalla para dos jugadores, que se enfrenten simultáneamente en el mismo dispositivo jugando uno contra otro.

Idioma

Mostrar el texto en inglés como un único idioma, y conseguir que el videojuego sea fácil de entender incluyendo la menor cantidad de texto posible.

1.3 Origen del Proyecto

Hace aproximadamente 8 años, un grupo de desarrolladores creó como hobby un pequeño juego estilo Tower Defense clásico. La aplicación se desarrolló en formato Flash de forma que alojada en un servidor, fuera accesible a través de navegadores web de PC's convencionales con el reproductor Flash Player instalado.

En este proyecto, se estudia un nuevo enfoque estratégico para el grupo de desarrollo y para el juego, de forma que pasando por un profundo trabajo de adaptación, pueda servir para cubrir la necesidad actual del cliente.

Para ello, se evalúa ampliamente el mercado, descubriendo una oportunidad de negocio en adaptaciones de juegos estilo Tower Defense para plataformas móviles multitáctil, y se detecta un gap en la posibilidad de enfocar ese tipo de juego de forma que sea manejable para dos jugadores simultáneamente.

El valor de este proyecto, y lo que consume la mayor parte del esfuerzo de desarrollo es, por tanto, la transformación de una idea sin excesivo valor de mercado, en un nuevo producto que aprovecha la nueva tecnología de las plataformas multitáctiles, explotando un género de juego ya conocido para otorgarle un enfoque innovador y nunca antes visto.



Figura 2. Dispositivo de pantalla multitáctil [a.2]

1.4 Medios Empleados

A continuación se enumera la lista completa de medios empleados para el desarrollo del proyecto.

Ordenador

Con requerimientos suficientes para ejecutar con fluidez las aplicaciones necesarias para el desarrollo del proyecto.

Entorno de desarrollo

Adobe Flash Professional CS6.

Entorno de ejecución

Adobe AIR para plataformas móviles. Opera en conjunto con Flash para permitir que la aplicación pueda compilarse para dispositivos móviles táctiles.

Editor de Audio

Audacity. Software libre que se empleará para el tratamiento básico de archivos de audio (ajuste de volúmenes y pequeñas correcciones) si se requiriese.

Cuentas de desarrollador

De aplicaciones móviles de Google y Apple. Estas cuentas son proporcionadas por el cliente, concediendo acceso de desarrollador al programador de la aplicación, para que pueda hacer pruebas en dispositivos reales.

iPad 2

Para probar la aplicación en una versión de iPad de capacidad intermedia con el sistema operativo actualizado.

Tableta Android

De gama intermedia, para probar la aplicación en un dispositivo con sistema operativo Android de capacidad estándar.

1.5 Estructura de la memoria

Diferenciando las partes principales del desarrollo, el documento se divide en ocho bloques descritos a continuación:

1. Introducción y objetivos

Introducción y motivación del proyecto. Se enumeran los objetivos y metas principales del mismo, así como los medios empleados para llevarlos a cabo. Se define la estructura de la memoria y se explican términos técnicos que se emplearán a lo largo del documento.

2. Estudio de viabilidad del sistema y Análisis socioeconómico

Se estudia la solicitud del cliente en relación con la situación actual del mercado en el que se desea publicar la aplicación, evaluando productos similares para poder concluir si el suyo encaja en la demanda actual y estudiando las posibles alternativas de diseño.

3. Gestión del proyecto

Se describen en detalle la planificación y estimación de costes del proyecto, además de su ciclo de vida, y se enumeran los recursos necesarios para su desarrollo incluyendo los recursos humanos, con sus respectivos roles y funciones dentro del equipo de desarrollo.

4. Desarrollo de la aplicación

Análisis del sistema con enumeración de requisitos de usuario y software, así como casos de uso. Se define la arquitectura y modelado del sistema como parte de su diseño. También se describe el entorno utilizado para su implementación.

5. Guía de implantación

Se describen todos los pasos necesarios para instalar y configurar los programas que harán falta para ejecutar la aplicación en un ordenador común con sistema operativo Windows XP SP3 o superior.

6. Manual de usuario

Se muestra el manual de uso del producto para un usuario genérico de la aplicación. En este caso, cualquier jugador del juego. Cualquier usuario deberá ser capaz de usar la aplicación siguiendo este manual.

7. Conclusiones

Se estudia el sistema una vez desarrollado, y se analiza si se han logrado cumplir todos los objetivos del cliente, ofreciendo líneas futuras de desarrollo. También se analiza lo que ha supuesto personalmente el desarrollo del proyecto para el autor de la memoria.

8. Referencias

Enumeración de las referencias y enlaces a sitios web que se han empleado para la elaboración de esta memoria y el desarrollo del proyecto en general.

1.6 Definiciones

Adobe Creative Cloud

Es un servicio online de Adobe que proporciona acceso a todos sus programas de diseño y edición a través de la nube. En lugar de vender sus programas, Adobe, desde el último año, ofrece acceso a este servicio a cambio de una cuota mensual asequible.

Adobe Flash Professional

Una de las herramientas más comunes de Adobe. Es un software de tratamiento de gráficos vectoriales en el que también se permite manejar código Actionscript.

AIR

Adobe Integrated Runtime. Es un entorno de ejecución multiplataforma que utiliza Flash, entre otros, y permite la ejecución de RIA (Rich Internet Applications) como aplicaciones de escritorio.

Android

Es el Sistema operativo de Google comúnmente utilizado para dispositivos móviles táctiles. Está basado en el kernel de Linux.

AS (ActionScript)

Lenguaje de programación orientada a objetos para la plataforma Flash.

AS3 (ActionScript)

Es la versión más extendida y más completa del lenguaje ActionScript. Mejora considerablemente el rendimiento de las aplicaciones Flash.

Audacity

Software libre multiplataforma empleado para la edición y tratamiento de cualquier formato de audio.

Balance (videojuegos)

En videojuegos, se entiende como balance la cualidad de un juego de disponer de una mecánica equilibrada, en la que todos los elementos interactúan de forma apropiada adecuándose a la capacidad del jugador.

Blizzard Entertainment

Empresa desarrolladora y distribuidora de videojuegos estadounidense fundada en 1994. Autora de juegos de gran fama mundial como World of Warcraft y Starcraft 2, que son actualmente de los más vendidos en el mundo.

Bug

En informática, un error de software que desencadena un resultado no deseado.

Gap

En el ámbito de la ingeniería de software, es sinónimo de: Vacío, carencia, necesidad.

Género (de un juego)

El género de un videojuego especifica su clasificación atendiendo a elementos que tenga en común con otros grupos de videojuegos.

Google Chrome

Es el navegador web desarrollado por Google apoyándose en diferentes infraestructuras de código abierto.

iOS

Sistema operativo para plataformas móviles de Apple. Actualmente su última versión es iOS 7, y sólo se permite su uso en dispositivos de la marca Apple. La letra “i” es una sigla distintiva de los productos Apple.

iPad

Línea de tabletas diseñadas y comercializadas por la marca Apple. Actualmente en versión 3 con sistema operativo iOS 6.

Jugabilidad (en videojuegos)

Atributo de un videojuego que especifica la medida en la que este es comprensible y accesible para el usuario, y la coherencia con la que se establece la comunicación entre el juego y el jugador a través de la interfaz de usuario.

Layout

Se entiende como Layout en términos de software informático, la distribución espacial de los componentes de un programa en el monitor.

Matriz de trazabilidad

Matriz que se desarrolla en el análisis de un software para comprobar la correspondencia entre los requisitos de usuario y de software.

Mecánica de juego

En videojuegos, se entiende como mecánica al conjunto de reglas que condicionan la forma en la que un juego ha de ser jugado.

Modos landscape/portrait (en dispositivos móviles)

Estos modos diferencian los tipos de visualización que puede tener una aplicación en un dispositivo móvil. Landscape ofrece la aplicación en horizontal y portrait en vertical.

Motor (videojuego)

Es el conjunto de rutinas de programación que hacen posible la creación y representación de un videojuego. Se distinguen diferentes tipos de motores según su función principal, como la del renderizado de gráficos, gestión de eventos físicos, de sonido, etc.

Multijugador (videojuegos)

Característica de un videojuego que lo distingue por ser jugable por más de un jugador, generalmente al mismo tiempo.

Mute (botón de juego)

La opción “mute” de un videojuego, al igual que en dispositivos musicales, silencia cualquier sonido que pudiera producirse por la aplicación hasta volver a ser desactivada.

Pop up

Término genérico para denominar a las ventanas emergentes.

RTS (Real Time Strategy)

Subgénero dentro de los juegos de estrategia, que diferencia a aquellos en los que sus eventos se desarrollan en tiempo real, y no por turnos.

SDK (Software Development Kit)

Herramienta de desarrollo de software que permite crear aplicaciones para un sistema concreto.

Simulador (de dispositivos)

Es una herramienta diseñada para emular en un ordenador el comportamiento de un determinado dispositivo y probar el funcionamiento de aplicaciones diseñadas para ese dispositivo en concreto.

Smartphone

Término inglés para denominar a los teléfonos inteligentes, que son teléfonos móviles construidos sobre una plataforma informática móvil para ofrecer servicios similares a los que ofrece un ordenador.

Tablet

Término inglés para denominar a las tabletas. Son un híbrido entre un Smartphone y un ordenador portátil. Su manejo es táctil y normalmente son más grandes y potentes que un teléfono inteligente.

Tower Defense

Género de videojuegos comprendido dentro de la estrategia, que se caracteriza por ofrecer una mecánica en la que el jugador debe construir una estructura defensiva, basada en torres y armamento no móvil, para proteger una base o fortaleza del ataque de multitud de enemigos.

Warcraft III

Título de un juego de la compañía Blizzard Entertainment, clasificado como RTS, que gozó de gran éxito en su lanzamiento en 2002 y en los años posteriores.

Zip (formato de compresión)

Uno de los principales formatos de compresión de archivos informáticos.

2 Estudio de viabilidad del sistema y Análisis socioeconómico

2.1 Establecimiento del alcance del sistema

En este apartado se describen las necesidades planteadas por el cliente, y partiendo de ellas se estudia el alcance del sistema, tratando de definir de la manera más clara posible los objetivos y requisitos que se han de cumplir.

2.1.1 Estudio de la solicitud

El cliente ha solicitado el desarrollo de un juego estilo Tower Defense de dos jugadores para plataformas móviles Android e iOS.

El mercado actual de juegos para tabletas es muy amplio. Como es difícil conseguir un número elevado de descargas sin medios de marketing ni presupuesto suficiente, que es el caso del cliente, su necesidad es disponer de un juego sencillo, estable, de un género ya conocido y que aporte un detalle de originalidad con respecto al resto. Con ello pretende ser capaz de evaluar las posibilidades de publicación en los diferentes mercados de móviles con un producto que no le conlleva una inversión excesiva.

El cliente sugiere que el juego sea del estilo Tower Defense, por tratarse de un género que ya conoce y en el que será capaz de reconocer su calidad. Además sugiere, como aportación a su originalidad, que sea un juego para 2 jugadores.

Valorando su solicitud, se pueden identificar los siguientes objetivos.

- Elaborar un juego de bajo coste.
- El juego debe ser estilo Tower Defense.
- El juego debe disponer de un modo para 2 jugadores simultáneos.
- El juego debe ser simple, para poder dedicar mayor esfuerzo a que su calidad sea destacable.
- Debe estar disponible para dos plataformas: Android e iOS.

2.1.2 Identificación del alcance del sistema

El videojuego será sencillo y de un solo escenario, con un diseño apropiado para dos jugadores que jueguen simultáneamente en el mismo dispositivo. Ofrecerá un balance correcto en una dinámica de juego ágil y divertida.

Menús y pantallas

El flujo de menús será ágil y directo, con un máximo de dos pantallas antes de la batalla y el mínimo de opciones posibles. El jugador principal tendrá el control de la aplicación en los menús principales, y una vez dentro de la pantalla de batalla, los dos jugadores dispondrán de las mismas opciones, exceptuando los botones de acceso a opciones principales (pausar el juego, opciones de gráficos...), que estarán accesibles en un lateral en el centro de la pantalla para que ambos jugadores puedan acceder a ellos.

Durante la batalla, la pantalla se dividirá en dos áreas de juego, para el jugador 1 y el jugador 2. Se distribuirán los elementos de juego de forma que la situación física correcta en la que hayan de colocarse los dos jugadores para jugar sea uno enfrente del otro, con la tableta en posición vertical, y con acceso al 50% de la pantalla que más próximo a cada cual se encuentre.

Una vez concluida una batalla entre los dos jugadores, el área de la pantalla destinada a cada jugador les informará por separado del resultado de la partida (victoria o derrota), y será el ganador el que tenga la opción de volver al menú principal para empezar otra partida. Una vez en el menú, el control volverá a ser del jugador principal hasta volver a entrar en la pantalla de batalla.

Aprendizaje

El juego será intuitivo, de forma que los jugadores puedan aprender a jugar tan solo probando y descubriendo las opciones por sí mismos. Cuando deseen acceder a más información, el juego dispondrá de una guía básica accesible desde un botón común para los dos jugadores visible desde el modo batalla. Ésta guía, así como los menús comunes de configuración del juego, se mostrarán en horizontal ocupando el 100% de la pantalla.

Dinámica de las batallas

Como en cualquier otro Tower Defense, el objetivo de cada jugador será evitar que sus enemigos recorran por completo el camino que les lleva de un punto de salida a otro de llegada, y para ello dispondrán de la opción de construir torres defensivas y objetos dinámicos que les permitan destruir enemigos.

La diferencia fundamental en la dinámica de este videojuego con respecto al resto de Tower Defense, es que los enemigos además de generarse automáticamente en oleadas por el propio juego, será cada jugador el que decida qué enemigo enviar a su oponente y en qué momento.

Así pues, cada jugador, además de la opción de construir objetos defensivos, podrá enviar enemigos a su oponente, aportando una variante ofensiva al género. La forma en la que un jugador gestione sus recursos virtuales comprando opciones de ataque o defensa definirá quién gane la partida.

2.2 Migración a dispositivos móviles

Desde que el desarrollo de aplicaciones para dispositivos móviles se ha convertido en un negocio lucrativo, los desarrolladores de juegos han de plantearse en la creación de cada nuevo título de PC, la posibilidad de ofrecerlo también en formato Android, iOS o Windows Phone.

Además, existen formatos de juegos ya publicados en el mercado de PC, que por sus características, su actualización a formato móvil exige menos esfuerzo. En estos casos suele ser rentable crear una versión adaptada con nuevas funciones y ofrecerla en formato táctil.



Figura 3. Diferencias PC/Móvil Kingdom Rush [a.3]

Como se especifica en la introducción, este proyecto consiste en la migración y adaptación de un juego de navegador a un nuevo formato de aplicación para dispositivos móviles. En los siguientes apartados, se van a exponer los factores que se han de trabajar en este proceso, tanto a nivel genérico para cualquier aplicación, como para este caso en concreto.

2.2.1 Aspectos fundamentales de la migración

Existen funciones básicas que se han de mejorar o actualizar cuando se trabaja en la conversión de un juego a una versión optimizada para dispositivos móviles. A continuación se enumeran las principales.

Rendimiento

Aunque cada vez los móviles y tabletas disponen de mayor potencia y velocidad, hay una diferencia significativa en su rendimiento con respecto a los PCs de sobremesa y portátiles.

Cuando se va a trabajar en la versión de un juego para dispositivos móviles, el principal esfuerzo de desarrollo se ha de dedicar a optimizar al máximo el código de forma que la gestión de imágenes y animaciones por parte del procesador y la tarjeta gráfica del móvil sea lo más eficiente posible. El

rendimiento del juego y el consumo de batería están directamente ligados a estos procesos.

En el caso de los juegos en 3D, se suele trabajar con modelos de escasa poligonización (menor número de polígonos) y texturas de menor tamaño. La capacidad de los modeladores 3D para representar modelos con menor número de polígonos y texturas planas, manteniendo la calidad artística, es fundamental en estos casos.

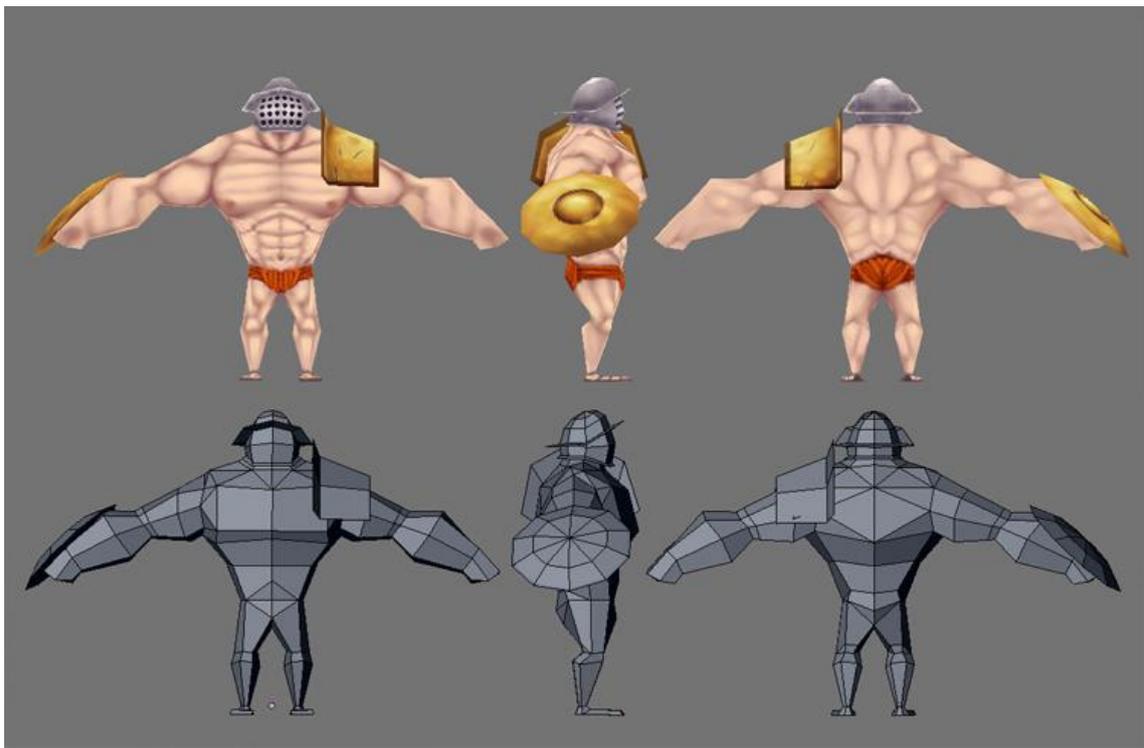


Figura 4. Modelos en baja poligonización [a.4]

En el caso del 2D, el trabajo consiste en conseguir que la tarjeta gráfica emplee el menor número de imágenes al mismo tiempo. Para ello, se suele hacer uso de Sprite sheets, que son imágenes que incluyen, divididos en sectores, un gran número fotogramas de diferentes animaciones empleadas en el juego, de forma que cuando el procesador gráfico pinta esas animaciones en pantalla, lo hace leyendo de la misma imagen, simplemente filtrando el área de la misma que ha de mostrar.

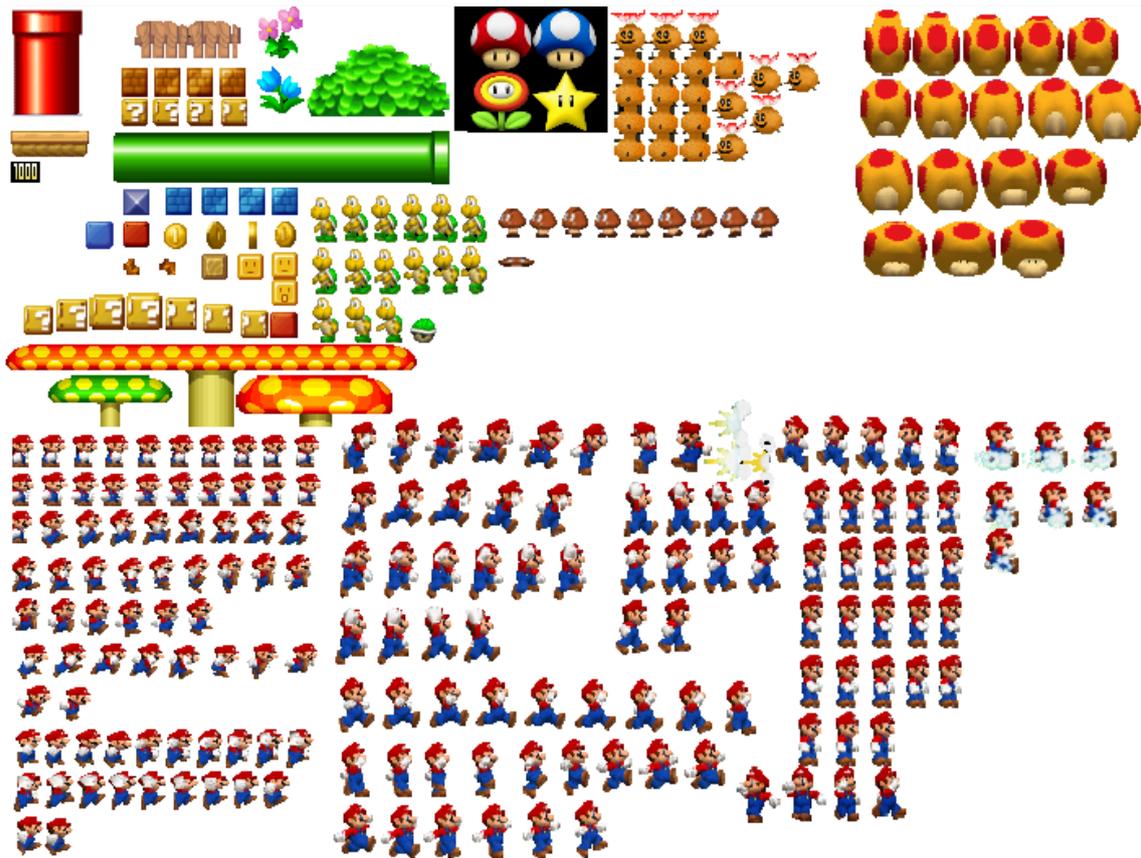


Figura 5. Sprite Sheet de New Super Mario Bros Wii [a.5]

Tamaño

Los dispositivos móviles disponen de unidades de almacenamiento mucho más pequeñas que las de un PC. Se ha de reducir al máximo el tamaño que ocupa el juego para que el jugador pueda descargarlo y almacenarlo con comodidad.

Actualmente, un tamaño aceptable para un juego de móvil es de 5-20 Mb. y de 20-150 Mb. para tabletas.

La reducción de la resolución de imágenes y la conversión del formato de audio .wav sin comprimir a formatos comprimidos como .ogg son los principales métodos para agilizar el peso total de la aplicación.

Formato táctil

Todos los eventos que en un juego de PC se manejan con teclado y ratón, han de transformarse de forma que sean accesibles de forma táctil. Esto implica un profundo trabajo de diseño, ya que son formas muy diferentes de manejar un

juego por parte del usuario y la conversión puede tener un efecto negativo en la jugabilidad.

En el caso del ratón, la mecánica es sencilla. El evento “Clicar” se sustituye por “Tocar”, manteniendo también la dinámica de arrastrar mediante el empleo del dedo en lugar del cursor del ratón. Hay que tener en cuenta que un evento “Tocar” es mucho menos preciso que uno “Clicar”, por lo que las áreas en las que el usuario pulsará, deben ser más grandes y fácilmente accesibles, sin solaparse nunca unas con otras.

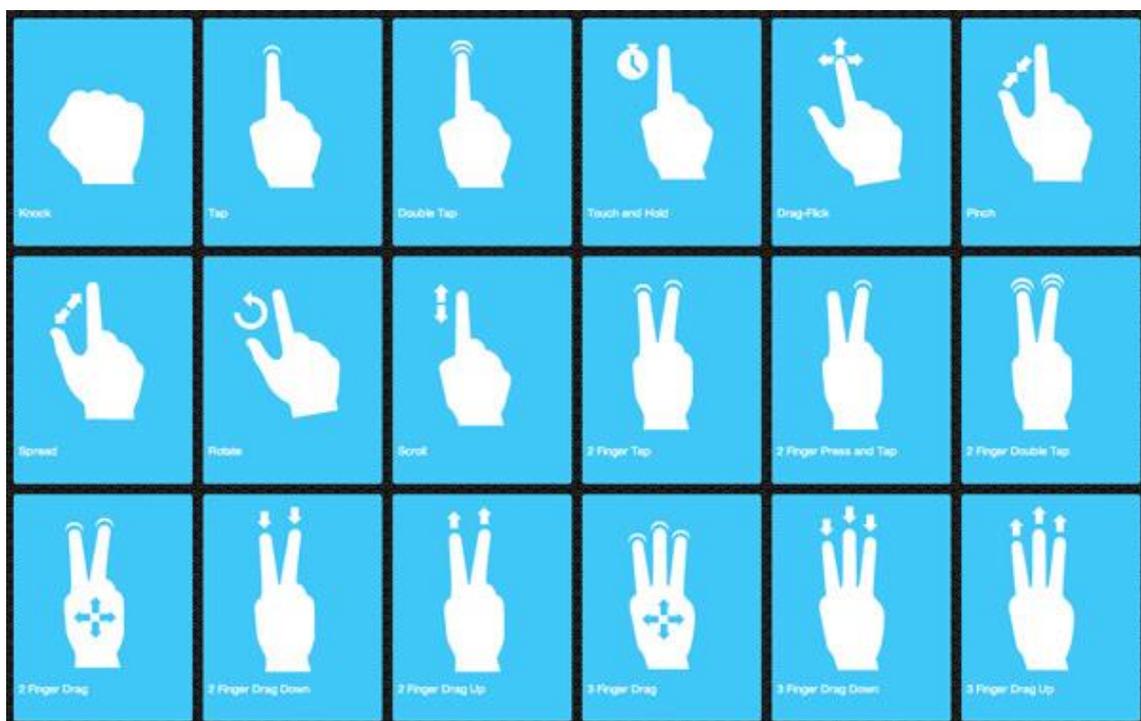


Figura 6. Diferentes tipos de eventos táctiles [a.6]

Es más complejo el caso de los eventos de teclado, ya que es más difícil sustituirlos por eventos táctiles.

Comúnmente en los juegos, el teclado suele utilizarse para mover a un personaje a través de los cursores o las teclas W,A,S y D, y para ejecutar ciertas acciones del personaje a través de la barra espaciadora o letras como Z, X, C... . Normalmente el teclado hace la función que haría el mando en una consola, por lo que en dispositivos móviles, cuando es necesario, no se intenta emular el comportamiento de un teclado, si no de trasladar ese comportamiento a lo que supondría en un mando de consola, y simularlo en el móvil.

Normalmente, esto se consigue ofreciendo al usuario la visualización de un Joystick y diferentes botones virtuales en su dispositivo, permitiéndole manejarlos de forma táctil.



Figura 7. Emulación de mando de consola en dispositivo móvil [a.7]

Los dispositivos móviles disponen además de acciones táctiles adicionales que pueden utilizarse para enriquecer la experiencia de usuario, como los ya conocidos de agrandar y encoger, o arrastrar con más de un dedo.

Tamaño de pantalla

Los móviles y tabletas disponen de pantallas mucho más pequeñas que las de cualquier PC. Textos, iconos y elementos de juego resultan mucho menos visibles cuando se visualizan desde un dispositivo móvil.

La iconografía y los tipos de letra deben ser mucho más grandes, y el escenario de juego debe ofrecerse desde una perspectiva más cercana, permitiendo, si es necesario, que el jugador pueda acercar o alejar la cámara de juego para visualizar el escenario desde diferentes perspectivas mientras la interfaz se mantiene estática y accesible en todo momento.



Figura 8. Aproximación de cámara en versión móvil de Fieldrunners [a.8]

Además del tamaño de las pantallas, hay que tener en cuenta las diferentes relaciones de aspecto con las que cuentan los diferentes dispositivos del mercado: 4:3, 3:2, 5:3, 16:9 y 16:10. Normalmente, lo más apropiado es basar el juego en una relación de 4:3, y extender el área visualizable a lo ancho según corresponda en los demás dispositivos, de forma que la información que se muestre en esas áreas añadidas no sea fundamental para el jugador.

Conexión a internet

Especialmente si se trabaja para tabletas, hay que tener en cuenta que a diferencia de casi la totalidad de los PCs, el dispositivo del usuario podría no estar conectado a internet en el momento de jugar, por lo que es necesario evitar en la medida de lo posible que el juego requiera conexión de datos con un servidor, permitiendo jugar en modo sin conexión.

Usuario final

Por último, se ha de contar con que el usuario de juegos de móviles no es el mismo que el de juegos de PC. En el caso de los móviles, el desarrollador se va a encontrar con un público mucho mayor pero más casual y menos acostumbrado

a formatos concretos de juegos, con lo que se ha de trabajar en tutoriales de enseñanza y mecánicas de juego más sencillas.

2.2.2 Migración específica para este proyecto

Conociendo los principales factores que genéricamente se evalúan a la hora de migrar un juego de PC a una versión móvil, en este apartado se exponen cuáles y en qué medida se han de trabajar para este proyecto en concreto, tanto en lo relativo a factores técnicos de los dispositivos, como en lo relacionado con las mecánicas de juego, que se han de adaptar para el nuevo modo de dos jugadores.

Traslado de eventos de ratón a eventos táctiles

- Los eventos “Tocar” y “Arrastrar” táctiles emularán a los propios de formato ratón, empleando la característica de dispositivos multitouch, que permite generar un identificador asociado a cada evento táctil y así diferenciar los de diferentes jugadores aunque se produzcan simultáneamente.
- Se suprimirán todos los comportamientos asociados al evento de ratón “Over” (que se produce cuando el cursor del ratón se sitúa encima de un objeto), proporcionando la información correspondiente al pulsar por primera vez en un elemento o en diferentes tipos de interfaces.
- Se editarán los tamaños de las interfaces y botones para que la relación entre el jugador y la aplicación sea más amigable y orientada al manejo táctil.

Características técnicas

- El juego se compilará a través de la SDK de Adobe AIR para crear archivos .ipa y .apk compatibles con iPad y dispositivos Android respectivamente.
- Los gráficos de elementos dinámicos se convertirán a secuencias de archivos de imagen .png para optimizar el rendimiento del juego en dispositivos móviles de escasa potencia.
- El tamaño de pantalla se ajustará a dispositivos con diferentes resoluciones, auto-escalándose hasta la máxima altura del dispositivo y mostrando gráficos adicionales para cubrir el ancho total.

Mecánicas de juego

- Se incrementará la vida de los enemigos de forma independiente para cada jugador, y aumentará en función del número de enemigos enviados (estrategia ofensiva).
- Una nueva interfaz de compra de Elementos Especiales estará disponible para el jugador en todo momento. Si dispone de energía suficiente, podrá comprar y/o mejorar el elemento especial que prefiera al margen del número de torres construidas.
- El envío de enemigos por parte del jugador hacia su rival se gestionará a través de una nueva interfaz específica para ello.
- Se producirá un envío automático de 2 enemigos aleatorios cada 30 segundos, cuyo género dependerá del nivel de enemigos del que disponga el jugador rival en el momento de generarse el envío.
- Se editarán completamente los atributos de los enemigos para amoldarse al estilo de juego multijugador, proponiéndolos como distintos recursos de ataque disponibles para ambos jugadores, relacionando su precio con el poder de ataque teórico que le corresponde y ofreciendo un balance correcto con el precio de los elementos defensivos que se utilizarán para contrarrestarlos.

Mecánicas de elementos específicos del juego

- La torreta Gatling se comprará en la interfaz de objetos especiales y actuará al margen del resto de torres, apareciendo en un punto predefinido.
- El helicóptero se manejará de la siguiente manera: si el jugador pulsa con el dedo sobre él y arrastra hasta una nueva posición, marcará el punto de destino del helicóptero, que se dirigirá hacia él con la velocidad correspondiente a su nivel de evolución, salvo que se le marque un nuevo objetivo.
- La unidad Muerciélago volará sobre puntos aleatorios del sector del mapa del jugador contrario al que ha sido enviado. Mientras esté viva, suprimirá la posibilidad de envío de enemigos al jugador afectado.
- Se incluirá una nueva unidad Estrella, que se generará hacia un jugador aleatorio cada vez que se recoja una estrella amarilla en el escenario.
- Se incluirá una nueva unidad Corazón, que podrá aparecer aleatoriamente en el terreno de cualquier jugador cada vez que se le envíen enemigos de forma automática, y otorgará un punto de vida al jugador que la destruya.

2.3 Estado del Arte y Estudio de la situación actual

En este apartado se estudiará el mercado actual de videojuegos para dispositivos móviles, profundizando en los Tower Defense, para valorar las principales opciones que se ofrecen en este género y sus puntos fuertes. Se identificarán las características comunes para plantear mejoras que ofrezcan avances cualitativos.

Ya que el objetivo del videojuego es ser publicado en sistemas Android e iOS, que además suponen la mayor parte de mercado de juegos para dispositivos móviles, el estudio de la situación actual se centrará en estas dos plataformas.

2.3.1 Situación del mercado

El mercado de aplicaciones para dispositivos móviles es inmenso. Se trata de un mercado mundial al que cualquier desarrollador puede acceder con una mínima inversión, y gracias a ello, hay millones de aplicaciones disponibles para cualquier cliente. El 33% del total de aplicaciones descargadas son videojuegos, que se dividen según su género, y dentro de la estrategia, el Tower Defense es uno de los sub-géneros más populares.



Figura 9. Ejemplo de juego estilo Tower Defense [a.9]

El mercado de aplicaciones móviles en general

Por la cantidad de usuarios con acceso al mercado de aplicaciones móviles, la capacidad potencial de éxito de cualquier juego publicado en él es muy grande. Existen alrededor de 500 millones de jugadores, que han invertido en juegos cerca de 7 billones de dólares.

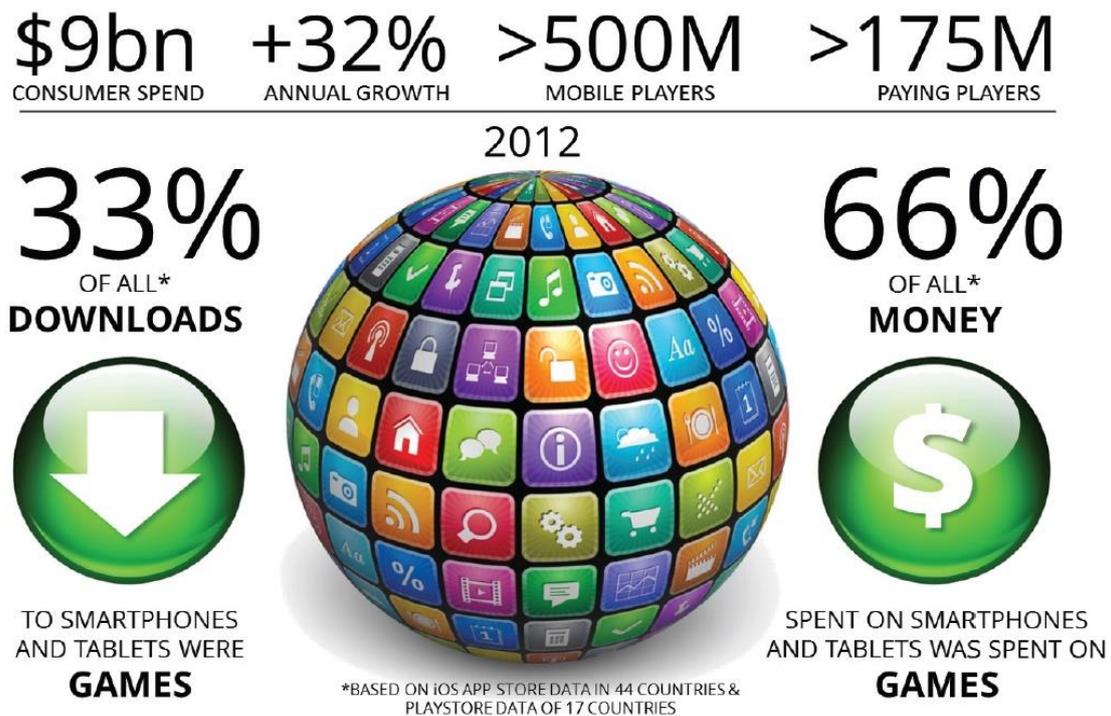


Figura 10. Descargas globales de aplicaciones y juegos [1]

Es interesante tener en cuenta que en los últimos años, se ha observado un incremento importante en el número de mujeres que descargan videojuegos para plataformas móviles, especialmente es países occidentales.

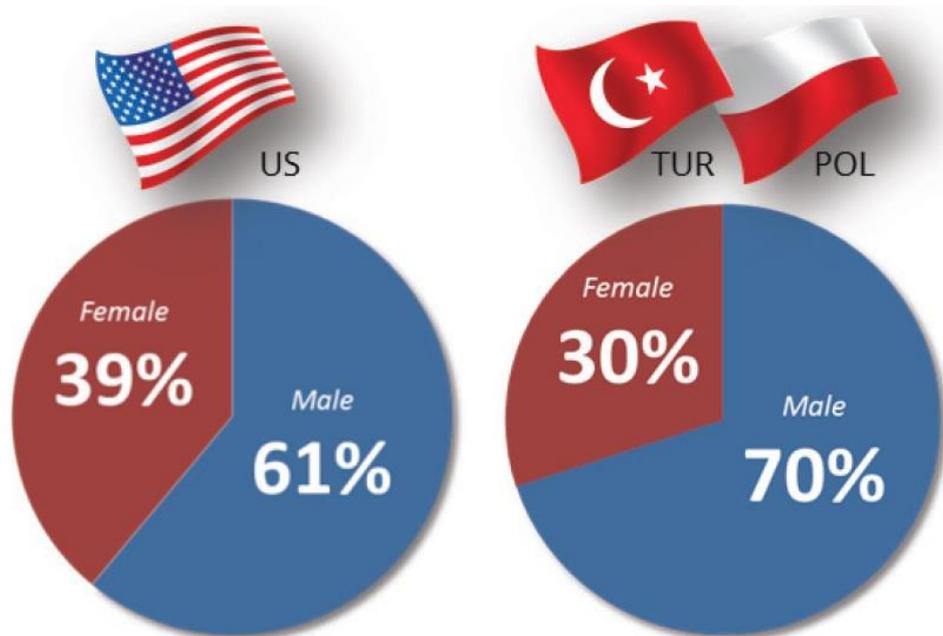


Figura 11. Distribución de descargas entre hombres y mujeres [1]

Idioma

En todos los países, la mayoría de los usuarios instalan juegos en sus dispositivos, pero es importante tener en cuenta cuántos de ellos invierten dinero en ellos. Reino Unido lidera este campo, y Estados Unidos y Australia tienen también porcentajes muy elevados. Teniendo esto en cuenta, así como las poblaciones de dichos países, se hace evidente que en caso de publicar en un solo idioma, lo ideal es escoger el inglés.

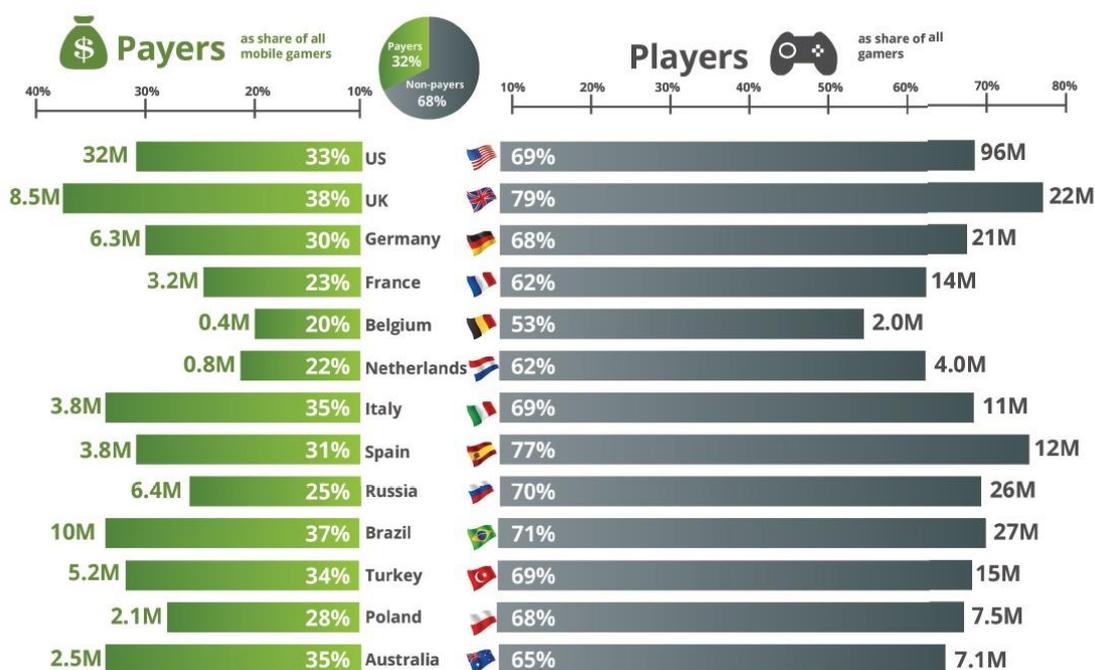


Figura 12. Descargas por País [1]

Beneficios

Cada vez está más extendido el método de negocio Free-to-play, que consiste en ofrecer juegos de descarga gratuita, que después ofertan pequeñas mejoras o elementos adicionales de pago dentro de la aplicación (in-App Purchases). Para un desarrollador sin capacidad para invertir en marketing, se hace casi obligatorio partir de un juego gratuito para facilitar su difusión e intentar generar ingresos dentro del juego.

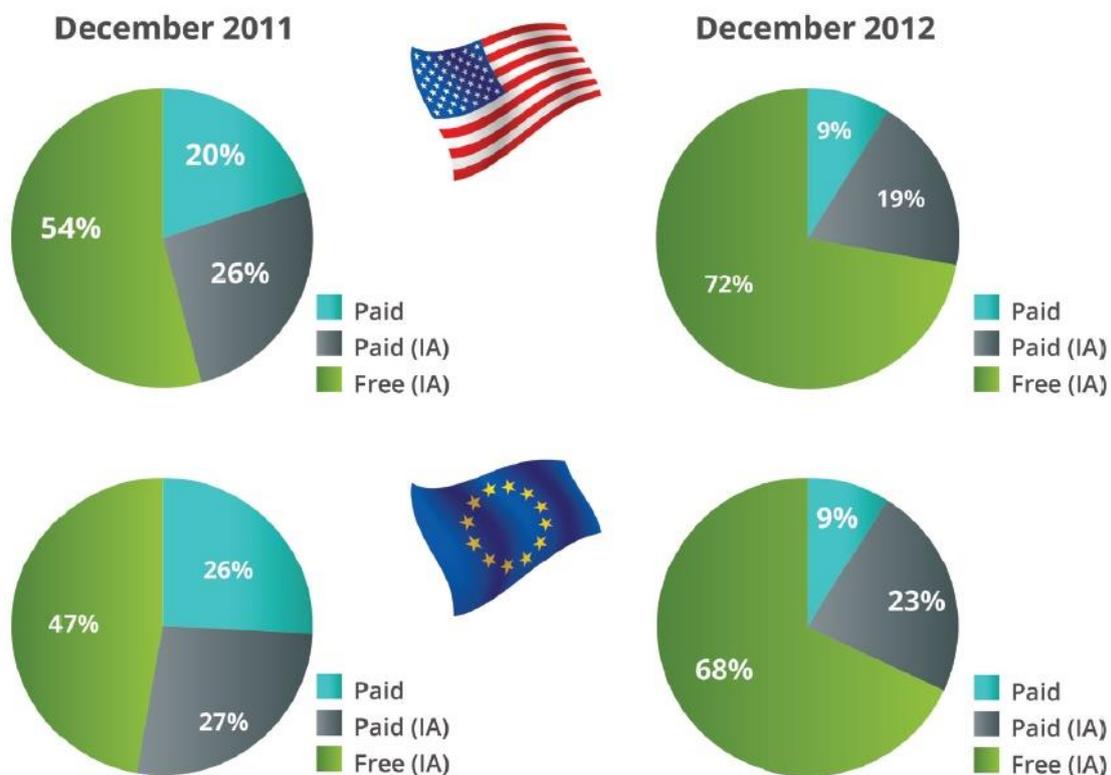


Figura 13. Beneficios según el tipo de aplicación [1]

Apple contra Google Play

Existen diferentes mercados accesibles a la hora de publicar una aplicación según el sistema operativo del que disponga el dispositivo objetivo y la marca del mismo. La mayoría de empresas y desarrolladores independientes, optan por centrarse en los dispositivos con sistemas Android e iOS, que actualmente abarcan más del 90% del mercado mundial tanto de Smartphones como de tabletas.

Aunque Google Play está teniendo un crecimiento mayor, en cuanto a ingresos, el mercado iOS de Apple domina este campo en Europa y Estados Unidos con datos de finales de 2012. Los beneficios de Google Play en Japón, son significativamente superiores.

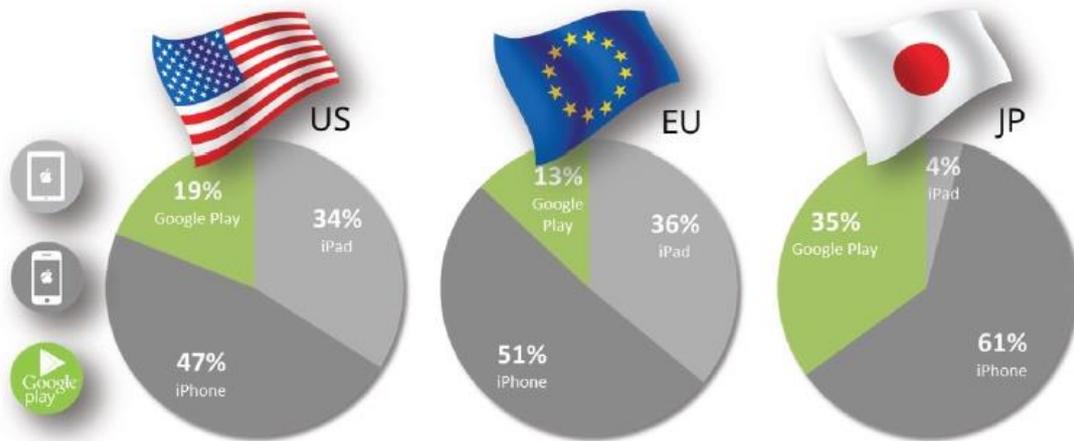


Figura 14. Distribución de beneficios por plataforma [1]

Esta situación es fácilmente comprensible, si se analiza demográficamente a los jugadores de ambos mercados, comprobando que los usuarios de Apple disponen generalmente de un nivel de ingresos mayor.

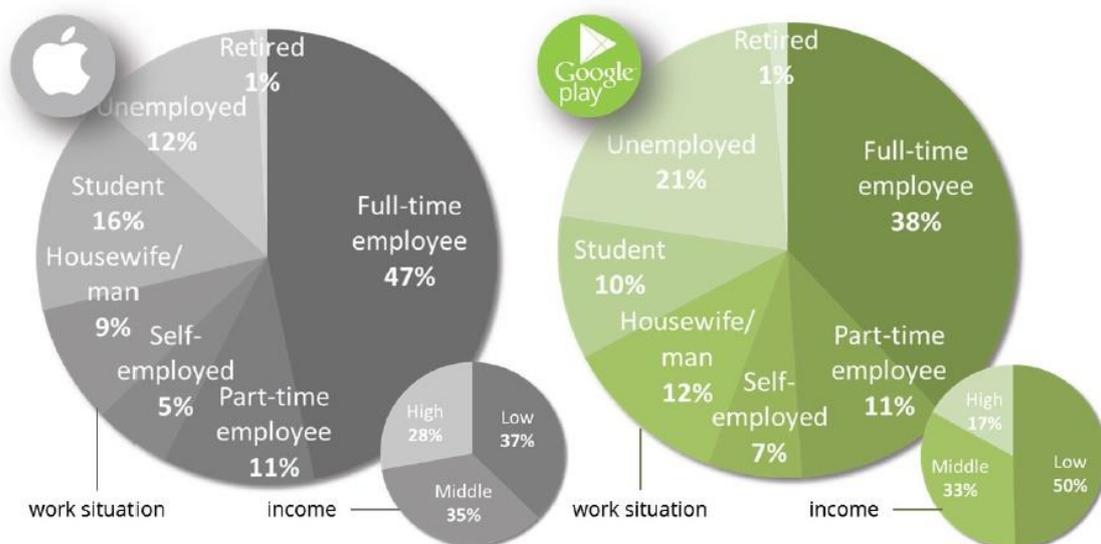


Figura 15. Estudio demográfico de usuarios de iOS y Android [1]

Aplicaciones específicas para tabletas

Dentro de las ventas de dispositivos móviles, los Floating Screen Devices (tabletas y consolas de mano) ganan cada vez más importancia en relación con ordenadores de sobremesa y portátiles, y generan un creciente volumen de beneficios.

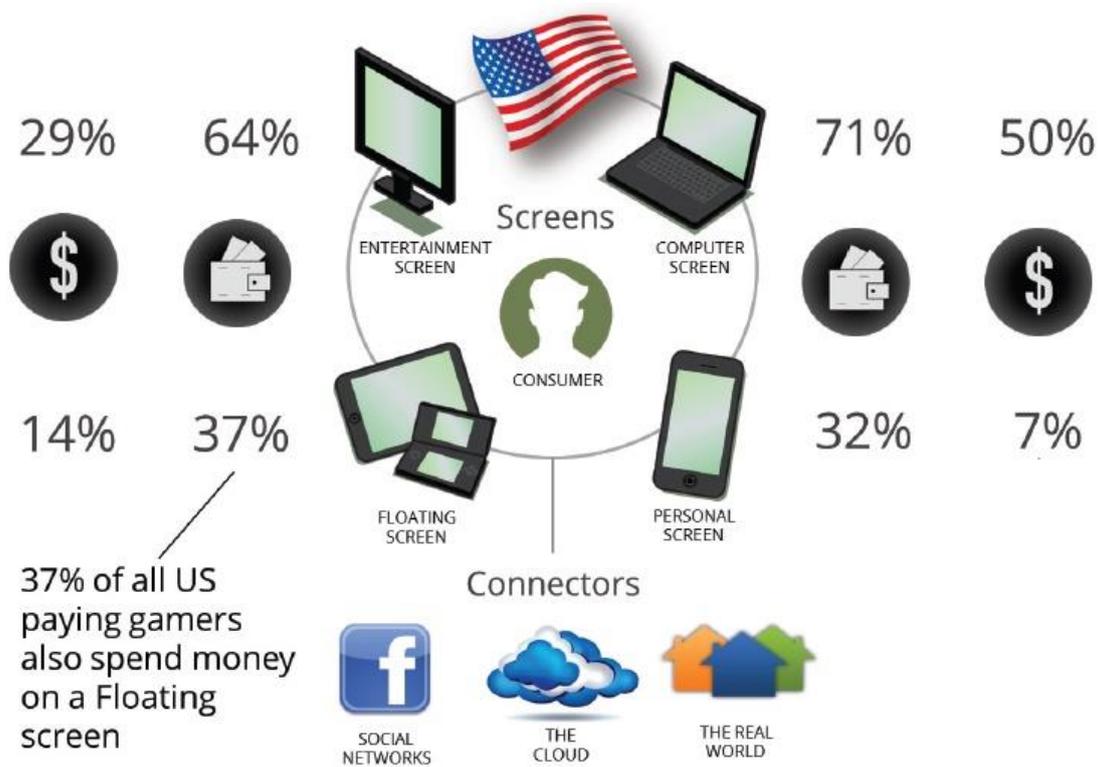


Figura 16. Distribución de ingresos por tipo de dispositivo [1]

Generalmente, el usuario de tableta es también usuario de Smartphone, pero gasta dinero en aplicaciones en ambos dispositivos. Se puede entender por tanto, que cada vez más usuarios dedican sus Smartphones a la comunicación y adquieren una tableta para el entretenimiento.

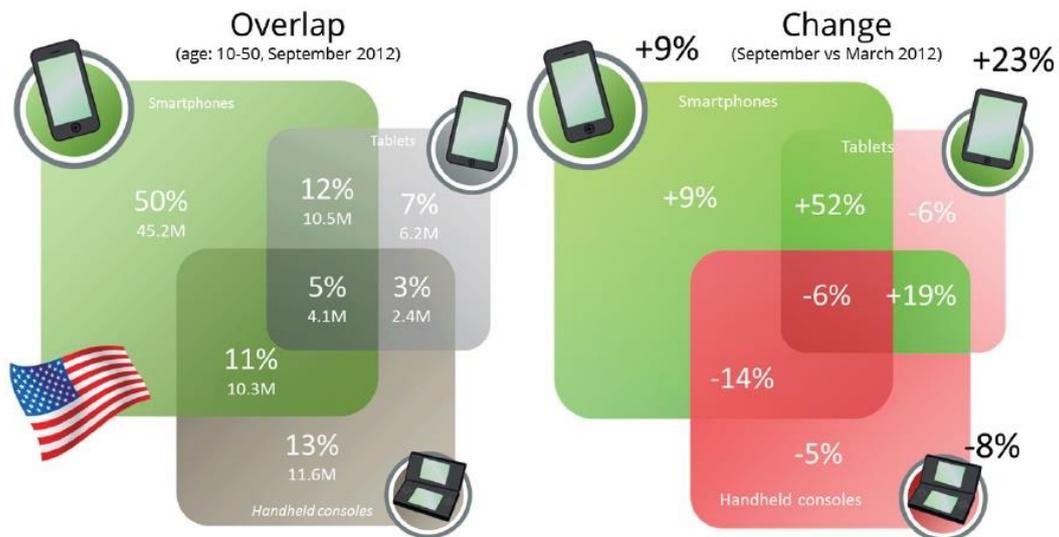


Figura 17. Porcentajes por tipo de dispositivo [1]

Por el tamaño y potencia de las tabletas en comparación con los Smartphones, desarrollar juegos específicamente para ellas aporta ventajas adicionales. Aun siendo un mercado menor, es una plataforma más orientada para que el cliente disfrute de este tipo de aplicaciones, y analizando la previsión de adquisiciones de dispositivos para los próximos años, se hace cada vez más evidente la clara predilección por parte de los usuarios de disponer de una tableta además de un Smartphone.

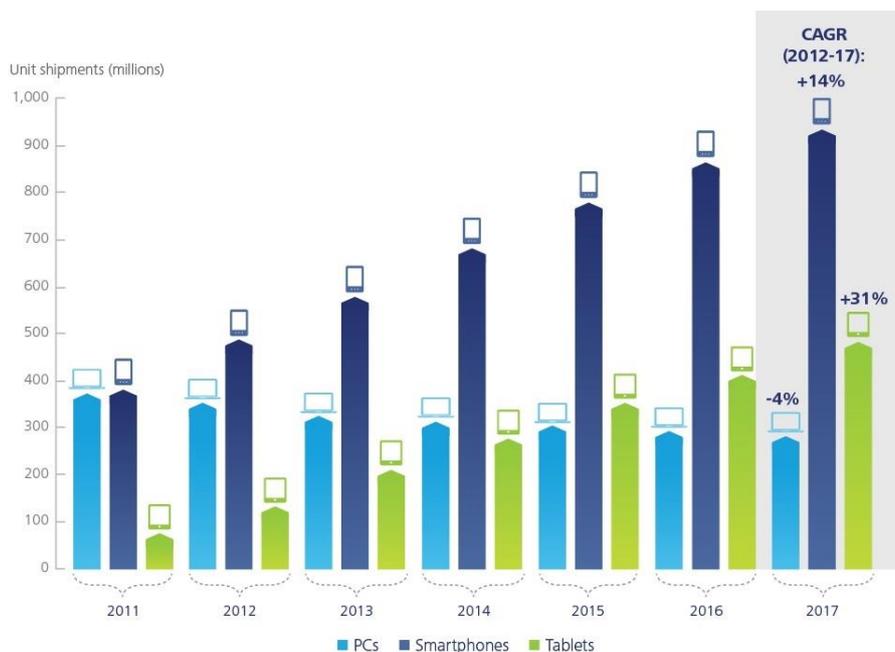


Figura 18. Predicción de ventas de Portátiles, Smartphones y Tabletas [2]

2.3.2 Situación del género Tower Defense

Como se ha citado en la introducción al apartado sobre la situación general del mercado, Tower Defense está clasificado en los mercados de juegos para dispositivos móviles como un sub-género de la estrategia, y goza de mucho seguimiento.

Descripción del género

La mecánica de este tipo de juegos consiste en que el jugador, con visión completa de un escenario determinado, debe conseguir defenderse de oleadas de enemigos que recorren dicho escenario a través de un camino definido y visible. Éstos enemigos no atacan a ningún personaje ni estructuras que maneje el jugador, simplemente recorren el camino definido de un punto inicial a otro final, y si una cantidad establecida de ellos alcanza el punto final, la partida se da por perdida. El objetivo para el jugador es, por tanto, aguantar el número de oleadas que para cada pantalla del juego se establezca, sin que la cantidad límite de enemigos haya recorrido el camino completo.

Origen

El Tower Defense es un género relativamente joven dentro de la aún no muy prolongada historia de los videojuegos. Tras el lanzamiento del juego de estrategia (RTS) Warcraft III, de la compañía Blizzard Entertainment, se convirtió en una práctica habitual para los jugadores, emplear un profundo editor de niveles personalizados que el juego incluye, para crear mini-juegos con mecánicas simples y adictivas con el fin de compartirlas con la comunidad que conformaban con el resto de jugadores. Estos mini-juegos se ejecutaban con el motor del propio Warcraft III, con lo que los jugadores que querían crear mini-juegos sólo tenían que preocuparse del diseño de la mecánica, empleando modelos 3d y programación ya existente.

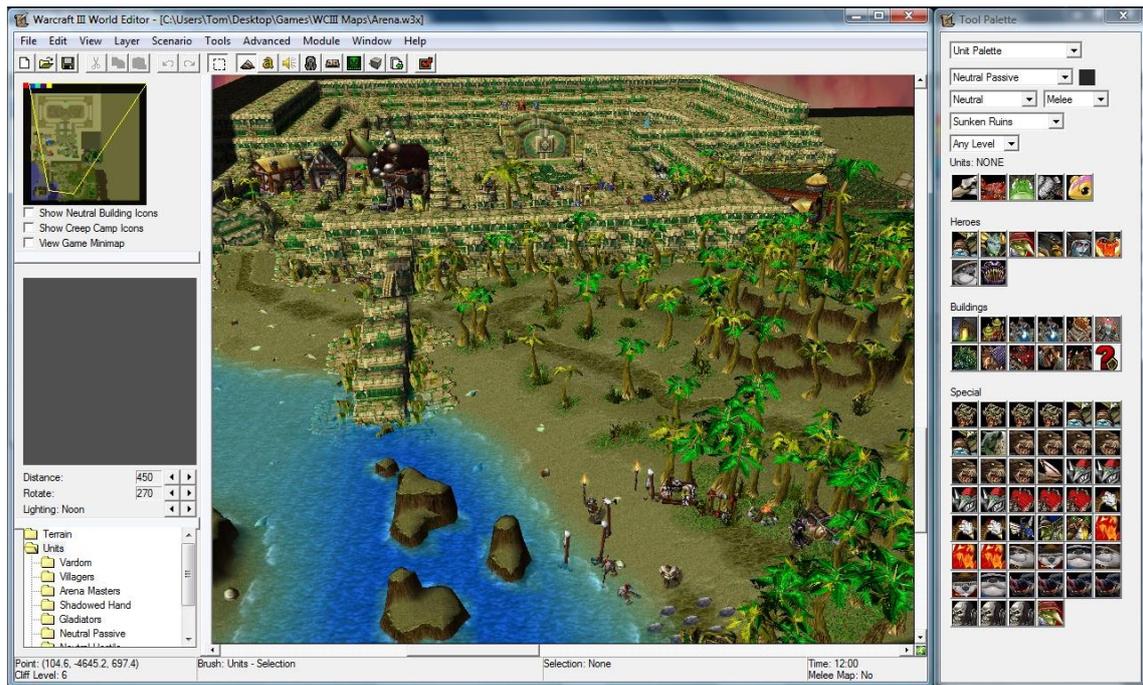


Figura 19. Editor de niveles de Warcraft III [a.10]

Aunque el hecho de incluir editores de niveles dentro de un juego para el uso de los jugadores se había dado en juegos anteriores, la fama que alcanzó Warcraft III hizo que esta herramienta añadida se convirtiera en fuente de inspiración para miles de personas, que explotando al máximo las posibilidades del editor y haciendo buen uso de su imaginación, hicieron que vieran la luz multitud de nuevos géneros de videojuegos que actualmente disponen de títulos creados por las más importantes empresas del mundo.

Uno de los géneros surgidos a partir del editor de niveles de Warcraft III, es el Tower Defense. Empresas grandes y pequeñas han trabajado en este género y se pueden encontrar infinidad de títulos para cualquier plataforma.

2.3.3 Estudio de la competencia

Por la simplicidad de su mecánica de juego, trabajar en el género Tower Defense se ha convertido en un arma muy útil para que empresas primerizas se den a conocer. A continuación se exponen, describen y analizan los puntos fuertes de los principales Tower Defense del mercado.

Kingdom Rush

No fue el primer título destacable, pero sí ofreció un salto de calidad muy importante, haciendo que el género, que principalmente se centraba en plataformas web, empezase a ofrecerse para dispositivos móviles. Hoy cuenta con cerca de un millón de descargas en Android e iOS.



Figura 20. Kingdom Rush [a.11]

Las claves de su éxito fueron un gran trabajo de diseño, ofreciendo una curva de aprendizaje muy agradable para el jugador, la introducción de un buen número de características nunca antes vistas, como el manejo de soldados y héroes o la opción de mejorar características de las torres a medida que avanza la historia, y la estética, muy apropiada para el tipo de jugador amante del género.

Plants vs Zombies

Aunque no fue ideado para tabletas, su éxito en PC garantizó un traslado rentable a plataformas móviles. Se trata de un clásico del Tower Defense, que plantea una perspectiva diferente para el jugador, situando a los enemigos en diferentes caminos que se defienden independientemente.



Figura 21. Plants vs Zombies [a.12]

Cuando un género está tan asentado como el Tower Defense, es difícil proponer un cambio tan significativo y que sea tan rápidamente aceptado por la comunidad de jugadores. Con una estética original y colorida, este juego ha conseguido convertir su nuevo planteamiento o forma de jugar, en una vertiente que muchas compañías ya siguen dentro del género.

Jelly Defense

Se trata del Tower Defense más orientado a destacar en el aspecto audiovisual. Aunque no propone nuevas características, la calidad de los gráficos y el sonido lo convierten en uno de los más destacados.



Figura 22. Jelly Defense [a.13]

Es un ejemplo de que se pueden conseguir buenos resultados potenciando un aspecto en concreto del juego, especialmente si el producto en conjunto está asentado en una buena base. En un Tower Defense, que ya tiene un público garantizado por los seguidores del género, conseguir una calidad gráfica y de sonido tan alta, simplificando la jugabilidad, puede abrir además puertas de nuevos usuarios que acudan por el impacto audiovisual conseguido.

Field Runners

Field Runners es un ejemplo de cómo explotar al máximo el concepto en el que se basa el género Tower Defense, la satisfacción de destruir el mayor número de enemigos posibles con gran cantidad de armas. Este juego permite al jugador disponer de un arsenal inmenso de torres y trampas que puede colocar libremente en el campo de batalla.



Figura 23. Field Runners [a.14]

Empleando gráficos renderizados a partir de animaciones 3d, consigue una estética tridimensional aun tratándose de un juego en 2d. La estética está muy cuidada, y el principal punto a favor de este juego es la libertad con la que el jugador puede colocar torres en el escenario, definiendo él mismo el camino que deben seguir los enemigos.

Bloons Tower Defense

Esta saga cuenta ya con 5 títulos y está orientada a plataformas web, donde comenzó como uno de los primeros títulos del género y, sin ofrecer gran calidad gráfica, obtuvo gran cantidad de seguidores en su comienzo, que ahora lo clasifican como un clásico.



Figura 24. Bloons Tower Defense 5 [a.15]

Al contrario que Jelly Defense, Bloons TD deja al margen el apartado audiovisual para, con unos gráficos mediocres, centrar su atractivo en la jugabilidad, proponiendo multitud de nuevas opciones e infinidad de pantallas y retos para los jugadores en cada versión de la saga.

Pirate Legends TD

Este título demuestra que una compañía con recursos puede invertir en el género y obtener beneficios. No aporta ninguna novedad específica, sólo emula a los principales títulos, ofreciendo una calidad más que aceptable en cuanto a gráficos, sonido y jugabilidad, y se divulga gracias a una inversión importante en marketing.



Figura 25. Pirate Legends TD [a.16]

Este juego concluye el estudio de la competencia, que abarca todos los juegos del género Tower Defense que han logrado cierto éxito en cuanto al número de jugadores y descargas.

Cada cual ha sabido explotar un aspecto determinado del género en su propio beneficio, consiguiendo, unos más que otros, impactantes resultados y en general productos de mucha calidad sin un esfuerzo demasiado grande en el desarrollo.

2.3.4 Perspectivas de impacto

Es difícil predecir el impacto en descargas que va a tener una aplicación para dispositivos móviles, especialmente si no se va a promocionar con ninguna campaña de marketing. El número de juegos disponibles totales y publicaciones diarias crece cada día.

Analizando los datos estadísticos de los que se dispone, se puede definir el perfil de un desarrollador independiente medio y basar en él las perspectivas de impacto que puede tener un juego de un desarrollador de perfil similar.

Los datos asociados al perfil citado se muestran en la siguiente figura. Teniendo en cuenta que siempre ofrecería juegos gratuitos, las estimaciones de beneficios se calculan en función de lo que podría obtener añadiendo publicidad.

	Google	Apple
Apps per developer	5	5
Download per app	60,000	40,000
Revenue per download	\$0.01	\$0.10

Figura 26. Expectativas de descargas estándar [3]

Teniendo en cuenta estos datos, y que el juego desarrollado en este proyecto en principio sólo estará disponible para tabletas, se puede anticipar un impacto teórico de 10.000 descargas incluyendo las de las plataformas iOS y Android. Se puede predecir también que la mayoría de ellas se producirían en los primeros meses, por clasificarse el juego dentro de los mercados como nueva publicación.

Hay que tener en cuenta también, que el juego está clasificado en el género de la estrategia, lo que supone una ventaja adicional a la hora de divulgarse, ya que este tipo de juegos disponen de un público mayor, aunque no especialmente leal en cuanto al tiempo que mantiene el juego instalado en sus dispositivos.

Mobile Game Loyalty Matrix

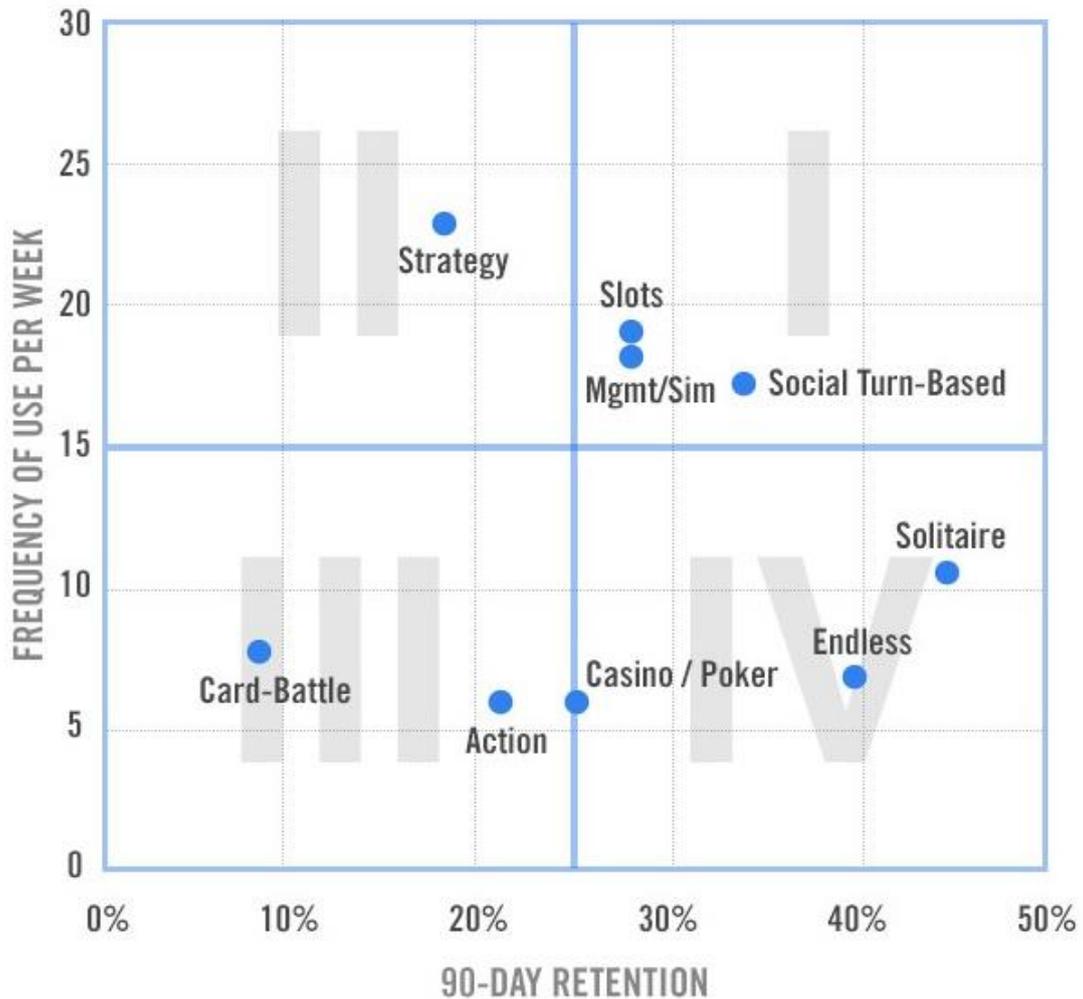


Figura 27. Lealtad de usuarios según el género del juego [4]

Se concluye que las perspectivas de impacto que se pueden establecer para el juego en el que se trabaja en este proyecto, pueden ser relativamente pobres en relación al tamaño total del mercado en el que va a ofrecerse. Sin embargo, se podría conseguir una cantidad de jugadores más que aceptable para establecer una base sobre la que buscar nuevos medios de promoción, teniendo en cuenta que la calidad general del juego en relación con la de su competencia será lo suficientemente alta como para atraer al público.

2.3.5 Marco regulador

Aun teniendo en cuenta que estrictamente el alcance de este proyecto no abarca la explotación de la aplicación, el próximo paso a seguir sería la publicación en mercados Android e iOS.

Por lo tanto, es obligatorio para el desarrollador conocer y cumplir con los términos y condiciones del contrato de las compañías dueñas de los mercados citados, es decir, la legislación aplicable al proyecto.

De este modo, cuando el cliente reciba el producto final, podrá publicarlo en estos mercados sin encontrar objeciones.

Los aspectos más importantes a cumplir de los acuerdos de distribución para desarrolladores son los siguientes.

Google Play

- Están prohibidas las aplicaciones que incluyan material sexual implícito.
- No se permite la publicación de imágenes que muestren violencia gratuita.
- No se permite publicar contenido que fomente el odio hacia los miembros de un grupo por su raza u origen étnico.
- No se debe sustituir la identidad de otras personas.
- No se deben infringir los derechos de propiedad intelectual de terceros.
- No se permite la divulgación de información de carácter privado y confidencial de los usuarios.
- No se permiten las actividades ilegales.
- No se permiten servicios que promuevan los juegos de azar online.

Apple App Store

Las condiciones de Apple App Store incluyen todas las citadas para Google Play, y adicionalmente se ha de cumplir con las siguientes:

- La aplicación debe estar libre de “bugs” y funcionar correctamente.
- La aplicación debe ser “útil” u “ofrecer un entretenimiento duradero” al usuario.
- No se aceptarán versiones de aplicaciones en fase de prueba.
- No se aceptarán copias de aplicaciones que ofrezcan un servicio ya cubierto por muchas otras.
- La aplicación no puede solicitar datos personales del usuario.

- Se debe utilizar una API de programación pública.
- No se puede incluir texto falso.
- En el momento de subir la aplicación, deben estar listas las páginas de soporte y de descripción de política de privacidad de la misma.
- No se puede imitar la interfaz de usuario de iPhone, iPod o iPad.
- La interfaz debe ser comprensible para el usuario.
- No se pueden desbloquear o permitir nuevas funcionalidades con mecanismos que no sean puramente del App Store.

Como se puede ver, las condiciones del mercado de Apple son mucho más restrictivas, por lo tanto, serán el punto de referencia.

2.3.6 Diagnóstico del estudio de la situación actual

Se hace evidente que el mercado ofrece muchas posibilidades a los juegos del género Tower Defense. Al tratarse de un estilo de juego tan específico, el cliente final del producto espera cierta fidelidad al género y valora la originalidad en detalles concretos o pequeños complementos. Se espera de ellos un nivel mínimo de calidad, ya que las reglas básicas están establecidas y hay una pequeña tradición de compromiso en los desarrolladores de este género.

La dinámica de dos jugadores simultáneos en modo batalla que sugiere el cliente, no se observa en los principales Tower Defense del mercado, pero mantiene el concepto básico del juego y, aunque no permite a un jugador jugar en solitario, se puede considerar como una opción que la comunidad aceptaría.

Limitar el juego a un solo escenario (pantalla de juego), evita una gran inversión de tiempo en diseño, lo que facilita dedicar tiempo de desarrollo a obtener un resultado final de calidad.

En definitiva, se establece que la solicitud del cliente es un proyecto viable y que encaja perfectamente en el mercado al que pretende dirigirse, aportando además, un nivel de originalidad único con respecto a las opciones que ofrece la competencia.

2.4 Entorno socio-económico

Como se ha expuesto en los apartados anteriores, las estrategias de explotación para juegos en plataformas móviles dependen de diversos factores, como el propio entorno en el que se ha desarrollado, o la disposición de recursos de distribución.

En el caso de este proyecto, ya que se parte de unos recursos muy limitados para distribuir o promocionar el juego, se va a planificar una estrategia que se iniciará con la búsqueda de un Publisher (empresa que se encarga de la distribución de juegos a cambio de un porcentaje de los beneficios), y que planteará una vía de explotación alternativa de bajo coste en caso de no conseguir un acuerdo de distribución convencional.

2.4.1 Búsqueda de Publisher

Existe una cantidad importante de empresas de este tipo a nivel global, y todas trabajan internacionalmente. A la hora de escoger a qué Publishers enviar un juego para negociar un acuerdo, es importante conocer qué profundidad, nivel de complejidad y calidad tienen los productos con los que trabajan. Empresas como [Chillingo](#) o [Rovio Stars](#) nunca mostrarán interés en juegos que no tengan el máximo potencial de éxito, por lo que en este caso se ha de enfocar la búsqueda hacia Publishers más interesados juegos de nivel medio.

Para esto, existe una plataforma destinada a poner en comunicación a desarrolladores independientes de juegos de calidad intermedia con empresas distribuidoras interesadas en este tipo de juegos. Esta plataforma es [FGL](#).

FGL

El servicio que ofrece FGL es el de evaluar, clasificar y alojar los juegos de los desarrolladores antes de ser publicados, para facilitar su búsqueda por parte de los Publishers registrados en la plataforma, que puján por los juegos para conseguir el mejor acuerdo posible.

Figura 28. Página principal de www.fgl.com

Una vez que el desarrollador acepta el precio y condiciones del contrato de la mejor oferta que propone un Publisher determinado, FGL también se encarga de facilitar la distribución del producto en diferentes mercados con la integración de una API que muestra anuncios dentro del juego, y reparte los beneficios obtenidos entre el desarrollador y su Publisher.

Se puede decir que en este caso los Publishers hacen más bien el papel de patrocinadores, invirtiendo un capital inicial en el juego apostando por su futura rentabilización mediante los beneficios obtenidos con anuncios, ya que FGL hace la mayor parte del trabajo de distribución.

Procedimiento para participar

Para optar a conseguir un acuerdo con un Publisher en FGL estos son los pasos que se han de seguir:

- Crear una cuenta de desarrollador en FGL.com.
- Solicitar acceso a la API adecuada para el entorno en el que se trabaja, en este caso Flash.
- Integrar la API en el juego de forma que muestre la interfaz de anuncios.

- Subir el archivo ejecutable del juego a FGL con la API integrada y el material de marketing necesario.

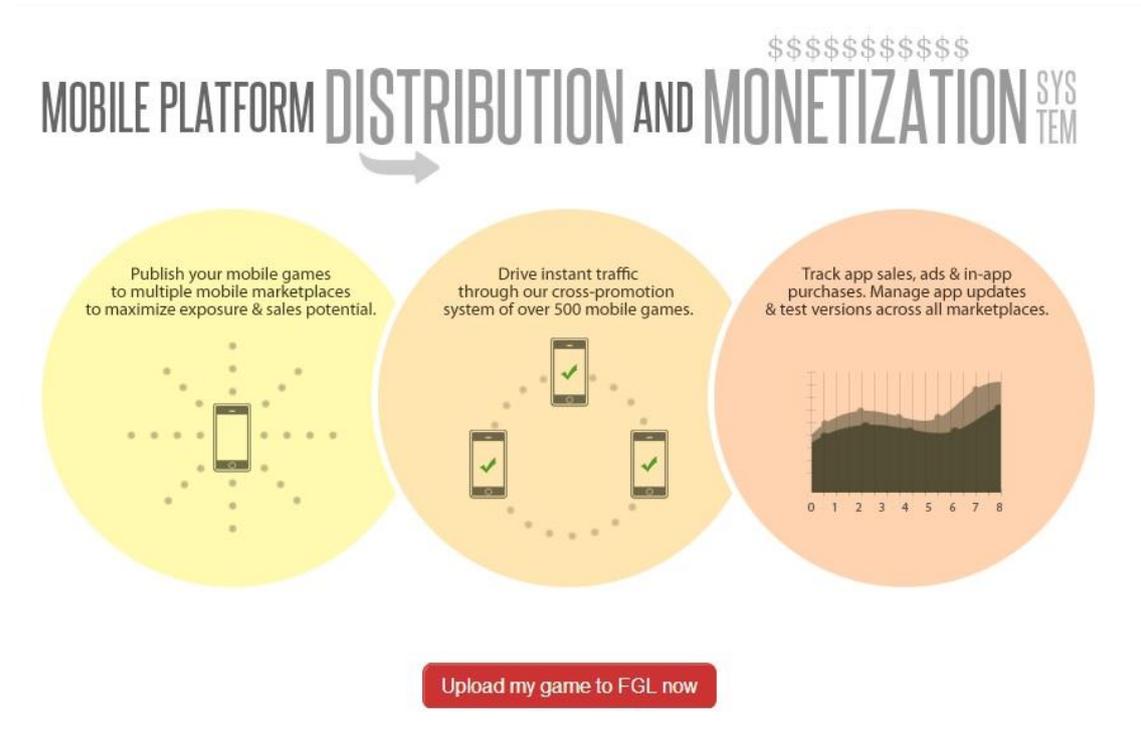


Figura 29. Subir juego a www.fgl.com

Una vez completado este procedimiento, FGL evaluará el juego y lo publicará para visualización privada de los Publishers. Se ha de esperar una media de 15-30 días para estudiar el interés que puedan mostrar.

Selección

Si se produjeran ofertas, se estudiarán valorando si cubren económicamente el coste que ha conllevado la producción del juego (teniendo en cuenta que FGL se lleva una cuota del 10% del precio total de la venta), y el porcentaje que el Publisher exige de los beneficios por publicidad.

2.4.2 Publicación independiente alternativa

En caso de no llegar a un acuerdo con un Publisher, al estar tratando el caso de un juego con dificultades de distribución, la solución óptima para encontrar mayor impacto en el número de descargas es hacer uso de la plataforma de distribución FGL prescindiendo del contrato con un Publisher.

El inconveniente de este caso es que no se dispondrá de beneficios directos en el momento de publicar el juego, sin embargo, se recibirán el 90% de los beneficios obtenidos por publicidad mediante la API de FGL.

2.5 Alternativas de diseño

Conociendo la idea principal del producto que se desea desarrollar, el mercado y plataformas en las que va a publicarse, es necesario estudiar las posibles alternativas que permiten llegar a la solución del problema para seleccionar la más adecuada. Para cada una de las alternativas, se analizarán los siguientes elementos.

Versatilidad del entorno

Existen entornos de desarrollo tanto específicos para sistemas operativos Android e iOS, como multiplataforma. Se estudiarán las capacidades específicas de cada uno.

Lenguaje de programación

Cada entorno de desarrollo emplea un lenguaje de programación diferente. Se analizará si es viable programar en el lenguaje específico requerido.

Dispositivo objetivo

Además de ofrecer la posibilidad de publicar un juego para un sistema operativo concreto, el entorno puede hacerlo de forma más o menos eficiente o con diferente grado de compatibilidad.

Coste

Emplear un entorno o software determinado para la elaboración de una aplicación, suele implicar un gasto. La mayoría de los entornos de desarrollo han de comprarse o bien exigen un pago concreto para la publicación en determinados sistemas operativos. También se analizará el coste en tiempo con su correspondiente impacto económico que conlleva cada alternativa (aprendizaje, investigación...).

2.5.1 Alternativa 1. Aplicaciones nativas

La primera alternativa está ideada para llegar a un resultado óptimo en cuanto a la estabilidad del producto final, al margen de un posible aumento del coste de recursos necesarios para llegar a la solución. Para ello se plantea el desarrollo del juego de forma nativa tanto para Android como para iOS.

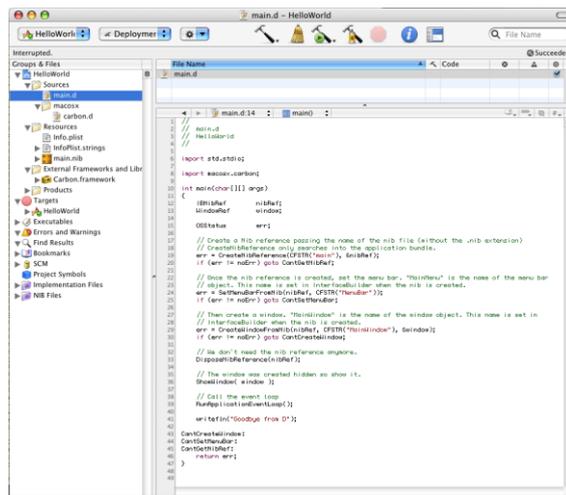


Figura 30. Xcode para Mac OS X [a.17]

Versatilidad del entorno

Desarrollar aplicaciones nativas para dos sistemas operativos conlleva un esfuerzo adicional, ya que requiere programarla una vez para cada sistema usando diferentes herramientas en cada caso. En este sentido, la alternativa 1 plantea un importante obstáculo, sobre todo teniendo en cuenta que Xcode, el entorno específico de desarrollo nativo para iOS, requiere del uso de un ordenador con sistema operativo Mac OS X.

Lenguaje de programación

Las aplicaciones nativas para Android se programan en Java, empleando el SDK de Android, que incluye diferentes herramientas de desarrollo, como un depurador de código y un simulador de teléfono o Tablet.

En el caso de iOS, se programa en Objective-C, que es un lenguaje orientado a objetos basado en C empleado en el citado entorno Xcode.

Tanto Java como Objective-C son lenguajes modernos y es relativamente fácil profundizar en ellos.

Dispositivo objetivo

Crear una aplicación nativa permite hacer uso de la mayor parte de las funcionalidades del dispositivo al que va dirigida, ya que el lenguaje y el entorno de desarrollo están creados específicamente para ello. Este es en la mayoría de los casos, el principal motivo que lleva a un desarrollador a elegir esta alternativa. En ningún otro caso se pueden conseguir resultados más fiables a la hora de manejar las características de un Smartphone o tableta.

Coste

Por el propio interés de los creadores de los diferentes sistemas operativos, los entornos y herramientas de desarrollo de aplicaciones nativas son gratuitos. Sin embargo, desarrollar un juego nativo para iOS requiere el empleo de un ordenador con sistema operativo Mac OS X, y puesto que no se dispone de uno para este proyecto, su adquisición supondría un coste adicional elevado.

Esta alternativa también supone un esfuerzo adicional en relación a las horas necesarias por parte del programador para desarrollarlo, ya que debe hacerlo dos veces.

2.5.2 Alternativa 2. Flash CS6

La segunda alternativa está planteada buscando una buena relación entre versatilidad, coste y fiabilidad. Empleando un entorno y lenguajes ya conocidos para buscar mayor eficiencia. Para ello se plantea el entorno Flash CS6.

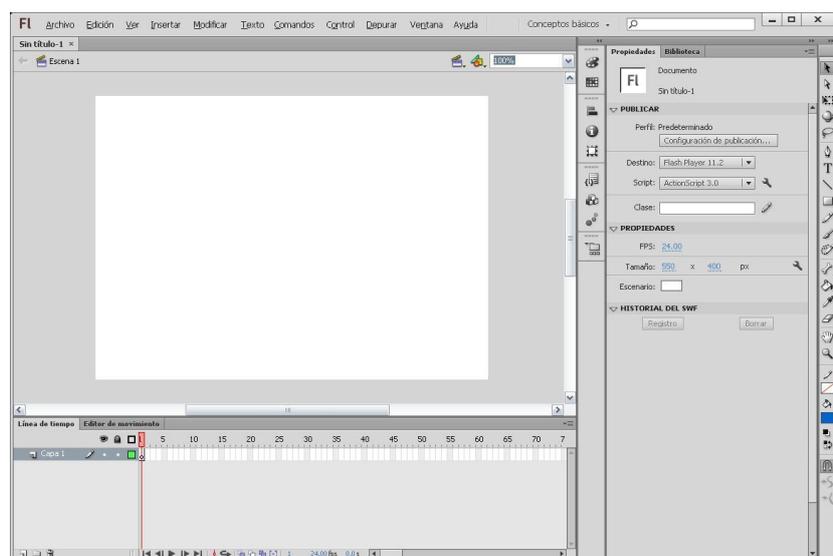


Figura 31. Flash CS6 [a.18]

Versatilidad del entorno

Gracias a AIR, el entorno de ejecución multiplataforma de Adobe, Flash ha pasado de ser una herramienta para la elaboración de páginas web y aplicaciones de escritorio, a convertirse en uno de los principales recursos en la elaboración de juegos para plataformas móviles.

AIR permite al desarrollador de Flash CS6, publicar una misma aplicación tanto para sistemas Android e iOS, como para web y escritorio. Por este motivo, en cuanto a versatilidad, esta alternativa es una opción más que viable.

Lenguaje de programación

El lenguaje empleado en el entorno Flash CS6 es Actionscript 3, un lenguaje orientado a objetos específico de Adobe que guarda muchas similitudes con Java. Al ser un lenguaje que ya se conoce, no requiere esfuerzo adicional de investigación por parte del programador.

Dispositivo objetivo

Las aplicaciones publicadas desde Flash CS6 para dispositivos móviles llevan incorporadas el motor de AIR para ejecutarse. Esto implica que la aplicación va a pesar unos 5-10 Mb extra. En este caso, al tratarse del desarrollo de una aplicación para tabletas, que disponen de discos duros de más capacidad que un Smartphone, ese exceso de peso no supone un problema.

En cuanto a estabilidad, hay que hacer un esfuerzo especial a la hora de programar y gestionar las imágenes para que el juego funcione correctamente.

Coste

Gracias al sistema de pago mensual de Adobe, publicar legalmente una aplicación en entornos Flash conlleva un gasto económico bajo.

El software de Flash CS6 está disponible para Windows 32-64 bits, y ya se dispone de él.

2.5.3 Alternativa 3. Unity3d

La tercera alternativa se plantea como una inversión de cara al futuro. Unity3d es un entorno muy joven que está teniendo mucho éxito en el campo del desarrollo de juegos para dispositivos móviles. Está orientado a juegos, y

aunque está construido para manejar modelos 3d, dispone de herramientas para desarrollar también juegos 2d como la mayoría de los Tower Defense.

Construir el juego en este entorno conlleva un riesgo adicional porque no se conoce aún con exactitud si Unity3d es un proyecto estable y sostenible, y no hay garantías de disponer del soporte apropiado, pero si se confía en que se asentará como un entorno estable, es muy interesante disponer de un juego desarrollado en él para posibles actualizaciones futuras.



Figura 32. Unity3d [a.19]

Versatilidad del entorno

En principio el entorno es muy versátil, y al igual que Flash CS6, permite la publicación para web, escritorio, Android e iOS.

Lenguaje

En Unity3d, se permite programar tanto en C# como en Javascript, a gusto del programador, ya que ninguno de ellos implica ninguna limitación. Se trata de lenguajes conocidos con mucha documentación disponible.

Dispositivo objetivo

Se ha comprobado que aplicaciones desarrolladas en Unity3d ofrecen un rendimiento correcto tanto en Android como en iOS.

Coste

Aun siendo una herramienta gratuita, se exige el pago de una licencia para desarrollar en iOS. Puede pagarse mensualmente, pero supone un gasto económico algo mayor que el de Flash CS6 y no se dispone de ella actualmente.

Habría que emplear tiempo adicional para aprender a manejar el entorno, ya que no se ha visto antes, e investigar el lenguaje apropiado para programar.

2.5.4 Selección de la mejor alternativa de diseño

Tras el estudio de las diferentes alternativas de solución, se valoran de 1-3 los parámetros escogidos para el análisis de las mismas. Se otorga una nota para cada parámetro de cada alternativa relativa al resto de opciones, y se escoge la que obtiene un resultado mayor.

	Versatilidad	Lenguaje	Dispositivo	Coste
Nativa	1	1	3	1
Flash CS6	2	3	2	3
Unity3d	3	2	1	2

Tabla 1. Estudio de alternativas de solución

Resultado

Alternativa 1. Aplicaciones nativas: **6**.

Alternativa 2. Flash CS6: **10**.

Alternativa 3. Unity3d: **8**.

Se selecciona por tanto Flash CS6 como alternativa óptima para este caso concreto, ya que ofrece la versatilidad que se busca en un lenguaje ya conocido por el programador, con un coste bajo y buenas garantías de rendimiento.

3 Gestión del proyecto

El objetivo de este proyecto es la implementación de un videojuego a petición de un cliente. Como en cualquier proyecto de desarrollo de software, para tener la mayor garantía de cumplir el objetivo, es necesario ceñirse a metodologías de trabajo establecidas y dedicar un esfuerzo inicial a definir el ciclo de vida del software.

3.1 Ciclo de vida

El modelo de ciclo de vida de un proyecto especifica el orden en el que se estructuran las tareas que componen su desarrollo completo. De las posibles opciones de ciclos lineales o secuenciales, para este proyecto se ha decidido escoger el **modelo en cascada retroalimentado**.

3.1.1 Modelo en cascada retroalimentado

Este tipo de ciclo de vida parte del modelo tradicional de desarrollo en cascada, que se denomina así por ordenar sus etapas de manera secuencial. Es un modelo óptimo para un desarrollo ágil y rápido, pero con el inconveniente de que retroceder de una etapa a otra anterior es muy costoso. Es un modelo ideal para proyectos muy bien definidos desde el primer momento y que se van a ver sujetos a pocas variaciones.

El modelo en cascada retroalimentado reduce los riesgos de que el desarrollo se vea obligado a retroceder varias etapas, porque en cada una de ellas se realiza una evaluación, para poder añadir retroalimentación entre etapas e incluso poder alterar o añadir algún requisito aun habiendo avanzado en la fase de diseño.

Este modelo se considera adecuado para este proyecto ya que, además de disponer de requisitos claros y muy específicos, el apartado artístico (gráficos y sonidos), ya está cubierto por su disponibilidad en el juego del que se parte, así que casi la totalidad del proyecto la realiza una sola persona, el programador.

3.1.2 Definición de fases del modelo

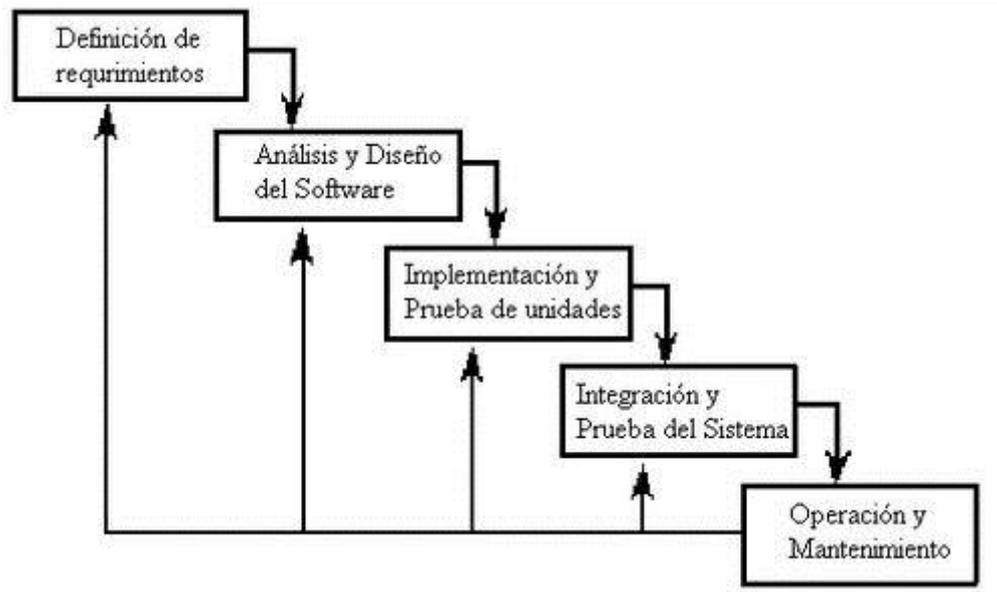


Figura 33. Modelo de ciclo de vida en cascada retroalimentado [11]

Análisis de requisitos

Estudiando la petición del cliente y las necesidades de los usuarios, se especifican los objetivos y se enumeran como requisitos.

Diseño del sistema

Se especifica la arquitectura del sistema, convirtiendo el análisis de requisitos en un diseño más complejo y detallado en el que se define en profundidad la estructura de la aplicación.

Implementación

Partiendo del diseño del sistema, se desarrolla finalmente el código del software para convertirlo en una aplicación real.

Pruebas

Se realizan pruebas de la aplicación para evaluar su correcto funcionamiento y que cumple los requisitos del cliente.

Mantenimiento

Durante esta fase, que puede extenderse todo lo que solicite el cliente, se corrigen los errores descubiertos y se añaden pequeñas mejoras o nuevos requisitos.

3.2 Organización del proyecto

3.2.1 RBS

RBS son las siglas de Resource Breakdown Structure (estructura jerárquica de recursos). En esta lista se enumeran los recursos humanos y materiales necesarios para la planificación y desarrollo del proyecto.

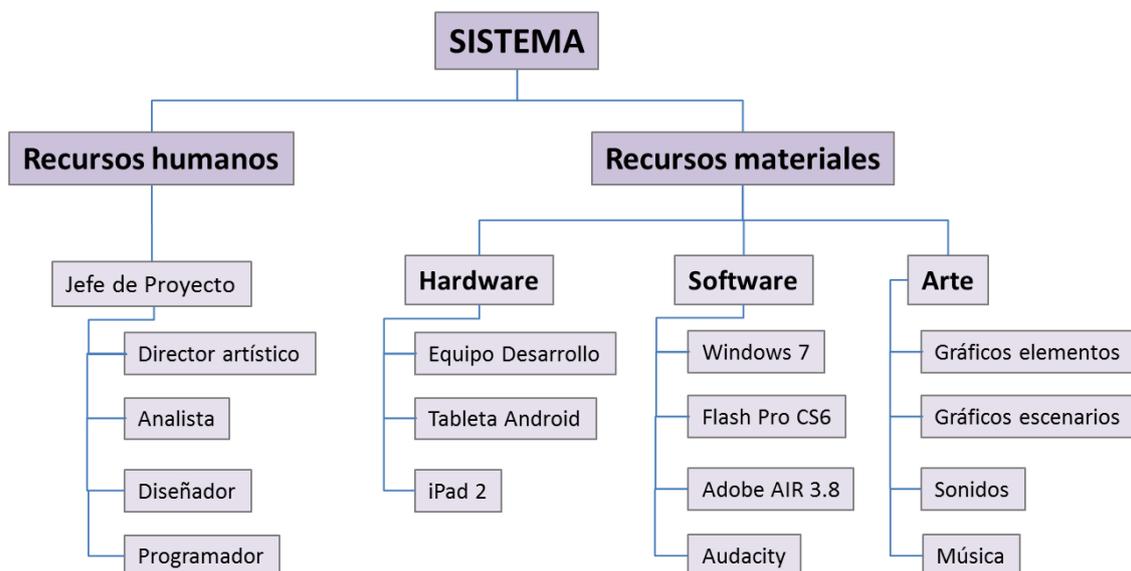


Figura 34. RBS del proyecto

3.2.2 Responsabilidades de los recursos humanos

Según se expresa en la RBS del proyecto descrita en el punto anterior, para este proyecto se emplean diferentes recursos humanos. A continuación se define el rol específico de cada uno.

Jefe de proyecto

Se encarga de la gestión y organización del proyecto. Es su responsabilidad que los trabajos se asignen entre los diferentes miembros del equipo de manera que se optimice el esfuerzo que realizan.

Analista

Su función es la de identificar y definir los requisitos de usuario a través de la comunicación con el cliente. Dando así, a los diseñadores, una visión clara de las necesidades a cubrir por el sistema.

Diseñador

Lleva a cabo la definición de modelos que permitan al programador desarrollar el sistema de tal manera que éste satisfaga las necesidades del cliente.

Programador

Se encarga de la codificación del sistema partiendo de los modelos definidos por el diseñador.

Director artístico

Supervisa la coherencia artística en el aspecto audiovisual de la aplicación durante todo el proyecto.

Al tratarse de un proyecto académico, al igual que el software final se desarrolla para un cliente hipotético, estos roles habrían de ser desempeñados por hipotéticos miembros de un equipo de trabajo.

3.3 Planificación

El proyecto se inicia el jueves 15/01/13, y su duración se prolonga hasta el martes 04/06/13, completándose en 101 días laborales (alrededor de cinco meses reales). Como el proyecto se realiza durante el curso académico, no se puede dedicar la totalidad de la jornada laboral al desarrollo del mismo. Cada día de trabajo corresponde aproximadamente 2 horas de dedicación.

En el siguiente diagrama de Gantt se muestran las diferentes fases del proyecto y su distribución en el tiempo.

Trabajo de Fin de Grado		101 days	Tue 15/01/13	Tue 04/06/13	
Estudio de Viabilidad del Sistema		26 days	Tue 15/01/13	Tue 19/02/13	
	Establecimiento del alcance del sistema	10 days	Tue 15/01/13	Mon 28/01/13	Jefe de proyecto
	Estudio de la situación actual	8 days	Tue 29/01/13	Thu 07/02/13	Jefe de proyecto
	Estudio de alternativas de diseño	8 days	Fri 08/02/13	Tue 19/02/13	Jefe de proyecto
Gestión del Proyecto		13 days	Wed 20/02/13	Fri 08/03/13	
	Ciclo de vida	4 days	Wed 20/02/13	Mon 25/02/13	Jefe de proyecto
	Organización del proyecto	3 days	Tue 26/02/13	Thu 28/02/13	Jefe de proyecto
	Planificación	4 days	Fri 01/03/13	Wed 06/03/13	Jefe de proyecto
	Estimación de costes	2 days	Thu 07/03/13	Fri 08/03/13	Jefe de proyecto
Desarrollo de la aplicación		51 days	Mon 11/03/13	Mon 20/05/13	
	Análisis	10 days	Mon 11/03/13	Fri 22/03/13	Analista; Director artístico
	Diseño	6 days	Mon 25/03/13	Mon 01/04/13	Diseñador; Director artístico
	Implementación	35 days	Tue 02/04/13	Mon 20/05/13	Programador; Director artístico
	Guía de implantación	8 days	Tue 21/05/13	Thu 30/05/13	Diseñador; Programador
	Manual de usuario	10 days	Tue 21/05/13	Mon 03/06/13	Analista; Director artístico
	Entrega	1 day	Tue 04/06/13	Tue 04/06/13	Jefe de proyecto
	Seguimiento	101 days	Tue 15/01/13	Tue 04/06/13	Jefe de proyecto

Figura 35. Diagrama de Gantt 1

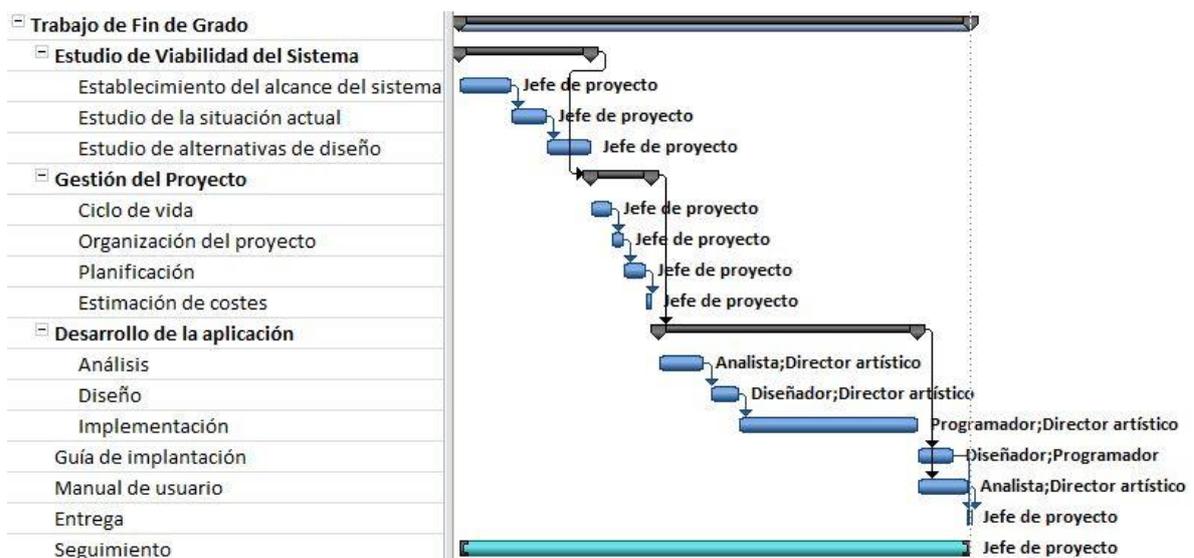


Figura 36. Diagrama de Gantt 2

3.4 Estimación de costes

Partiendo de la planificación, se conoce el número de horas aproximadas por rol que va a necesitar el proyecto. A continuación se desglosa el presupuesto.

Iván Lim Blanco

2.- Departamento:

Ingeniería del Software

3.- Descripción del Proyecto:

- Título: Análisis, Diseño e Implementación de un Juego de Estrategia para Dispositivos Móviles
- Duración (meses): 5
- Tasa de costes Indirectos: **20%**

4.- Presupuesto total del Proyecto (valores en Euros):

20,000 €

5.- Desglose presupuestario (costes directos)

PERSONAL				
Categoría	Días	Dedicación ¹	Coste/Hora	Coste (Euro)
Jefe de proyecto	101	91	30,00	2.730,00
Director artístico	61	61	20,00	1.220,00
Analista	20	40	25,00	1.000,00
Diseñador	14	28	15,00	420,00
Programador	43	86	15,00	1.290,00
Total				6.660,00

¹⁾ Del número total de jornadas laborales (8 horas) indicadas en el diagrama de Gantt, se ha dedicado solamente el 25% del tiempo a este proyecto.

EQUIPOS ¹					
Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste Imputable ^{II}
Flash CS6 (6 meses)	147,54	100	6	60	14,75
iPad 2	399,00	100	6	60	39,90
Samsung Galaxy Tab II	449,00	100	6	60	44,90
Total					99,55

^{II)} A la hora de realizar el proyecto los equipos informáticos de sobremesa ya se encontraban amortizados al 100%

^{III)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

6.- Resumen de costes

Costes totales	€
Personal	6.660
Amortización	100
Costes Indirectos	1.352
Total	8.111

Figura 37. Presupuesto

4 Desarrollo de la aplicación

Este capítulo abarca todo lo relacionado con el desarrollo de la aplicación, desde el análisis de los requisitos hasta el diseño del sistema y la guía de implantación del mismo. Al igual que el resto del documento, está estructurado de manera que guarde relación con el orden en el que se ha dispuesto el propio desarrollo de la aplicación.

4.1 Análisis

El análisis es una fase crucial en el desarrollo de una aplicación. Es la base para conocer el problema y matizar el funcionamiento del sistema. En este apartado se definirán los requisitos de usuario y de software y los casos de uso, definiendo previamente los términos con los que vamos a tratar.

4.1.1 Sobre los requisitos de Usuario y de Software

La parte principal de la fase de análisis, consiste en definir las necesidades y peticiones del cliente en forma de requisitos. De esta manera se dispone de una lista clara, concisa y estructurada de cada una de sus necesidades y de cómo van a implementarse en la aplicación.

Se distinguen dos tipos de requisitos, los requisitos de usuario (RU) y los requisitos de software (RS), y ambos se nombran y describen siguiendo un formato estándar de características que se muestra en la siguiente tabla.

ID: RS.N:001

Nombre:	Nombre del requisito		
Tipo de Requisito:	Tipo		
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Estabilidad		
Fuente:	Fuente		
Descripción:	Descripción		

Tabla 2. Ejemplo de Requisito de Usuario/Software

Para el identificador (ID) se ha escogido el siguiente formato.

RU.C: Requisito de Usuario de Capacidad.

RU.R: Requisito de Usuario de Restricción.

RS.F: Requisito de Software Funcional.

RS.N: Requisito de Software No Funcional.

De esta forma en los propios identificadores se reconocerá el tipo y sub-tipo de requisito, diferenciando numéricamente los comprendidos en el mismo bloque. El identificador está escrito por tanto componiendo la siguiente estructura:

Tipo . Sub-tipo : Número de requisito

A continuación se enumeran y describen el resto de parámetros estudiados en cada tabla de requisitos.

- **Tipo de requisito:** Clasificación estándar (de usuario/de software) explicada más adelante en este mismo apartado.

- **Necesidad:** Define la importancia que tiene la presencia del requisito, clasificándola en esencial, deseable y opcional.
- **Prioridad:** La preferencia en el orden de implementación del requisito. Puede ser alta, media o baja.
- **Verificabilidad:** Capacidad del cumplimiento del objetivo de ser reconocido una vez desarrollada la aplicación. Se clasifica en alta, media o baja.
- **Estabilidad:** Especifica si un requisito es susceptible de ser alterado durante el desarrollo.
- **Fuente:** Origen de la necesidad de aplicar este requisito.
- **Descripción:** Descripción del requisito.

Requisitos de Usuario

Los requisitos de Usuario son definidos por el cliente y el analista. Las necesidades del cliente son capturadas en requisitos, cuya definición no debe tener un nivel de detalle técnico, pero si definir con exactitud cada característica que el cliente quiere que la aplicación ofrezca.

Se distinguen dos tipos de requisitos de usuario:

- **Requisitos de Capacidad:** Definen qué es lo que debe hacer la aplicación.
- **Requisitos de Restricción:** Especifican cómo debe la aplicación cumplir determinada función.

Requisitos de Software

Una vez se disponen de los requisitos de usuario, el Analista, con el programador como audiencia, los desarrolla y refina para generar una nueva lista de requisitos más técnica, en la que se especifica con mayor claridad la lista de necesidades que el sistema ha de cubrir y de qué manera debe hacerlo.

A los componentes de esta nueva lista se los denomina Requisitos de Software, y si les diferencia en dos tipos principales:

- **Requisitos Funcionales:** Que son los equivalentes a los de capacidad, pero definidos a un nivel más bajo.
- **Requisitos No Funcionales:** Que equivalen a los de Restricción.

Según el estándar de la ESA, los requisitos no funcionales se dividen a su vez en:

- Consumo de recursos.
- Rendimiento.
- Fiabilidad y disponibilidad.
- Manejo de errores.
- De interfaz.
- Restricciones.
- Seguridad.

Enumeración de requisitos y trazabilidad

Una vez se enumeran todos los requisitos, se verifica su trazabilidad, comprobando que cada RS ha sido generado por un RU, y que cada RU genera al menos un RS. Esta comprobación se realiza a través de lo que conocemos como Matriz de trazabilidad.

La matriz de trazabilidad es la tabla que muestra la correspondencia entre un Requisito de Usuario con uno o más Requisitos de Software (trazabilidad hacia delante), o un Requisito de Software con uno o más Requisitos de Usuario (trazabilidad hacia atrás).

En los próximos apartados se enumeran en orden, y siguiendo el esquema expuesto, los requisitos de usuario y los requisitos de software, verificando después su trazabilidad.

4.1.2 Requisitos de Usuario

Requisitos de Restricción

ID: RU.R:001

Nombre:	Idioma del juego		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	Los textos del juego deben presentarse en inglés.		

Tabla 3. RU.R:001. Idioma del juego

ID: RU.R:002

Nombre:	Compatibilidad con tabletas Android		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	El sistema debe funcionar de forma estable en cualquier tableta con sistema operativo Android 2.1 o superior.		

Tabla 4. RU.R:002. Compatibilidad con tabletas Android

ID: RU.R:003

Nombre:	Compatibilidad con tabletas iPad		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	El sistema debe funcionar de forma estable en cualquier iPad versión 2 o superior.		

Tabla 5. RU.R:003. Compatibilidad con tabletas iPad

ID: RU.R:004

Nombre:	Visualización en dispositivos		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	El sistema debe ejecutarse en modo “landscape” en cualquier tipo de tableta.		

Tabla 6. RU.R:004. Visualización en dispositivos

ID: RU.R:005

Nombre:	Disposición multijugador		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	El sistema debe permitir que dos jugadores jueguen de forma simultánea colocados físicamente uno en frente del otro.		

Tabla 7. RU.R:005. Disposición multijugador**ID: RU.R:006**

Nombre:	Lenguaje apropiado		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Normativa aplicaciones Apple		
Descripción:	El texto del juego no debe contener palabras ni imágenes obscenas o inapropiadas.		

Tabla 8. RU.R:006. Lenguaje apropiado

Requisitos de Capacidad

ID: RU.C:001

Nombre:	Pantalla inicial		
Tipo de Requisito:	Capacidad		
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Usuario		
Descripción:	Antes del propio menú, el juego debe incluir una pantalla de bienvenida en la que el jugador habrá de pulsar para entrar en el menú principal del juego.		

Tabla 9. RU.C:001. Pantalla inicial

ID: RU.C:002

Nombre:	Menú Principal		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Usuario		
Descripción:	Una vez comience el juego, debe aparecer un menú con la opción de jugar para comenzar la partida.		

Tabla 10. RU.C:002. Menú Principal

ID: RU.C:003

Nombre:	Área de juego		
Tipo de Requisito:	Capacidad		
Necesidad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Usuario		
Descripción:	Una vez iniciada una batalla en el juego, cada jugador debe disponer de la mitad del área de la pantalla para visualizar los eventos concernientes a sus maniobras de ataque y defensa.		

Tabla 11. RU.C:003. Área de juego**ID: RU.C:004**

Nombre:	Victoria/Derrota		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	Al finalizar la partida, una interfaz independiente debe anunciar a cada jugador independientemente el resultado de la partida antes de volver al menú.		

Tabla 12. RU.C:004. Victoria/Derrota

ID: RU.C:005

Nombre:	Menús en batalla		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	Durante el desarrollo de cada batalla, ambos jugadores deben disponer de un acceso común a los menús de opciones de juego y guía de jugador.		

Tabla 13. RU.C:005. Menús en batalla**ID: RU.C:006**

Nombre:	Envío de enemigos		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Usuario		
Descripción:	Cada jugador debe contar con cierta variedad de enemigos para atacar a su adversario.		

Tabla 14. RU.C:006. Envío de enemigos

ID: RU.C:007

Nombre:	Torres defensivas		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Usuario		
Descripción:	Cada jugador debe disponer de al menos 3 tipos de torres. para defenderse de los ataques de su enemigo.		

Tabla 15. RU.C:007. Torres defensivas**ID: RU.C:008**

Nombre:	Elementos especiales		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Usuario		
Descripción:	Además de las torres, cada jugador debe disponer de al menos 3 habilidades u objetos especiales para defenderse.		

Tabla 16. RU.C:008. Elementos especiales

ID: RU.C:009

Nombre:	Recursos		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	El empleo de cualquier elemento de ataque o defensa tendrá un coste en un recurso virtual económico. Este recurso se incrementará progresivamente durante la batalla y al eliminar enemigos.		

Tabla 17. RU.C:009. Recursos

ID: RU.C:010

Nombre:	Interfaz integrada		
Tipo de Requisito:	Capacidad		
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Baja		
Fuente:	Usuario		
Descripción:	La interfaz del juego durante la batalla debe estar integrada en la propia ambientación del juego.		

Tabla 18. RU.C:010. Interfaz integrada

ID: RU.C:011

Nombre:	Diversión táctil		
Tipo de Requisito:	Capacidad		
Necesidad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Usuario		
Descripción:	Alguna de las habilidades especiales debe hacer uso de la funcionalidad arrastrar-soltar para hacer más divertido su manejo.		

Tabla 19. RU.C:011. Diversión táctil

ID: RU.C:012

Nombre:	Daño específico		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	Cada torre u ataque especial debe ser especialmente efectivo contra un tipo determinado de enemigos y menos útil contra otros para aumentar la diversidad de opciones de ataque.		

Tabla 20. RU.C:012. Daño específico

ID: RU.C:013

Nombre:	Nivel de enemigos		
Tipo de Requisito:	Capacidad		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Usuario		
Descripción:	El jugador más agresivo en ataque debe disponer de una bonificación, como una mejora a los enemigos que él envía.		

Tabla 21. RU.C:013. Nivel de enemigos

4.1.3 Requisitos de Software

Requisitos No Funcionales

ID: RS.N:001

Nombre:	Idioma del juego		
Tipo de Requisito:	Restricciones		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Cualquier texto del juego será presentado en inglés.		

Tabla 22. RS.N:001. Idioma del juego

ID: RS.N:002

Nombre:	Compatibilidad con sistemas operativos		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	El sistema ha de desarrollarse en Flash Professional CS6 y compilarse con la SDK de AIR 3.7, compatible con los sistemas Android e iOS especificados por el usuario.		

Tabla 23. RS.N:002. Compatibilidad con sistemas operativos

ID: RS.N:003

Nombre:	Rendimiento		
Tipo de Requisito:	Rendimiento		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Se utilizarán el menor número de gráficos posible en una resolución global de la aplicación de 800x600 para asegurar el buen rendimiento de la aplicación en dispositivos antiguos.		

Tabla 24. RS.N:003. Rendimiento**ID: RS.N:004**

Nombre:	Visualización en dispositivos		
Tipo de Requisito:	Interfaz		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Al compilar, debe forzarse a la aplicación a ejecutarse en modo "landscape" en todos los dispositivos.		

Tabla 25. RS.N:004. Visualización en dispositivos

ID: RS.N:005

Nombre:	Disposición multijugador		
Tipo de Requisito:	Interfaz		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Se distribuirá el escenario de juego de forma que cada jugador visualice su área de juego en el área de la pantalla correspondiente al 50% del total que se encuentre en la parte cercana a su posición física. Los menús se manejarán por el jugador principal.		

Tabla 26. RS.N:005. Disposición multijugador

ID: RS.N:006

Nombre:	Lenguaje apropiado		
Tipo de Requisito:	Restricción		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Normativa aplicaciones Apple		
Descripción:	El texto del juego no debe contener palabras ni imágenes obscenas o inapropiadas.		

Tabla 27. RS.N:006. Lenguaje apropiado

Requisitos Funcionales

ID: RS.F:001

Nombre:	Pantalla inicial		
Tipo de Requisito:	Funcional		
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	Se incluirá una pantalla de bienvenida con el título y la imagen principal del juego, que avanzará al menú al ser clicada en cualquier punto de la pantalla.		

Tabla 28. RS.F:001. Pantalla inicial

ID: RS.F:002

Nombre:	Menú Principal		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	Se incluirá un menú principal de juego con el título y un único botón de acceso a la pantalla de juego. Ésta pantalla estará orientada para ser visualizada por el jugador principal.		

Tabla 29. RS.F:002. Menú Principal

ID: RS.F:003

Nombre:	Área de juego		
Tipo de Requisito:	Funcional		
Necesidad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	La pantalla de juego se dividirá en dos bloques que dispondrán de la misma información, interfaz y fragmento del escenario como si fuera un espejo.		

Tabla 30. RS.F:003. Área de juego**ID: RS.F:004**

Nombre:	Color de equipo		
Tipo de Requisito:	Funcional		
Necesidad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	La diferencia fundamental en la parte gráfica de la porción de escenario de un jugador y del otro será que pequeños detalles se representarán en un color diferente, rojo para el jugador principal y azul para el secundario.		

Tabla 31. RS.F:004. Color de equipo

ID: RS.F:005

Nombre:	Área táctil		
Tipo de Requisito:	Funcional		
Necesidad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	Los eventos táctiles ocurridos durante la batalla en la porción de la pantalla de un jugador determinado, se considerarán como generados por ese usuario en concreto.		

Tabla 32. RS.F:005. Área táctil**ID: RS.F:006**

Nombre:	Victoria/Derrota		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Al finalizar la partida, una interfaz independiente con el fondo del color su equipo, informará a cada jugador en su porción de la pantalla si ha ganado o perdido.		

Tabla 33. RS.F:006. Victoria/Derrota

ID: RS.F:007

Nombre:	Menús en batalla		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	En un lateral del centro de la pantalla se mostrarán botones de acceso al menú de opciones y a la guía de jugador. El acceso a cualquiera de estos menús pausará la partida, para volver a retomarla al cerrarse el menú.		

Tabla 34. RS.F:007. Menús en batalla

ID: RS.F:008

Nombre:	Menú inGame de Opciones		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Se mostrará en forma de pop up e incluirá opciones para activar o desactivar música y efectos, y de rendimiento, habilitando o no las sombras y animaciones de personajes.		

Tabla 35. RS.F:008. Menú inGame de Opciones

ID: RS.F:009

Nombre:	Menú inGame de Guía de jugador		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	En forma de pop up y, en pestañas, mostrará información, descripciones de elementos y gráficos explicativos de la efectividad de cada elemento defensivo en cada enemigo.		

Tabla 36. RS.F:009. Menú inGame de Guía de jugador

ID: RS.F:010

Nombre:	Envío de enemigos		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	Se implementarán 8 tipos de enemigos. Diferenciando entre voladores/terrestres y especificando lo resistentes que son para cada tipo de elemento defensivo.		

Tabla 37. RS.F:010. Envío de enemigos

ID: RS.F:011

Nombre:	Enemigo especial		
Tipo de Requisito:	Funcional		
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	Uno de los enemigos dispondrá de un comportamiento especial. Sobrevolará el área enemiga otorgando a todos los personajes de su equipo una bonificación de velocidad.		

Tabla 38. RS.F:011. Enemigo especial

ID: RS.F:012

Nombre:	Torres defensivas		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	Habrá 4 tipos de torres. La primera tendrá un daño genérico y disparará flechas; La segunda disparará bombas a objetivos terrestres; la tercera será anti aérea y hará daño extra a enemigos voladores; la cuarta ralentizará a los enemigos.		

Tabla 39. RS.F:012. Torres defensivas

ID: RS.F:013

Nombre:	Construcción de torres		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	En el escenario se colocarán troncos de árboles en los que el jugador podrá construir las torres pulsando sobre ellos. Se le permitirá elegir qué tipo de torre quiere construir.		

Tabla 40. RS.F:013. Construcción de torres

ID: RS.F:014

Nombre:	Elementos especiales		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	4 elementos de defensa especiales. Una metralleta automática; una bomba que podrá ser lanzada por el jugador cada cierto tiempo; un helicóptero manual de ataque a unidades aéreas y una avispa que ralentizará a los enemigos.		

Tabla 41. RS.F:014. Elementos especiales

ID: RS.F:015

Nombre:	Evolución de defensas		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	Los elementos defensivos podrán evolucionarse al clicar sobre ellos, pagando un coste proporcional a la mejora.		

Tabla 42. RS.F:015. Evolución de defensas**ID: RS.F:016**

Nombre:	Recurso principal		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	El recurso principal se definirá como energía. Todos los elementos de ataque o defensa y sus evoluciones tendrán un precio en energía para el jugador.		

Tabla 43. RS.F:016. Recurso principal

ID: RS.F:017

Nombre:	Matriz de precios		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Se definirá en una matriz el coste de cada elemento de ataque y defensa para poder ajustar los precios con facilidad durante el balance del juego.		

Tabla 44. RS.F:017. Matriz de precios

ID: RS.F:018

Nombre:	Interfaz integrada		
Tipo de Requisito:	Funcional		
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Baja		
Fuente:	Analista		
Descripción:	Los botones de envío de enemigos se integrarán en el escenario como si formaran parte de él. El resto de opciones se mostrarán en menús emergentes cuando corresponda.		

Tabla 45. RS.F:018. Interfaz integrada

ID: RS.F:019

Nombre:	Diversión táctil		
Tipo de Requisito:	Funcional		
Necesidad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Media		
Fuente:	Analista		
Descripción:	La bomba, helicóptero y avispa (elementos especiales) serán manejados por el jugador pulsando sobre ellos y arrastrando el dedo al punto al que deseen moverlos.		

Tabla 46. RS.F:019. Diversión táctil

ID: RS.F:020

Nombre:	Daño específico		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Se definirá en una matriz el porcentaje de daño que cada elemento defensivo hace a cada enemigo para diferenciar la efectividad de cada uno.		

Tabla 47. RS.F:020. Daño específico

ID: RS.F:021

Nombre:	Nivel de enemigos		
Tipo de Requisito:	Funcional		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Alta		
Fuente:	Analista		
Descripción:	Cada enemigo enviado por un jugador incrementará el nivel de los próximos que envíe. El nivel se mostrará visualmente en la interfaz de juego de su pantalla y aumentará la resistencia general de los enemigos que envía.		

Tabla 48. RS.F:021. Nivel de enemigos

4.1.4 Trazabilidad de requisitos

A continuación se expone la matriz de trazabilidad de requisitos de usuario y sistema, en la que como se explicó anteriormente, se muestran las relaciones hacia atrás y hacia delante entre los diferentes requisitos de usuario planteados por el cliente, y los requisitos de software desarrollados por el diseñador.

Horizontal azul: Requisitos de Usuario de Restricción.

Horizontal verde: Requisitos de Usuario de Capacidad.

Vertical azul: Requisitos de Software No Funcionales.

Vertical verde: Requisitos de Software Funcionales.

	01	02	03	04	05	06	01	02	03	04	05	06	07	08	09	10	11	12	13	
01	X																			
02		X	X																	
03		X	X																	
04				X																
05					X															
06						X														
01							X													
02								X												
03					X				X											
04					X				X											
05					X				X											
06										X										
07											X									
08												X								
09													X							
10																X				
11																	X			
12																		X		
13																			X	
14																				X
15																	X	X		
16																				X
17																				X
18																				X
19																				X
20																				X
21																				X

Tabla 49. Matriz de Trazabilidad

4.1.5 Casos de uso

Los casos de uso son una herramienta empleada para el análisis de la relación entre usuario y sistema. Describen los pasos o actividades que deben realizarse para llevar a cabo un proceso, expresándolos en una tabla como la que sigue.

ID: Identificador

Título:	Ejemplo de Caso de uso
Actor:	Actor
Descripción:	Descripción del caso.
Precondiciones:	Se han de cumplir para que se lleve a cabo el proceso.
Post-condiciones:	Se producen por la realización del proceso.
Escenario Principal:	Escenario principal.
Escenario Alternativo:	Escenario alternativo.
Referencia Diagrama	Sección correspondiente del diagrama de casos de uso.

Tabla 50. Esquema de Casos de uso

A continuación, siguiendo el esquema y descripciones del ejemplo se van a enumerar los casos de uso de la aplicación desarrollada en este proyecto.

ID: CU:000

Título:	Cerrar la aplicación
Actor:	Usuario
Descripción:	El usuario cierra la aplicación.
Precondiciones:	El sistema debe haberse iniciado.
Post-condiciones:	Se detienen los servicios y se cierra la aplicación.
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “home” de su dispositivo. 4. La aplicación se cierra.
Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.</p>
Referencia Diagrama	Menú Principal

Tabla 51. CU:000. Cerrar la aplicación

ID: CU:001

Título:	Iniciar una partida
Actor:	Usuario
Descripción:	El usuario inicia una partida.
Precondiciones:	El sistema debe haberse iniciado.
Post-condiciones:	Se inicia una partida de dos jugadores.
Escenario Principal:	1. El usuario inicia la aplicación desde su dispositivo.

	<ol style="list-style-type: none"> 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “Start Battle” del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores.
Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.</p>
Referencia Diagrama	Menú Principal

Tabla 52. CU:001. Iniciar una partida

ID: CU:002

Título:	Construir una torre
Actor:	Usuario
Descripción:	Un usuario construye una torre.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p>
Post-condiciones:	Se crea una torre para el usuario en la posición seleccionada.
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “Start Battle” del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores. 5. El usuario pulsa en un tronco y se le muestra la interfaz de torres para comprar. 6. El usuario pulsa en la torre que desea comprar. 7. La torre se construye y el tronco es sustituido por el gráfico de la torre correspondiente.
Escenario	1' y 2'. El usuario puede ya haber iniciado la aplicación y

Alternativo:	<p>encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón "back" de su dispositivo para volver al Menú Principal.</p> <p>6'. El usuario podría no disponer de energía suficiente y el proceso no se completaría.</p>
---------------------	---

Referencia Diagrama [Menú de interacción con Torre](#)

Tabla 53. CU:002. Construir una torre

ID: CU:003

Título:	Mejorar una torre
----------------	-------------------

Actor: Usuario

Descripción:	El usuario mejora una torre.
---------------------	------------------------------

Precondiciones:

- El sistema debe haberse iniciado.
- Se debe haber iniciado el modo batalla.
- Debe haberse construido al menos una torre.

Post-condiciones:	<p>Se mejora la torre seleccionada a su siguiente nivel.</p> <p>Se descuenta al usuario la energía empleada.</p>
--------------------------	--

Escenario Principal:

1. El usuario inicia la aplicación desde su dispositivo.
2. La aplicación accede al Menú Principal.
3. Pulsa el botón "Start Battle" del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. El usuario pulsa en un tronco y se le muestra la interfaz de torres para comprar.
6. El usuario pulsa en la torre que desea comprar.
7. La torre se construye en la posición seleccionada y la interfaz de comprar torre desaparece.
8. El usuario pulsa sobre la torre construida.

- 9. El menú de mejorar o vender torre aparece.
- 10. El usuario pulsa sobre el botón de mejorar torre.
- 11. La torre se mejora y la interfaz desaparece.

Escenario Alternativo:

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón "back" de su dispositivo para volver al Menú Principal.
- 6'. El usuario podría no disponer de energía suficiente y el proceso no se completaría.
- 9'. La torre podría encontrarse en máximo nivel y la opción de mejorarla no se haría visible.
- 10'. El usuario podría no disponer de energía suficiente para mejorar la torre y la acción no se completaría.

Referencia Diagrama [Menú de interacción con Torre](#)

Tabla 54. CU:003. Mejorar una torre

ID: CU:004

Título:	Vender una torre
Actor:	Usuario
Descripción:	El usuario vende una torre.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p> <p>Debe haberse construido al menos una torre.</p>
Post-condiciones:	<p>Se mejora la torre seleccionada a su siguiente nivel.</p> <p>Se otorga al usuario la energía correspondiente a la venta.</p>
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal.

3. Pulsa el botón “Start Battle” del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. El usuario pulsa en un tronco y se le muestra la interfaz de torres para comprar.
6. El usuario pulsa en la torre que desea comprar.
7. La torre se construye en la posición seleccionada y la interfaz de comprar torre desaparece.
9. El usuario pulsa sobre la torre construida.
10. El menú de mejorar o vender torre aparece.
11. El usuario pulsa sobre el botón de vender torre.
12. Un tronco sustituye a la torre y la interfaz desaparece.

Escenario Alternativo:

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.
- 6'. El usuario podría no disponer de energía suficiente y el proceso no se completaría.

Referencia Diagrama [Menú de interacción con Torre](#)

Tabla 55. CU:004. Vender una torre

ID: CU:005

Título:	Enviar un enemigo
Actor:	Usuario
Descripción:	El usuario envía un enemigo a su oponente.
Precondiciones:	El sistema debe haberse iniciado. Se debe haber iniciado el modo batalla.
Post-condiciones:	Se envía un enemigo al oponente.

	Se descuenta al usuario la energía empleada.
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “Start Battle” del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores. 5. El usuario pulsa en el icono correspondiente al enemigo que desea enviar. 6. El enemigo es generado en el escenario del rival.
Escenario Alternativo:	<p>1’ y 2’. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3’. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.</p> <p>5’. El usuario podría no disponer de energía suficiente y el proceso no se completaría.</p>
Referencia Diagrama	Interactuar con Interfaz de Unidades

Tabla 56. CU:005. Enviar un enemigo

ID: CU:006

Título:	Comprar habilidad especial
Actor:	Usuario
Descripción:	El usuario compra una habilidad especial.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p>
Post-condiciones:	<p>Se habilita la habilidad especial.</p> <p>Se descuenta al usuario la energía empleada en su compra.</p>
Escenario Principal:	1. El usuario inicia la aplicación desde su dispositivo.

2. La aplicación accede al Menú Principal.
3. Pulsa el botón “Start Battle” del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. El usuario pulsa en el icono correspondiente a la habilidad que desea comprar.
6. La habilidad se activa para su uso por parte del usuario.

Escenario Alternativo:

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.
- 5'. El usuario podría no disponer de energía suficiente y el proceso no se completaría.

Referencia Diagrama [Interactuar con Interfaz de Habilidad Especial](#)

Tabla 57. CU:006. Comprar habilidad especial

ID: CU:007

Título: Mejorar habilidad especial

Actor: Usuario

Descripción: El usuario mejora una habilidad especial.

Precondiciones: El sistema debe haberse iniciado.
Se debe haber iniciado el modo batalla.
Se debe haber comprado la habilidad especial que se desea mejorar.

Post-condiciones: Se mejora la habilidad especial.
Se descuenta al usuario la energía empleada en su compra.

Escenario Principal:

1. El usuario inicia la aplicación desde su dispositivo.
2. La aplicación accede al Menú Principal.

3. Pulsa el botón “Start Battle” del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. El usuario pulsa en el icono correspondiente a la habilidad que desea comprar.
6. La habilidad se activa para su uso por parte del usuario.
7. El usuario pulsa en el botón correspondiente a la habilidad especial.
8. La habilidad especial se mejora.

Escenario Alternativo:

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.
- 5'. El usuario podría no disponer de energía suficiente y el proceso no se completaría.
- 7'. El usuario podría no disponer de energía suficiente y el proceso no se completaría.
- 7''. La habilidad podría estar en su máximo nivel, en tal caso no se permitirían mejoras adicionales.

Referencia Diagrama [Interactuar con Interfaz de Habilidad Especial](#)

Tabla 58. CU:007. Mejorar habilidad especial

ID: CU:008

Título:	Cambiar calidad gráfica
Actor:	Usuario
Descripción:	El usuario cambia la calidad gráfica.
Precondiciones:	El sistema debe haberse iniciado. Se debe haber iniciado el modo batalla.
Post-condiciones:	Se cambia la calidad gráfica de toda la aplicación

Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “Start Battle” del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores. 5. El usuario pulsa en el icono de opciones de sistema. 6. El menú de opciones se muestra pausando la partida. 7. El usuario pulsa en el botón correspondiente a la calidad gráfica deseada. 8. La calidad gráfica de la aplicación se ajusta.
-----------------------------	--

Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.</p> <p>7'. El sistema podría estar ejecutándose en la calidad gráfica seleccionada. En este caso, no se producirían cambios.</p>
-------------------------------	---

Referencia Diagrama [Menú de Ajustes](#)

Tabla 59.CU:008. Cambiar calidad gráfica

ID: CU:009

Título:	Habilitar tipos de sonido
Actor:	Usuario
Descripción:	El usuario activa o desactiva diferentes tipos de sonidos.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p>
Post-condiciones:	Se activan o desactivan diferentes tipos de sonidos.
Escenario Principal:	1. El usuario inicia la aplicación desde su dispositivo.

2. La aplicación accede al Menú Principal.
3. Pulsa el botón “Start Battle” del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. El usuario pulsa en el icono de opciones de sistema.
6. El menú de opciones se muestra pausando la partida.
7. El usuario pulsa en el botón correspondiente al tipo de sonido que desea habilitar o deshabilitar.
8. Se ajustan las restricciones de reproducción de sonidos.

Escenario Alternativo:

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.

Referencia Diagrama [Menú de Ajustes](#)

Tabla 60. CU:009. Habilitar tipos de sonido

ID: CU:010

Título: Terminar batalla

Actor: Usuario

Descripción: El usuario vuelve al Menú Principal.

Precondiciones: El sistema debe haberse iniciado.
Se debe haber iniciado el modo batalla.

Post-condiciones: Se vuelve al Menú Principal.

Escenario Principal:

1. El usuario inicia la aplicación desde su dispositivo.
2. La aplicación accede al Menú Principal.
3. Pulsa el botón “Start Battle” del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.

	<ol style="list-style-type: none"> 5. El usuario pulsa en el icono de opciones de sistema. 6. El menú de opciones se muestra pausando la partida. 7. El usuario pulsa en el botón “End Battle” 8. El sistema vuelve al Menú Principal.
Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.</p>
Referencia Diagrama	Menú de Ajustes

Tabla 61. CU:010. Terminar batalla

ID: CU:011

Título:	Retomar batalla
Actor:	Usuario
Descripción:	El usuario continúa la batalla.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p>
Post-condiciones:	Se continúa la partida tras haber entrado en el menú.
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “Start Battle” del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores. 5. El usuario pulsa en el icono de opciones de sistema. 6. El menú de opciones se muestra pausando la partida. 7. El usuario pulsa en el botón “Resume Battle” 8. Se cierra el menú y se reanuda la partida.

Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón "back" de su dispositivo para volver al Menú Principal.</p>
Referencia Diagrama	Menú de Ajustes

Tabla 62. CU:011. Retomar batalla

ID: CU:012

Título:	Victoria/Derrota
Actor:	Usuario
Descripción:	Uno de los jugadores gana la batalla.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p>
Post-condiciones:	Se pasa al Menú de Victoria/Derrota.
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón "Start Battle" del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores. 5. Uno de los jugadores ve reducidas sus vidas a 0. 6. La batalla termina. 7. Se pasa al Menú de Victoria/Derrota, mostrando como ganador al jugador opuesto al que tiene su marcador de vidas en 0.
Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el</p>

botón “back” de su dispositivo para volver al Menú Principal.

Referencia Diagrama [En Partida](#)

Tabla 63. CU:012. Victoria/Derrota

ID: CU:013

Título: Consultar guía de juego

Actor: Usuario

Descripción: El usuario accede a la guía de juego.

Precondiciones: El sistema debe haberse iniciado.
Se debe haber iniciado el modo batalla.

Post-condiciones: Se muestra la guía de juego.

Escenario Principal:

1. El usuario inicia la aplicación desde su dispositivo.
2. La aplicación accede al Menú Principal.
3. Pulsa el botón “Start Battle” del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. El usuario pulsa en el icono Guía de Juego.

Escenario Alternativo:

1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.

3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.

Referencia Diagrama [En Partida](#)

Tabla 64. CU:013. Acceder a la Guía de juego

ID: CU:014

Título: Activar/desactivar sonido

Actor:	Usuario
Descripción:	El usuario activa o desactiva el sonido.
Precondiciones:	El sistema debe haberse iniciado. Se debe haber iniciado el modo batalla.
Post-condiciones:	Se activa el sonido si estaba desactivado o viceversa.
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “Start Battle” del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores. 5. El usuario pulsa en el icono de activar/desactivar sonido. 6. El sonido se deshabilita.
Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.</p> <p>6'. El sonido podría estar ya deshabilitado. En este caso el sonido se habilitaría.</p>
Referencia Diagrama	En Partida

Tabla 65. CU:014. Activar/desactivar sonido

ID: CU:015

Título:	Lanzar Bomba
Actor:	Usuario
Descripción:	El usuario lanza una bomba.
Precondiciones:	El sistema debe haberse iniciado. Se debe haber iniciado el modo batalla.

Se debe haber comprado la habilidad especial Bomba.

Post-condiciones:

La bomba cae en el escenario dañando a los enemigos afectados.

1. El usuario inicia la aplicación desde su dispositivo.
2. La aplicación accede al Menú Principal.
3. Pulsa el botón "Start Battle" del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. El usuario pulsa en el botón de compra de la habilidad especial Bomba.

Escenario Principal:

6. El objeto Bomba se habilita.
7. El usuario pulsa sobre el objeto Bomba.
8. La bomba sigue al dedo del usuario mientras lo mantenga pulsado
9. El usuario suelta el dedo de la pantalla.
10. La bomba cae en el último punto en el que el dedo del usuario mantuvo contacto con la pantalla.

Escenario Alternativo:

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón "back" de su dispositivo para volver al Menú Principal.
- 5'. El usuario podría no disponer de energía suficiente para comprar la habilidad especial. En este caso el proceso no se completaría.
- 5''. El usuario podría ya haber comprado la habilidad especial Bomba.
- 7'. El usuario podría ya haber lanzado la bomba y haber transcurrido menos de 20 segundos desde el lanzamiento. En este caso el objeto Bomba no estaría disponible.
- 8'. El usuario podría mover el dedo fuera del área de pantalla dedicada a su escenario. En este caso la bomba no le seguiría.

Tabla 66. CU:015. Lanzar Bomba

ID: CU:016

Título:	Mover Helicóptero
Actor:	Usuario
Descripción:	El usuario maneja el Helicóptero.
Precondiciones:	El sistema debe haberse iniciado. Se debe haber iniciado el modo batalla. Se debe haber comprado la habilidad especial Helicóptero.
Post-condiciones:	El Helicóptero se mueve hasta la posición indicada.
Escenario Principal:	<ol style="list-style-type: none">1. El usuario inicia la aplicación desde su dispositivo.2. La aplicación accede al Menú Principal.3. Pulsa el botón "Start Battle" del Menú Principal.4. La aplicación carga la pantalla de batalla de dos jugadores.5. El usuario pulsa en el botón de compra de la habilidad especial Helicóptero.6. El objeto Helicóptero se habilita.7. El usuario pulsa sobre el objeto Helicóptero.8. El marcador de posicionamiento del Helicóptero sigue al dedo del usuario mientras lo mantenga pulsado9. El usuario suelta el dedo de la pantalla.10. El Helicóptero avanza hasta la posición marcada con el marcador de posicionamiento.
Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el</p>

botón “back” de su dispositivo para volver al Menú Principal.

5'. El usuario podría no disponer de energía suficiente para comprar la habilidad especial. En este caso el proceso no se completaría.

5''. El usuario podría ya haber comprado la habilidad especial Helicóptero.

8'. El usuario podría mover el dedo fuera del área de pantalla dedicada a su escenario. En este caso el marcador de posicionamiento del Helicóptero no le seguiría.

Referencia Diagrama [Interactuar con Habilidad Especial](#)

Tabla 67. CU:016. Mover Helicóptero

ID: CU:017

Título:	Mover Avispa
Actor:	Usuario
Descripción:	El usuario maneja la Avispa.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p> <p>Se debe haber comprado la habilidad especial Avispa.</p>
Post-condiciones:	La Avispa aparece en la posición indicada.
Escenario Principal:	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal. 3. Pulsa el botón “Start Battle” del Menú Principal. 4. La aplicación carga la pantalla de batalla de dos jugadores. 5. El usuario pulsa en el botón de compra de la habilidad especial Avispa. 6. El objeto Avispa se habilita. 7. El usuario pulsa sobre el objeto Avispa.

	<p>8. La Avispa sigue al dedo del usuario mientras lo mantenga pulsado</p> <p>9. El usuario suelta el dedo de la pantalla.</p> <p>10. La Avispa queda posicionada en el último punto pulsado.</p>
Escenario Alternativo:	<p>1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.</p> <p>3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón "back" de su dispositivo para volver al Menú Principal.</p> <p>5'. El usuario podría no disponer de energía suficiente para comprar la habilidad especial. En este caso el proceso no se completaría.</p> <p>5''. El usuario podría ya haber comprado la habilidad especial Avispa.</p> <p>8'. El usuario podría mover el dedo fuera del área de pantalla dedicada a su escenario. En este caso la Avispa no le seguiría.</p>
Referencia Diagrama	Interactuar con Habilidad Especial

Tabla 68. CU:017. Mover Avispa

ID: CU:018

Título:	Recoger Estrella
Actor:	Usuario
Descripción:	El usuario recoge una Estrella.
Precondiciones:	<p>El sistema debe haberse iniciado.</p> <p>Se debe haber iniciado el modo batalla.</p>
Post-condiciones:	Se otorga la bonificación especial al usuario correspondiente a la estrella recogida.
Escenario Principal:	1. El usuario inicia la aplicación desde su dispositivo.

2. La aplicación accede al Menú Principal.
3. Pulsa el botón “Start Battle” del Menú Principal.
4. La aplicación carga la pantalla de batalla de dos jugadores.
5. Una Estrella Amarilla cae en el escenario.
6. El usuario pulsa sobre la Estrella.
7. Se envía una Unidad Estrella al enemigo.

Escenario Alternativo:

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.
- 5'. La Estrella podría ser Azul. En este caso se aplicarán los beneficios de ese tipo de estrella.
- 7'. Si la Estrella fuera azul, se otorgaría una bonificación de 25 puntos de energía.

Referencia Diagrama [En Partida](#)

Tabla 69. CU:018. Recoger Estrella

ID: CU:019

Título:	Enviar Unidad
Actor:	Usuario
Descripción:	El usuario envía un enemigo a su oponente.
Precondiciones:	El sistema debe haberse iniciado. Se debe haber iniciado el modo batalla.
Post-condiciones:	Se envía un enemigo al escenario del oponente.
Escenario Principal:	1. El usuario inicia la aplicación desde su dispositivo. 2. La aplicación accede al Menú Principal.

-
3. Pulsa el botón “Start Battle” del Menú Principal.
 4. La aplicación carga la pantalla de batalla de dos jugadores.
 5. El usuario pulsa sobre uno de los botones de enviar enemigos.
 6. Se genera el enemigo correspondiente en el escenario del oponente.

**Escenario
Alternativo:**

- 1' y 2'. El usuario puede ya haber iniciado la aplicación y encontrarse en el Menú Principal.
- 3'. El usuario puede encontrarse en el modo batalla o la pantalla de Victoria/Derrota. En este caso deberá pulsar el botón “back” de su dispositivo para volver al Menú Principal.
- 5'. El usuario podría no disponer de energía suficiente para adquirir la unidad seleccionada. En este caso el proceso no se completaría.

Referencia Diagrama [Interactuar con Interfaz de Unidades](#)

Tabla 70. CU:019. Enviar Enemigo

4.1.6 Diagramas de casos de uso

A continuación, y partiendo de los casos de uso, se muestran los diagramas de casos de uso, que externalizando al actor (el usuario de la aplicación), clasifican las funcionalidades del sistema. Las siglas H.E. se utilizarán para nombrar a las Habilidades Especiales.

Menú Principal

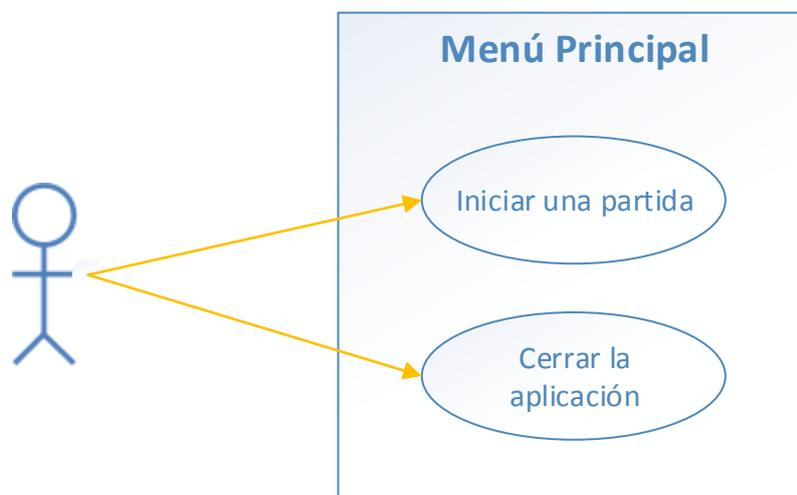


Figura 38. Diagrama de caso de uso: Menú Principal

En partida

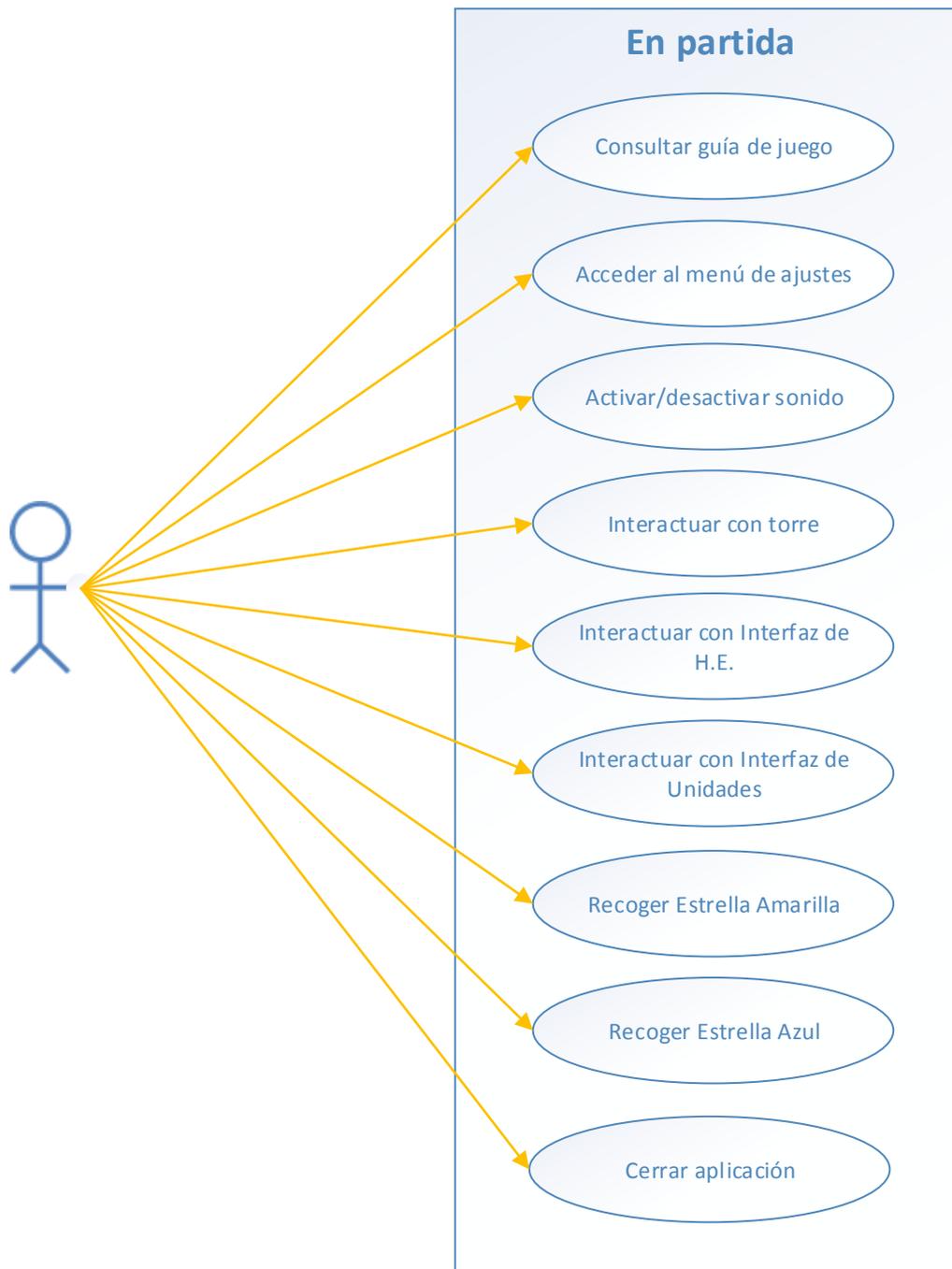


Figura 39. Diagrama de caso de uso: En partida

Interactuar con Interfaz de Unidades

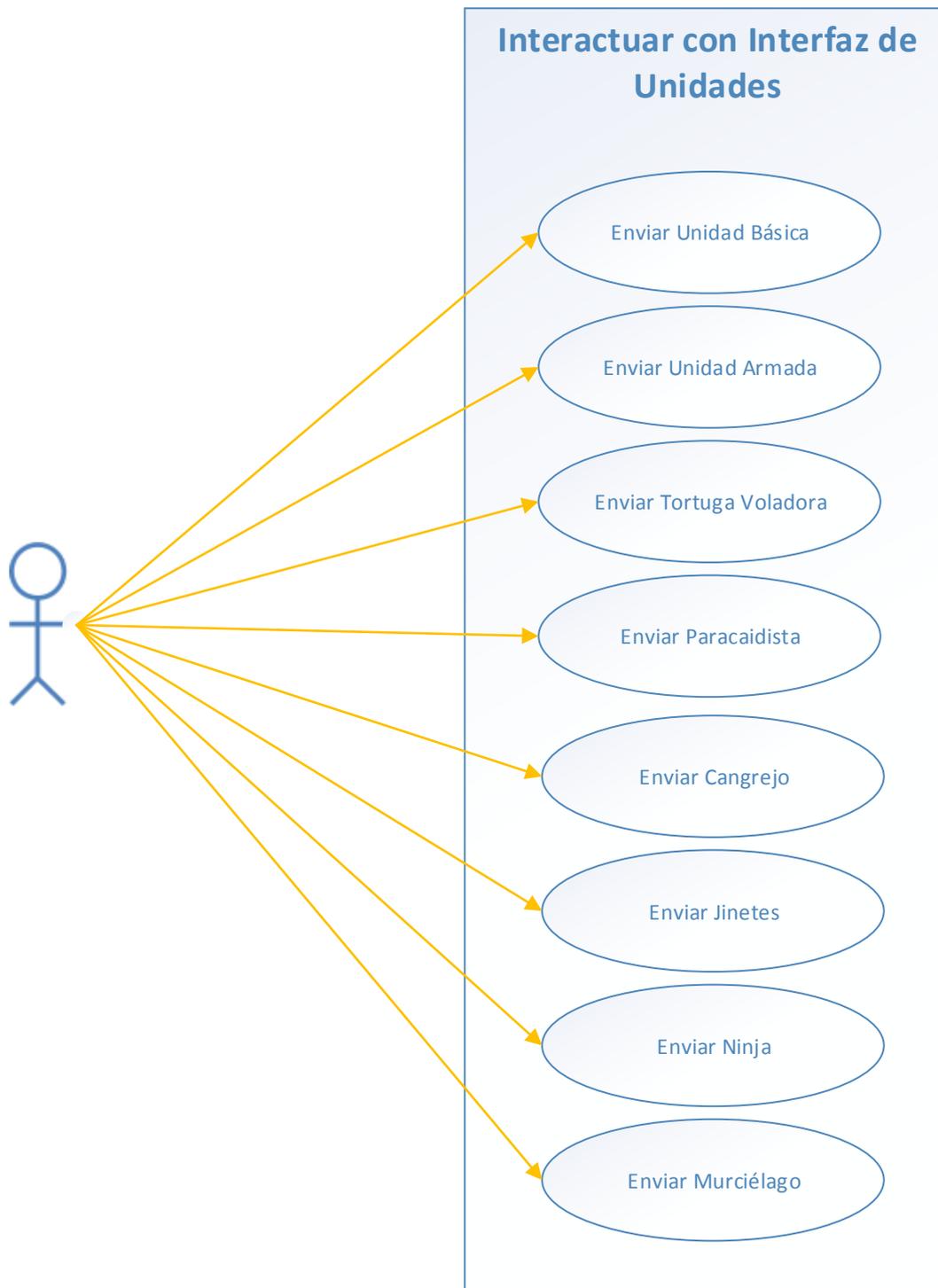


Figura 40. Diagrama de caso de uso: Interactuar con Interfaz de Unidades

Interactuar con Interfaz de Habilidades Especiales

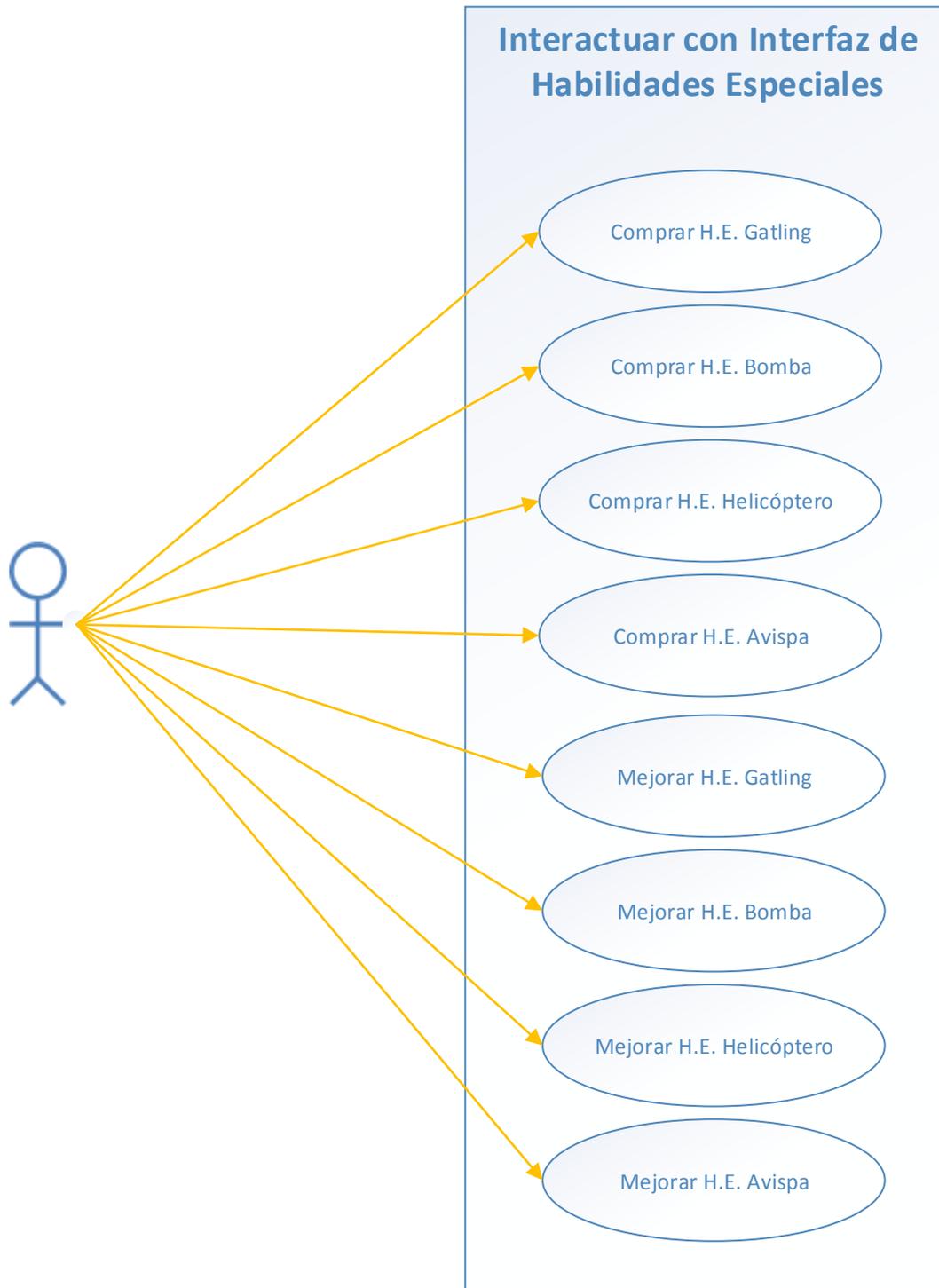


Figura 41. Diagrama de caso de uso: Interactuar con Interfaz de Habilidades Especiales

Menú de Interacción con Torre

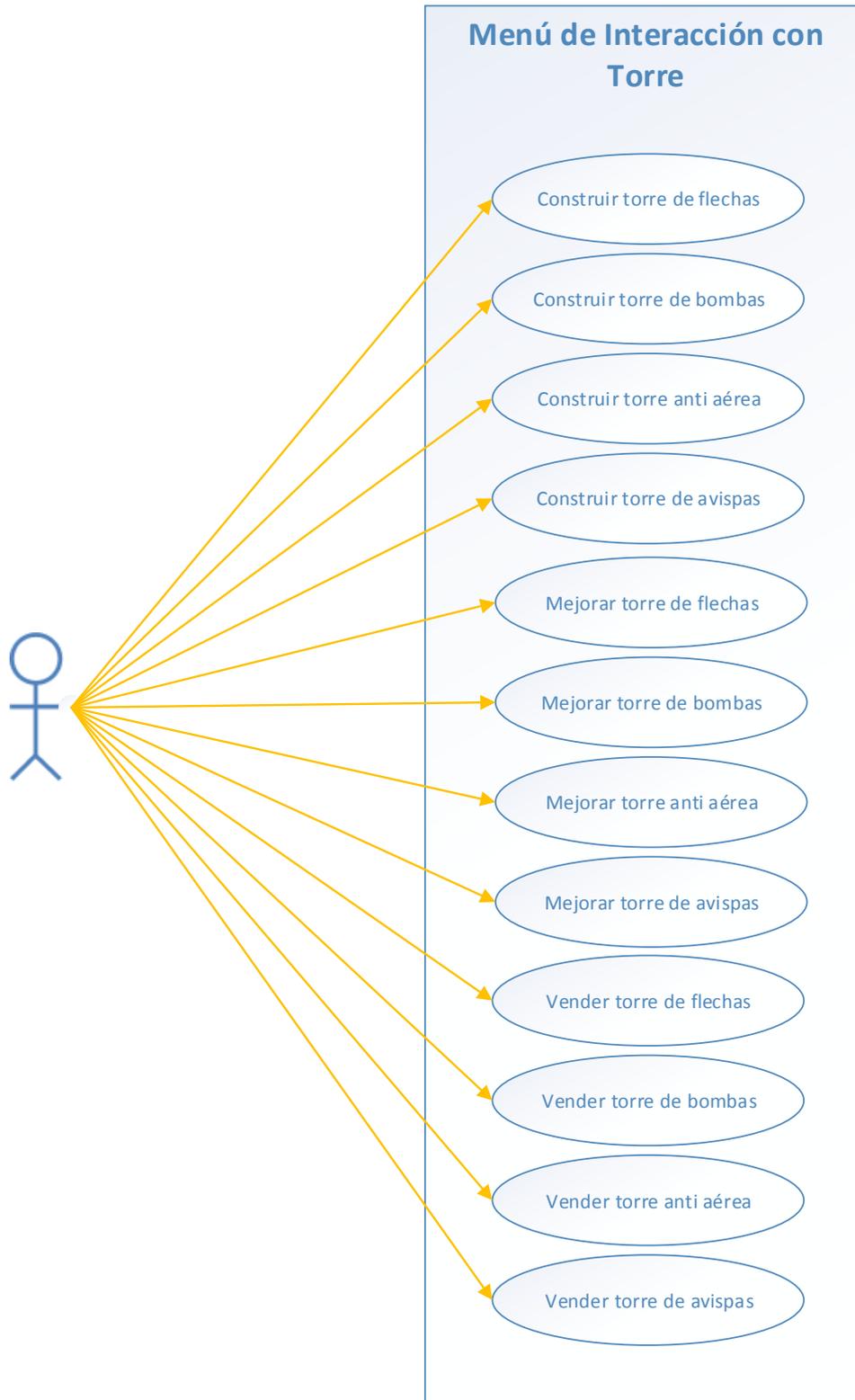


Figura 42. Diagrama de caso de uso: Menú de Interacción con Torre

Menú de Ajustes

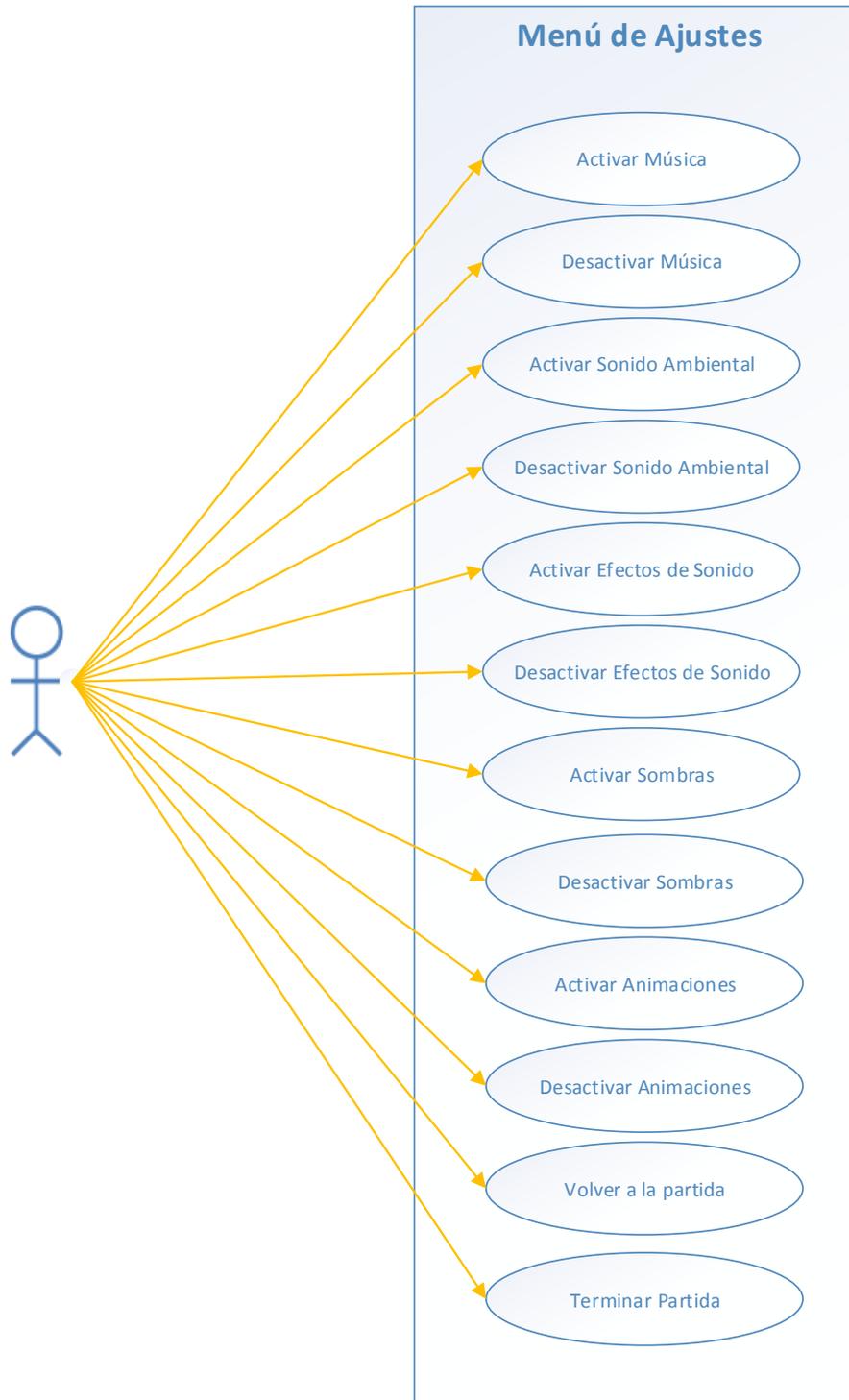


Figura 43. Diagrama de caso de uso: Menú de Ajustes

Guía de Juego

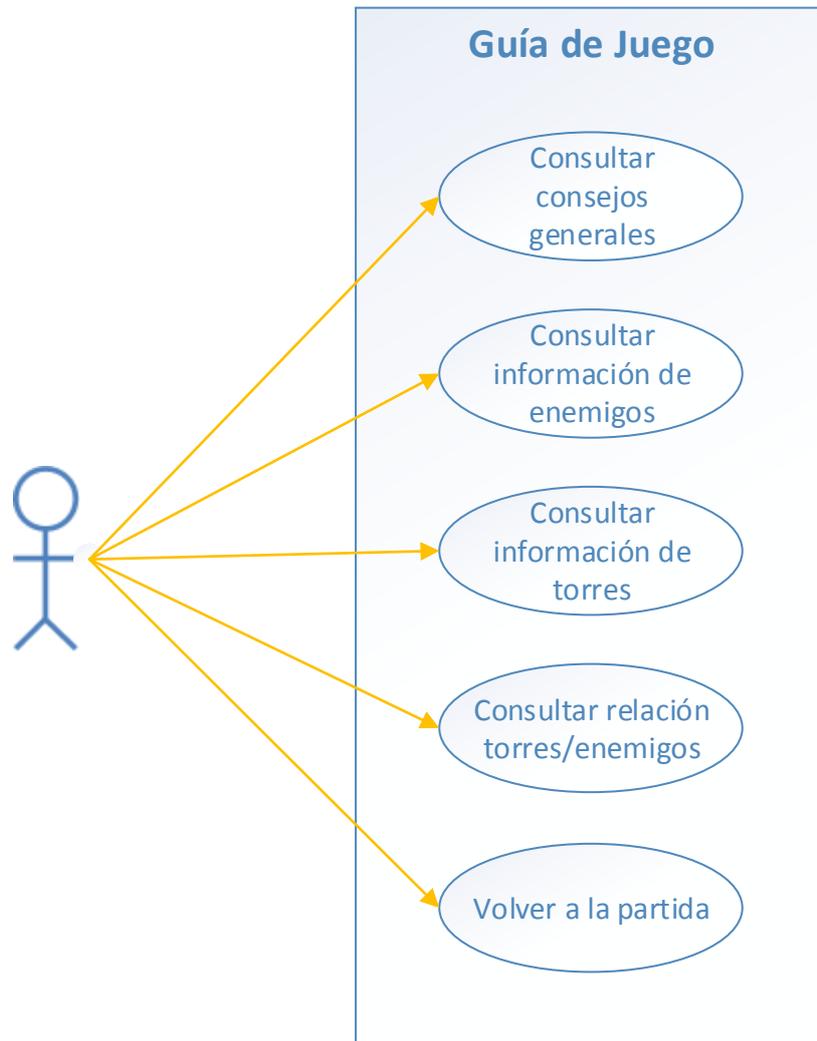


Figura 44. Diagrama de caso de uso: Guía de Juego

Interactuar con Habilidad Especial

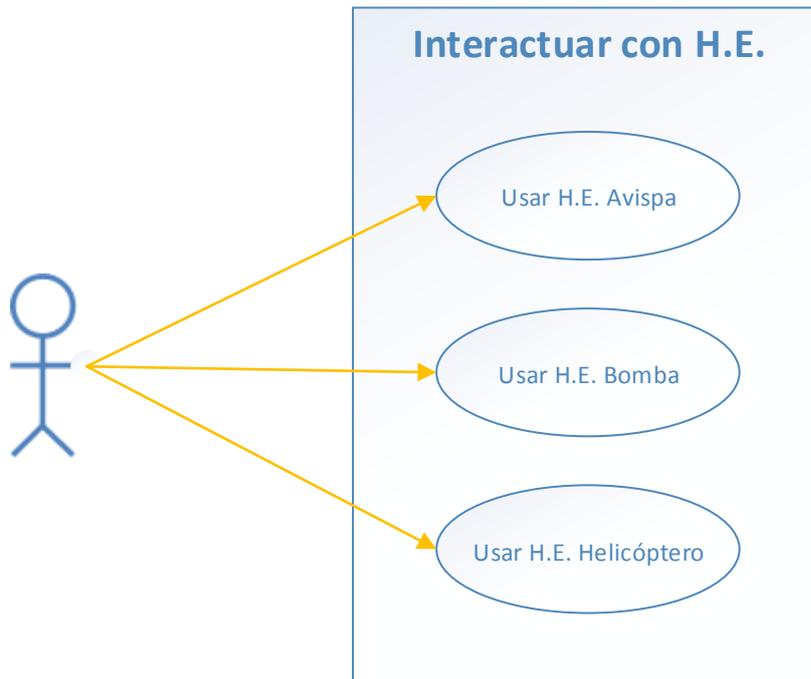


Figura 45. Diagrama de caso de uso: Interactuar con Habilidad Especial

Menú fin de partida

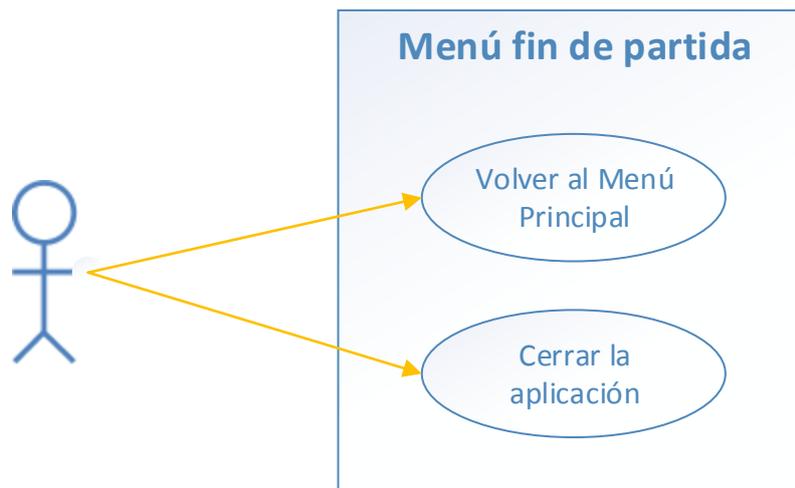


Figura 46. Diagrama de caso de uso: Menú fin de partida

4.2 Diseño

Una vez completada la fase de análisis, se comienza a dar una forma más real y menos abstracta al sistema. Se modela la aplicación de la forma más sólida posible para facilitar el paso posterior, que será la propia implementación de la aplicación por parte del programador.

4.2.1 Gestión de la migración a dispositivos de pantalla multitáctil

Como se ha descrito a lo largo del documento, la complejidad en la fase de implementación de este proyecto reside en la correcta adaptación de un juego de PC con interfaz de usuario orientada al manejo con ratón, a la nueva versión manejable por dos jugadores en dispositivos de pantalla multitáctil.

Esta adaptación conlleva un profundo esfuerzo de desarrollo en el diseño e implementación de la aplicación.

En los próximos puntos, se van a analizar los principales aspectos que se trabajan para conseguir una migración óptima.

Rendimiento

Aunque la capacidad y potencia de los dispositivos móviles aumenta cada día, el proyecto se enfoca de forma que la aplicación final pueda ser ejecutada sin problemas en dispositivos actuales de gama baja. A continuación se enumeran las principales vías que se siguen en este proyecto para mejorar el rendimiento del juego.

- Rediseño del sistema de “eliminación de basura” de la memoria, eliminando manualmente de la memoria tanto listeners (objetos de escucha) como referencias a objetos o imágenes que desaparecen de la pantalla.
- Implementación de un nuevo sistema de pintado de imágenes basado en la carga de secuencias de .png’s en lugar de imágenes en formato vector.
- Precarga de todas las clases del juego al iniciar la aplicación.
- Sustitución del método “alpha = 0” por la desactivación del parámetro “visible” (visible = false) para objetos que se hacen desaparecer temporalmente de la pantalla mediante la aplicación de propiedades de transparencia.
- Reducción máxima de aplicación de filtros en tiempo real en objetos animados.

- Limitación del parámetro “allow smoothing” en bitmaps a imágenes de principal importancia en el juego.
- Depuración del sistema de pintado de textos para limitar su empleo a los momentos necesarios durante la partida.

La reducción del empleo de filtros se realiza de forma que no haya una pérdida importante de calidad gráfica y visual en el juego. En los filtros que se mantienen, se reduce la calidad mínimamente.

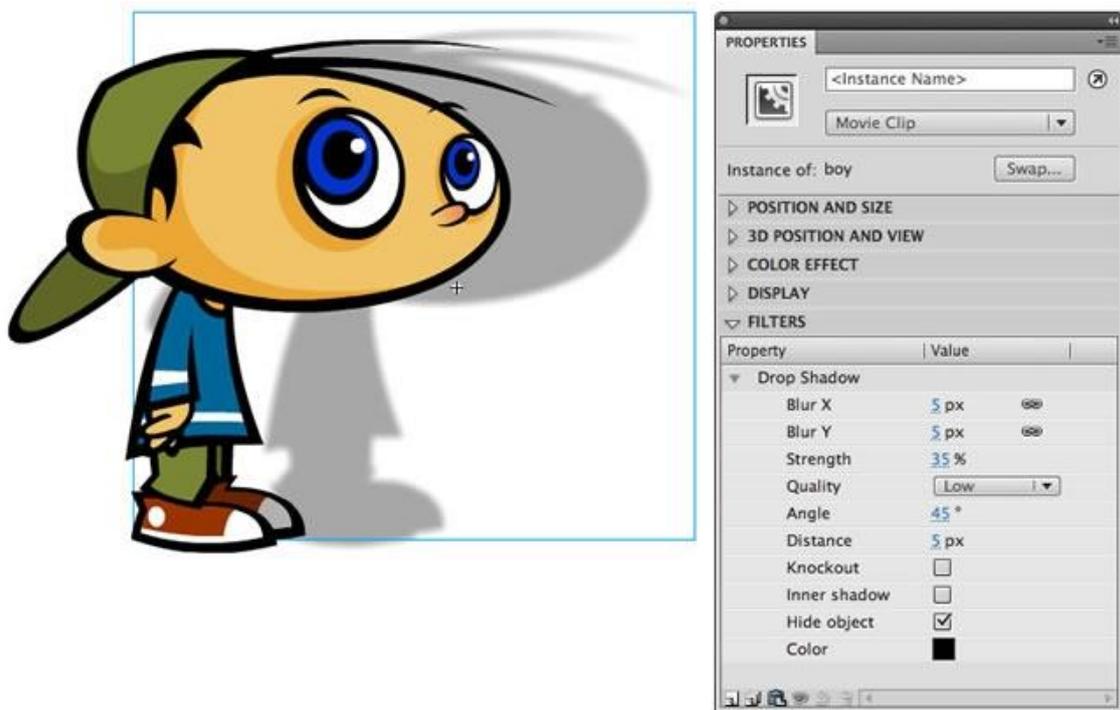


Figura 47. Pre-visualización de filtro de sombreado en Flash Professional [a.20]

En la sustitución de imágenes en formato vector por .png's, se tiene en cuenta que la calidad de la imagen en formato .png se pierde con el escalado. Por este motivo se emplean imágenes de una resolución óptima para permitir un cierto grado de escalado afectando al mínimo el rendimiento.

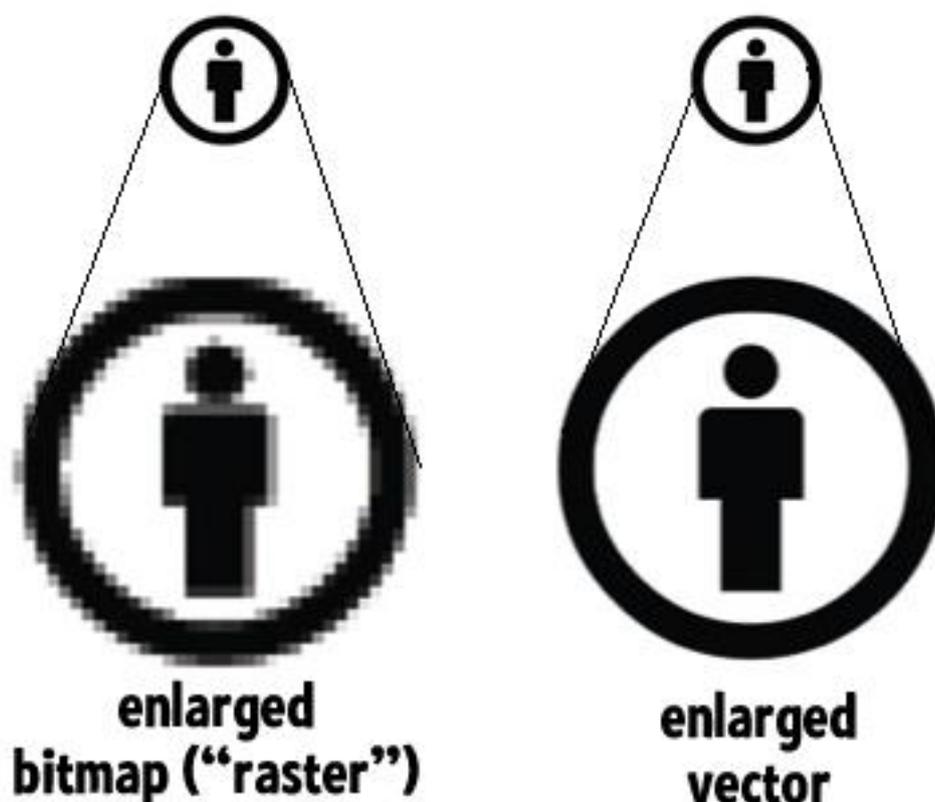


Figura 48. Diferencia de pérdida de calidad en el escalado en bitmaps con respecto al escalado en imágenes en formato vector [a.21]

La conversión del juego a modo de 2 jugadores, conlleva también una gran cantidad de cambios en su formato y diseño funcional que implican la necesidad de incluir nuevas funciones de gestión de objetos. Durante la adaptación se evalúa cada línea del código para simplificar y reducir el número de funciones y acciones que se van a demandar al procesador gráfico durante la ejecución de la aplicación.

Eventos multitáctiles (multitouch)

La capacidad de los dispositivos móviles y sistemas operativos iOS y Android de gestionar varios eventos táctiles simultáneamente es crucial para el desarrollo de este proyecto, ya que se trata de un juego de dos jugadores que pulsan la pantalla y generan eventos de arrastrado a la vez en el transcurso de la misma partida.



Figura 49. Evento estándar "Zoom" que requiere capacidad multitáctil del dispositivo [a.22]

Partiendo de que los dispositivos y sistemas modernos disponen de capacidad multitáctil, en este proyecto se rediseña completamente el sistema de gestión de eventos de ratón del juego del que se parte, sustituyéndolo por uno nuevo que responde a eventos táctiles.

En el nuevo sistema, se añade un parámetro touchID (identificador) a cada pulsación que se produce en la pantalla. De esta manera se mantiene referencia por separado de cada evento táctil que se produce, pudiendo gestionar independientemente, por ejemplo, que un usuario arrastre un objeto mientras el otro pulsa en un menú en la otra parte de la pantalla.

Eventos Over

Entendemos por un evento "over" el que se produce cuando el puntero del ratón está sobre un objeto. Este tipo de evento no existe en pantallas táctiles, ya que el dispositivo no puede reconocer si el dedo está sobre un objeto sin tocarlo.

Debido a esto, la información gráfica y textual que se ofrecía en el juego original, ha de ser mostrada de diferente manera, y para ello, se han distribuido todos los procesos informativos del antiguo evento "over" de forma que se

muestran al hacer una pulsación inicial, exigiendo una última pulsación de confirmación.

Tamaño de la interfaz

Para ajustar la comunicación del usuario con la interfaz y adaptarla a pantallas más pequeñas que la de un PC, se agrandan diferentes elementos gráficamente, y se desarrolla una función para que los elementos pequeños dispongan de un área más grande alrededor en la que poder ser tocados.

Objetivo

Todos estos procesos de optimización pretenden conseguir una mejora máxima en el rendimiento, permitiendo ofrecer una experiencia de usuario fluida en cualquier dispositivo.

4.2.2 Arquitectura de la aplicación

La arquitectura de la aplicación se basa en el patrón arquitectónico MVC (Modelo Vista Controlador). Siguiendo el modelo, se divide la aplicación en tres bloques diferenciados, separando datos, interfaz y lógica de negocio (especificaciones de la aplicación).

Este modelo permite dividir fácilmente al equipo de desarrollo y posibilitar la creación de librerías de uso para el sistema, que también serían válidas y reusables en desarrollos futuros.

A continuación se muestra esquemáticamente el patrón arquitectónico MVC, donde las líneas sólidas representan una relación directa y las discontinuas una indirecta.

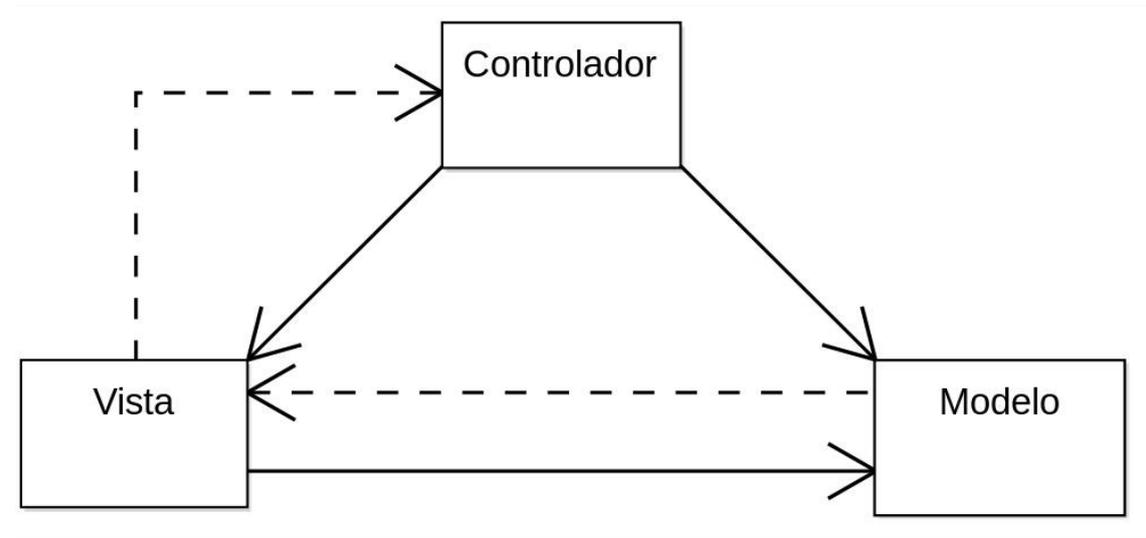


Figura 50. Esquema del patrón MVC [11]

Modelo: Gestiona los accesos a la información con la que el sistema opera (consultas y actualizaciones) y los privilegios de acceso descritos en la lógica del negocio. Las peticiones de acceso las envía el controlador.

Vista: A través de la interfaz de usuario, representa el modelo al usuario de forma adecuada para que éste pueda interactuar con él.

Controlador: Responde a las acciones del usuario y a cualquier tipo de evento para generar peticiones al modelo. Es el intermediario entre la vista y el modelo.

El flujo de control genérico para el MVC es el siguiente:

1. Interacción del usuario con la interfaz pulsando un botón.
2. La acción del usuario se recoge y es gestionada por el controlador.
3. El controlador actualiza el modelo con la nueva información.
4. El controlador envía a la vista los datos que debe mostrar.
5. La vista vuelve a entrar en espera de nuevas acciones del usuario para comenzar de nuevo el ciclo.

En el caso del sistema desarrollado en este proyecto, se puede estructurar el patrón MVC como se muestra en la siguiente figura, en la que el modelo es Flash CS6 y los ficheros de clase AS3 y el controlador Adobe AIR 3.7.

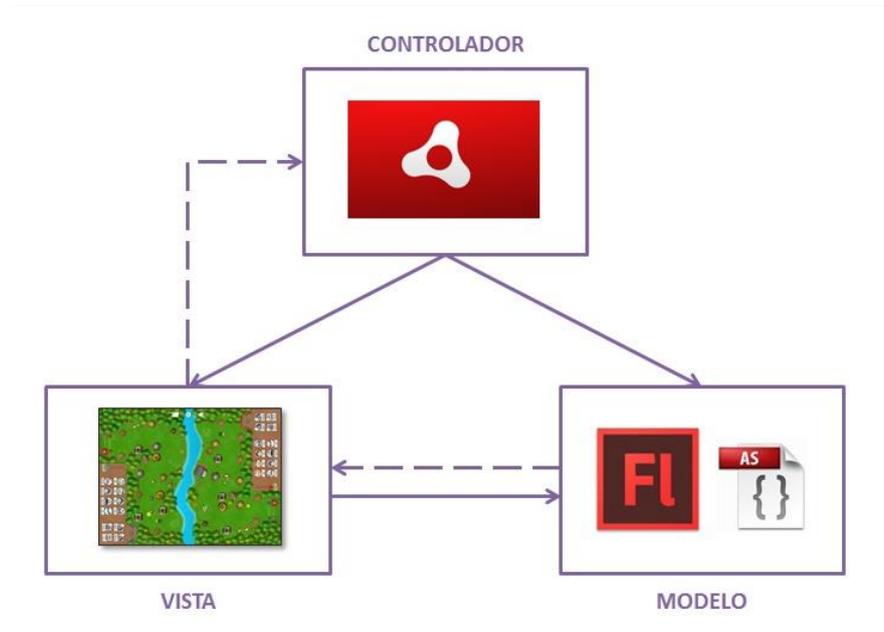


Figura 51. MVC del Sistema

4.2.3 Diagramas de flujo

Para facilitar una correcta comprensión del modelo antes de implementarlo, se desarrollan una serie de gráficos y diagramas que ofrecen una visión abstracta pero precisa de las diferentes partes del sistema.

Estos diagramas representan gráficamente procesos del sistema desde un punto de inicio a uno de finalización. Se emplea la siguiente simbología:

Óvalo azul: Inicio y fin del proceso.

Rectángulo azul: Actividad.

Rombo naranja: Decisión.

Paralelogramo inclinado verde: Gestión de datos.



Figura 52. Simbología en diagramas de flujo

Flujo general de la aplicación

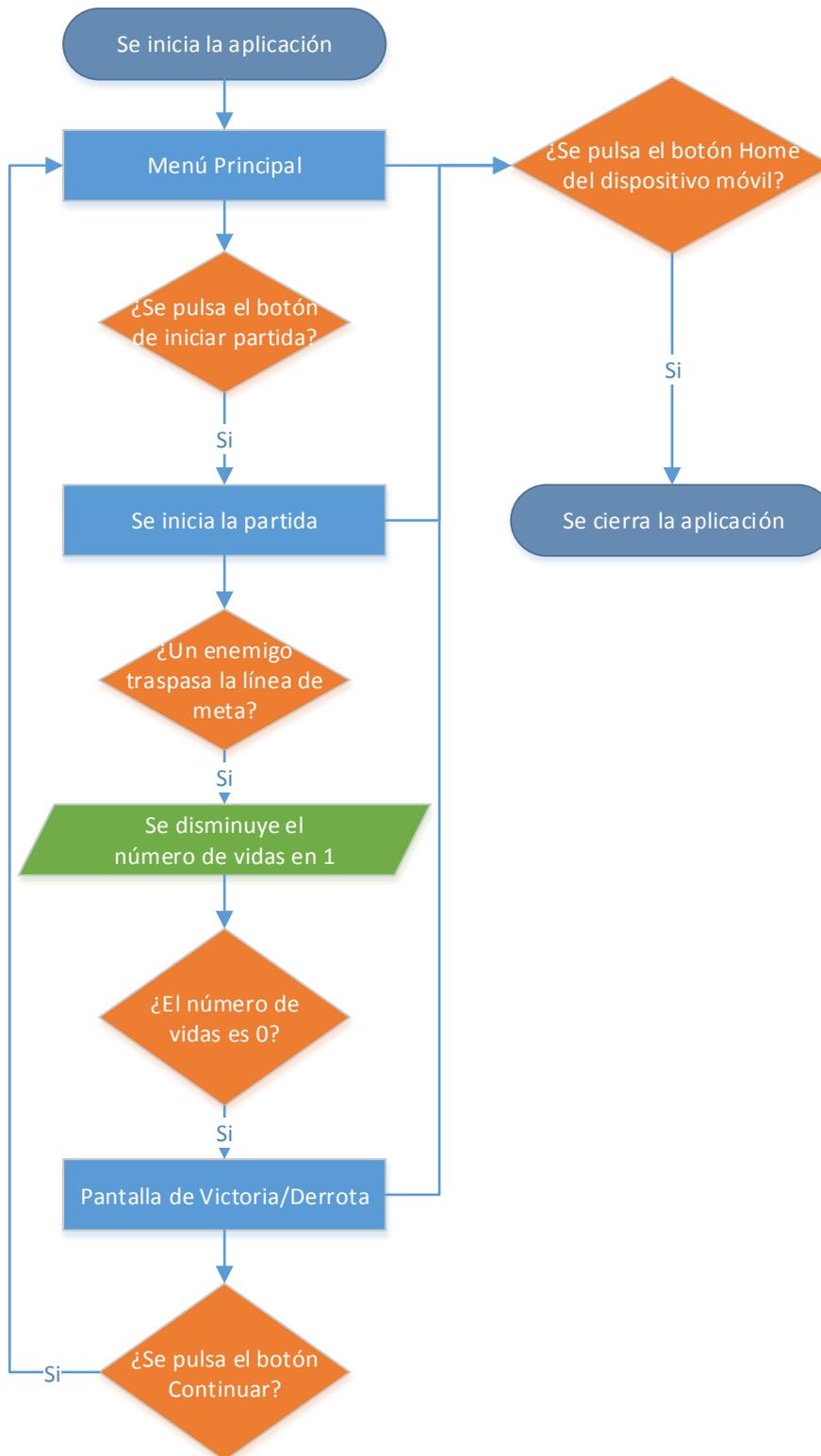


Figura 53. Diagrama de flujo general de la aplicación

Construcción y venta de torres

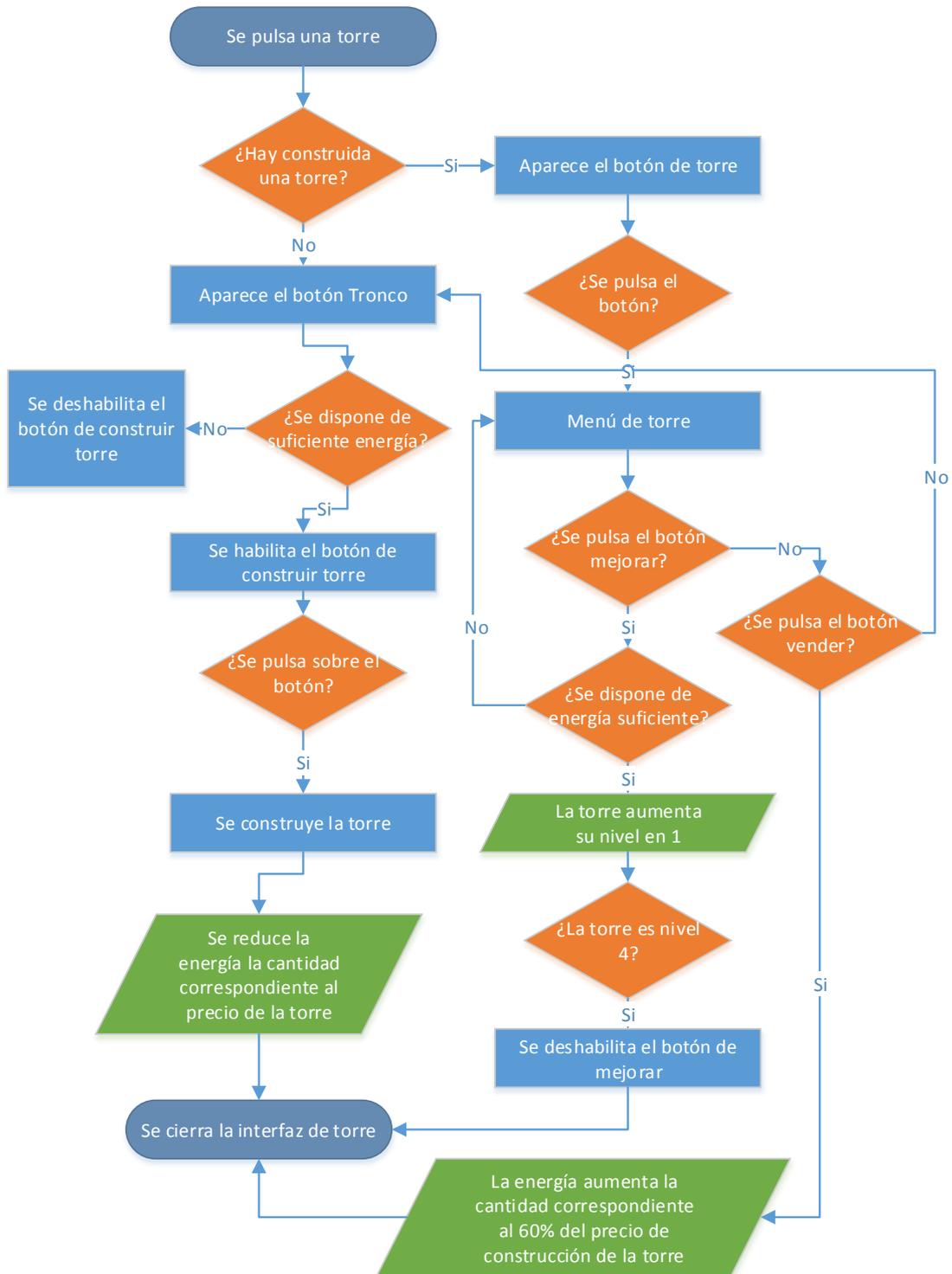


Figura 54. Diagrama de flujo de proceso de compra de torre

Habilidades Especiales

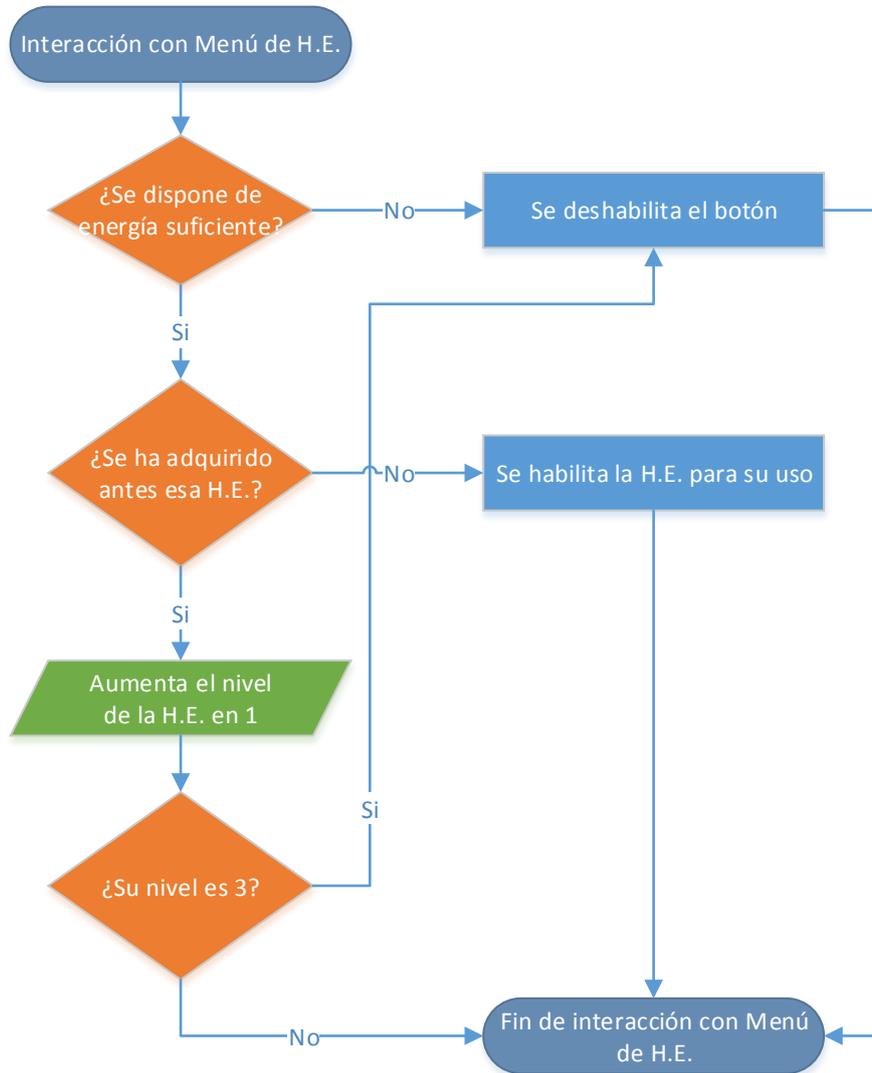


Figura 55. Diagramas de flujo de Habilidades Especiales

Envío de enemigos

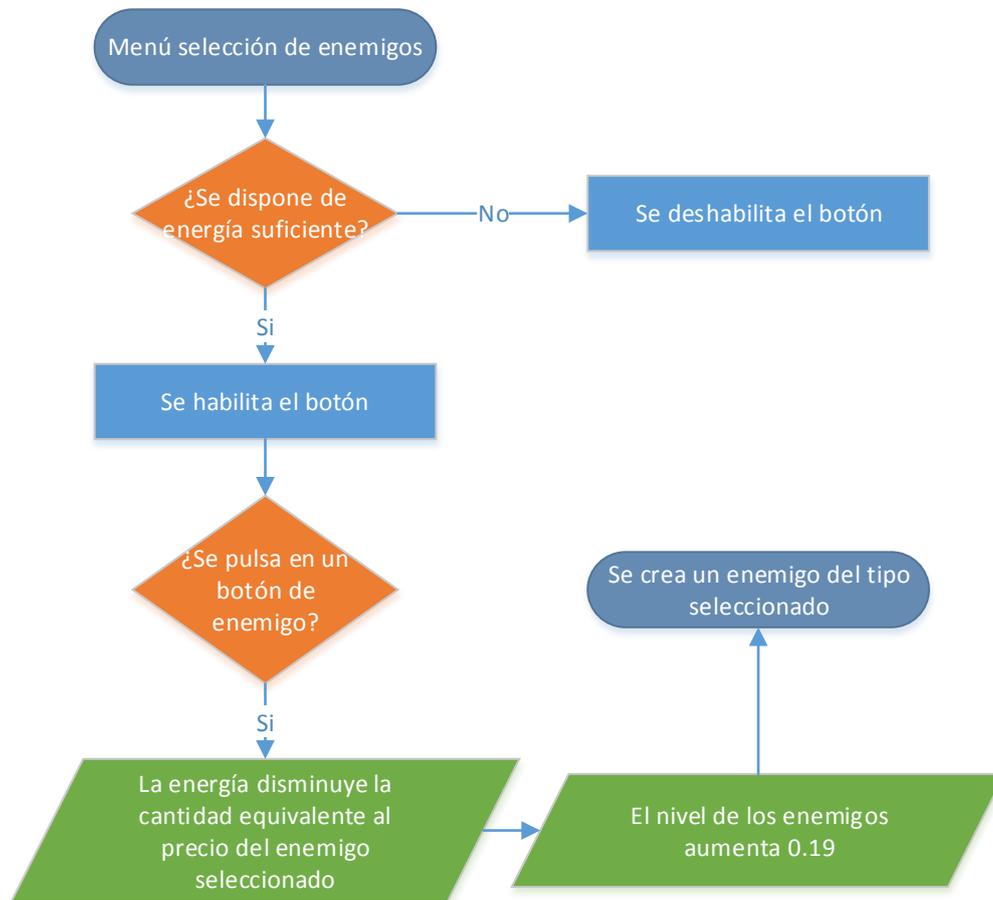


Figura 56 . Diagrama de flujo de envío de enemigos

Eliminación de enemigos

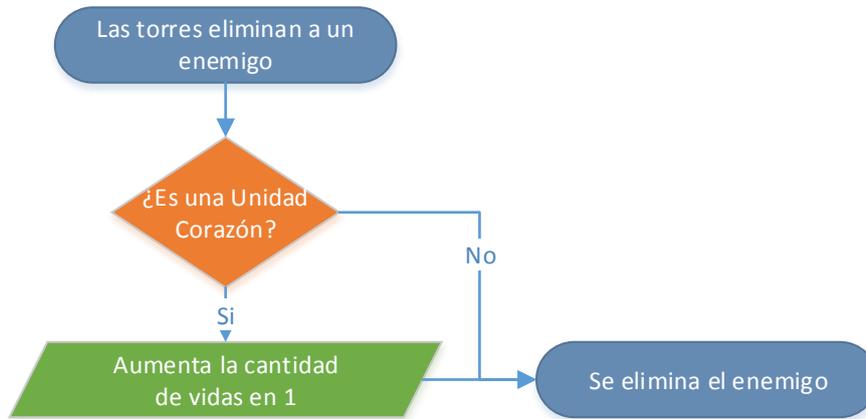


Figura 57. Diagrama de flujo de eliminación de enemigos

Estrellas

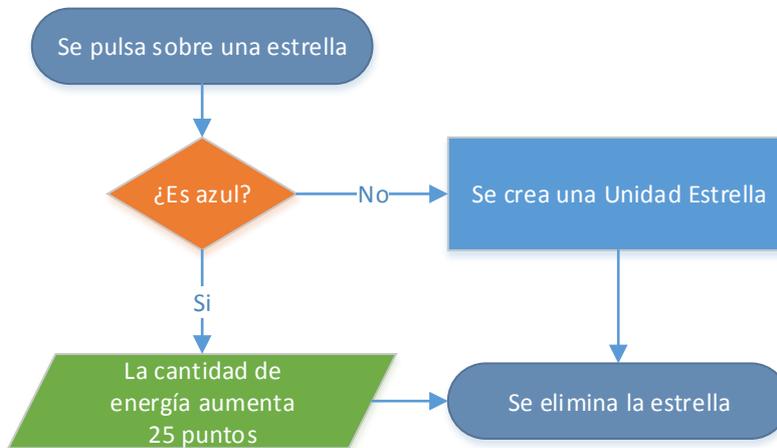


Figura 58. Diagrama de flujo de recolección de estrellas

4.2.4 Diseño de clases

El objetivo de esta sección es ilustrar mediante un diagrama de clases la estructura del sistema especificando las clases (con sus atributos y operaciones), y las relaciones entre ellas.

La información de esta sección servirá como referencia al programador a la hora de desarrollar la aplicación.

Diagrama de Clases

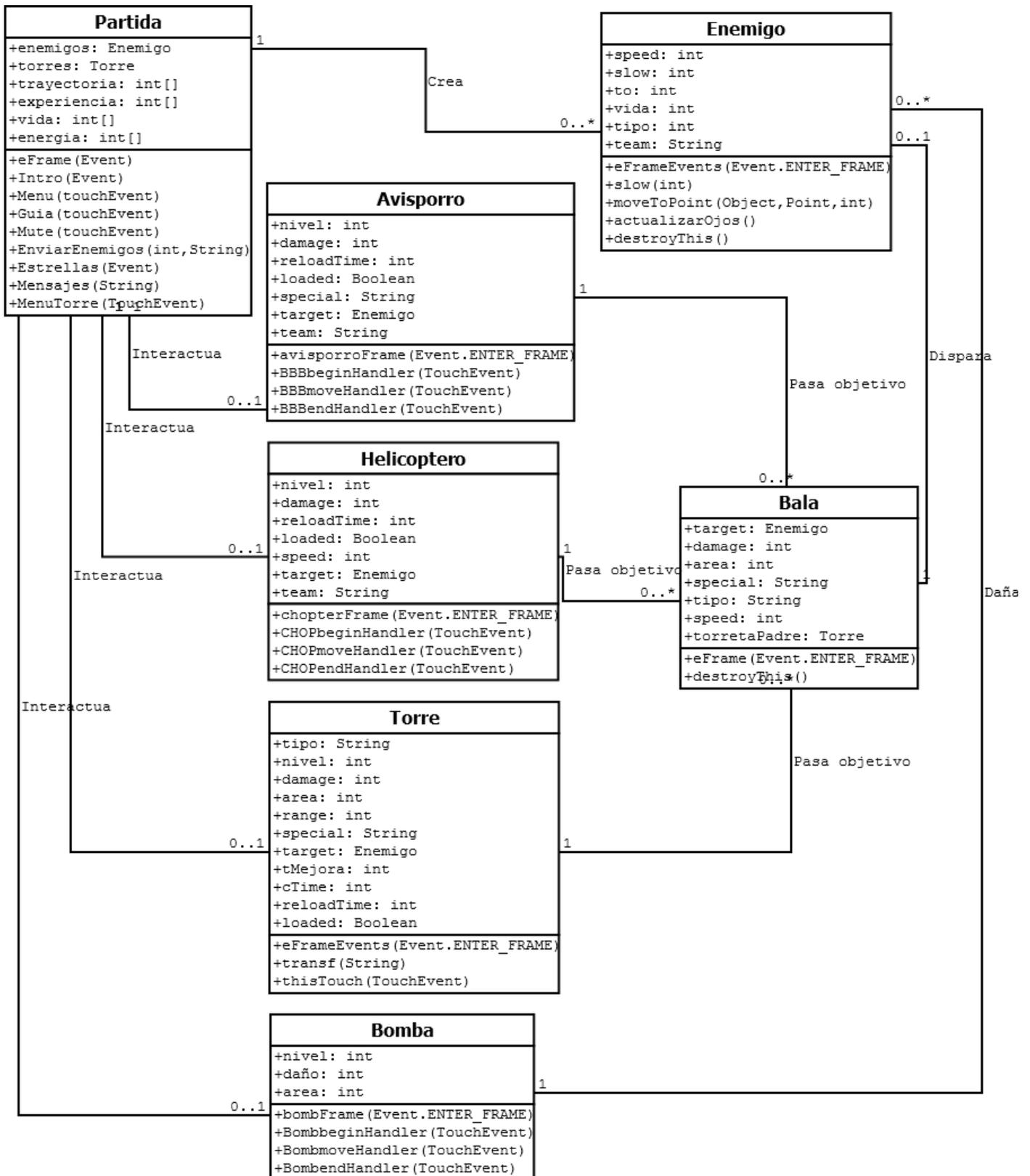


Figura 59. Diagrama de clases

A continuación se describen las responsabilidades, atributos y operaciones de cada clase.

Clase: Partida

Responsabilidades:	Gestiona los eventos y procesos que se desarrollan durante la partida.
Atributos:	<ul style="list-style-type: none">- enemigos: Enemigo. Conjunto de enemigos activos.- torres: Torre. Conjunto de torres activas.- trayectoria: int[.]. Recorrido de los enemigos.- experiencia: int[.]. Puntos de experiencia acumulados.- Vida: int[.]. Puntos de vida restantes.- Energía: int[.]. Energía disponible.
Operaciones:	<ul style="list-style-type: none">- eFrame (Event). Bucle principal de eventos.- Intro (Event). Cuenta atrás.- Menu (touchEvent). Menú Principal.- Guia (touchEvent). Menú de Guía.- Mute (touchEvent). Activar/Desactivar sonido.- EnviarEnemigos (int, String). Envío de enemigos.- Estrellas (Event). Caída de estrellas azules y amarillas.- Mensajes (String). Mensajes para jugadores.- MenuTorre (touchEvent). Compra/mejora de torres.

Figura 60. Definición de Clase: Partida

Clase: Avisporro

Responsabilidades:	Gestiona el manejo y evolución del objeto Avisporro.
Atributos:	<ul style="list-style-type: none">- nivel: int. Nivel de evolución de Avisporro.- damage: int. Daño. Depende del nivel.

- **reloadTime: int.** Tiempo de espera entre disparos.
- **loaded: Boolean.** Especifica si se ha cargado.
- **special: String.** Tipo de daño.
- **target: Enemigo.** Enemigo objetivo.
- **team: String.** Equipo que ha comprado el objeto.

Operaciones:

- **avisporroFrame (Event.ENTER_FRAME).** Bucle de eventos de la unidad especial.
- **BBBbeginHandler (TouchEvent).** Evento "Tocar".
- **BBBmoveHandler (TouchEvent).** Evento "Arrastrar".
- **BBBendHandler (TouchEvent).** Evento "Soltar".

Figura 61. Definición de Clase: Avisporro

Clase: Helicoptero

Responsabilidades:

Gestiona el manejo y evolución del objeto Helicóptero.

Atributos:

- **nivel: int.** Nivel de evolución del Helicóptero.
- **damage: int.** Daño. Depende del nivel.
- **reloadTime: int.** Tiempo de espera entre disparos.
- **loaded: Boolean.** Especifica si se ha cargado.
- **speed: int.** Velocidad de movimiento.
- **target: Enemigo.** Enemigo objetivo.
- **team: String.** Equipo que ha comprado el objeto.

Operaciones:

- **chopterFrame (Event.ENTER_FRAME).** Bucle de eventos de la unidad especial.
- **CHOPbeginHandler (TouchEvent).** Evento "Tocar".
- **CHOPmoveHandler (TouchEvent).** Evento "Arrastrar".
- **CHOPendHandler (TouchEvent).** Evento "Soltar".

Figura 62. Definición de Clase: Helicoptero

Clase: Torre

Responsabilidades: Gestiona el manejo y evolución de las torres.

- **tipo: String.** Tipo de torre.
- **nivel: int.** Nivel de evolución de la torre.
- **damage: int.** Daño. Depende del nivel.
- **area: int.** Area del daño. Depende del nivel.
- **range: int.** Rango de acción. Depende del nivel.
- **special: String.** Tipo de daño.
- **target: Enemigo.** Enemigo objetivo.
- **tMejora: int.** Tiempo necesario para evolucionar.
- **cTime: int.** Tiempo transcurrido en la evolución.
- **reloadTime: int.** Tiempo de espera entre disparos.
- **loaded: Boolean.** Especifica si se ha cargado.

Operaciones:

- **eFrameEvents (Event.ENTER_FRAME).** Bucle de eventos de cada torre.
- **trasf (String).** Gestiona transición de niveles de torre.
- **thisTouch (TouchEvent).** Evento "Tocar".

Figura 63. Definición de Clase: Torre

Clase: Bomba

Responsabilidades: Gestiona el manejo y evolución del objeto Bomba.

- **nivel: int.** Nivel de evolución de la bomba.
- **daño: int.** Daño. Depende del nivel.
- **area: int.** Area del daño. Depende del nivel.

Operaciones:

- **bombFrame (Event.ENTER_FRAME).** Bucle de eventos de la bomba.
- **BombbeginHandler (TouchEvent).** Evento "Tocar".

- **BombmoveHandler (TouchEvent).** Evento “Arrastrar”.
- **BombendHandler (TouchEvent).** Evento “Soltar”.

Figura 64. Definición de Clase: Bomba

Clase: Enemigo

Responsabilidades: Gestiona el movimiento y atributos de cada enemigo.

Atributos:

- **speed: int.** Velocidad de movimiento.
- **slow: int.** Efecto slow activado/desactivado.
- **to: int.** Próximo punto de destino.
- **vida: int.** Puntos de vida restantes.
- **tipo: int.** Tipo de enemigo.
- **team: String.** Equipo al que pertenece.

Operaciones:

- **eFrameEvents (Event.ENTER_FRAME).** Bucle de eventos de enemigos.
- **slow (int).** Efecto de ralentizado.
- **moveToPoint (Object, Point, int).** Movimiento hacia el siguiente punto y cambio de punto de destino.
- **actualizarOjos ().** Gestión de pintado de ojos según daño.
- **destroyThis ().** Eliminar objeto y borrar de memoria.

Figura 65. Definición de Clase: Enemigo

Clase: Bala

Responsabilidades: Gestiona el movimiento y atributos de cada bala.

Atributos:

- **target: Enemigo.** Enemigo objetivo.
- **damage: int.** Daño. Depende del nivel.
- **area: int.** Area de efecto del daño.

-
- **special: String.** Efecto especial.
 - **tipo: String.** Tipo de bala.
 - **speed: int.** Velocidad de movimiento.
 - **torrePadre: Torre.** Torre de origen.

Operaciones:

- **eFrame (Event.ENTER_FRAME).** Bucle de eventos de balas.
- **destroyThis ().** Eliminar objeto y borrar de memoria.

Figura 66. Definición de Clase: Bala

4.3 Implementación

La implementación es el proceso de nivel más bajo en el desarrollo de una aplicación y la fase definitiva en su creación. Gracias a los procedimientos previos de estudio, análisis y diseño, la implementación se puede realizar de forma guiada y escalonada, conociendo en todo momento el orden en el que se ha de trabajar y cuáles son los objetivos primarios y secundarios con sus respectivas prioridades.

4.3.1 Entorno

Gracias al tipo de entorno en el que se ha escogido trabajar, toda la aplicación se desarrolla con una misma herramienta, que gestiona tanto el código como los gráficos y el sonido. **Flash Professional CS6** es un entorno en el que hay que trabajar cuidadosamente para evitar problemas de rendimiento, pero permite un trabajo relativamente cómodo para proyectos de un único desarrollador.

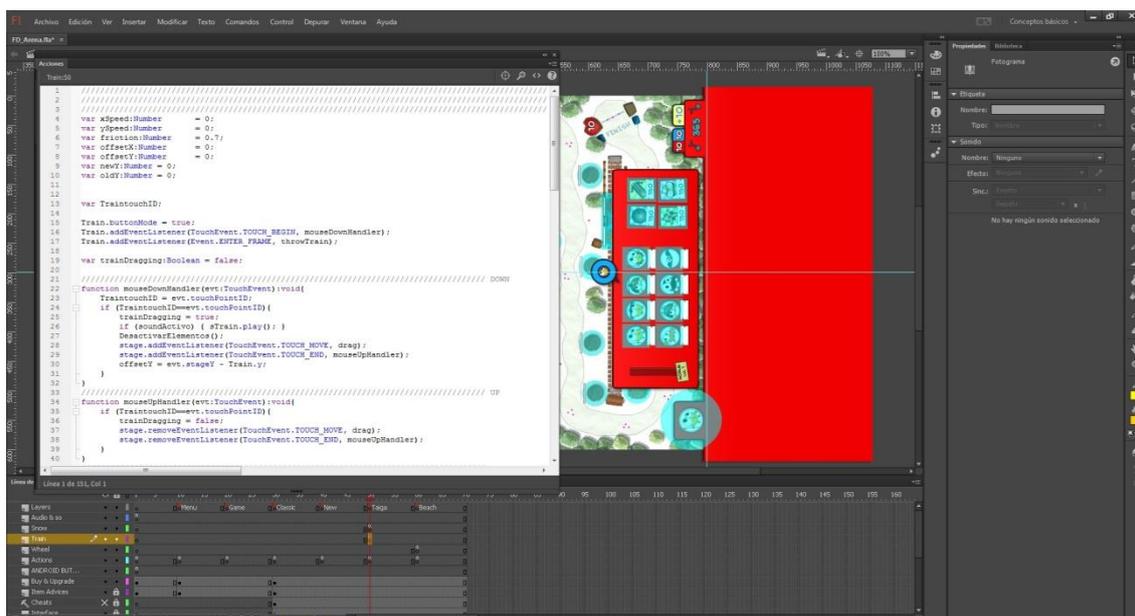


Figura 67. Layout de Flash

El lenguaje con el que se trabaja en este entorno es Actionscript 3, la versión más reciente y extendida de Actionscript, que es el lenguaje de programación orientada a objetos de Adobe para la plataforma Flash.

4.3.2 Gráficos y sonidos

Como ya se especificó en apartados anteriores, para el desarrollo de esta aplicación ya se cuenta con gráficos, sonidos y música. Todos estos elementos se han incluido dentro del proyecto en Flash, que genera automáticamente una clase para cada uno de ellos, a la que se puede invocar en cualquier momento desde el código.

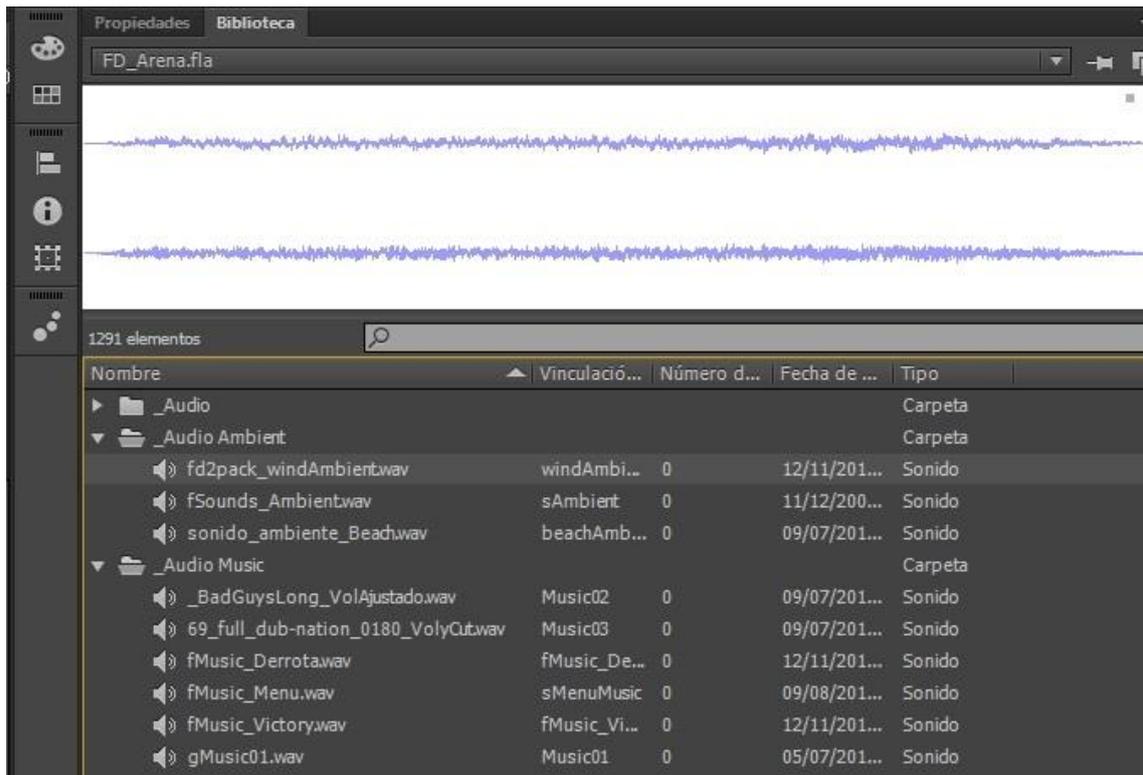


Figura 68. Librería de Flash para el proyecto

5 Guía de implantación

En este apartado se explica cómo instalar y configurar los programas necesarios para ejecutar la aplicación en un ordenador convencional.

Se da por hecho que el ordenador dispone ya de un sistema operativo Windows XP SP3 o superior con navegador Google Chrome, aunque se recomienda disponer de Windows 7 o Windows 8.

5.1 Instalación de Flash Professional CC

5.1.1 Nota previa sobre Adobe CS6 y CC

Tras la finalización del desarrollo del sistema que se documenta en esta memoria, implementado en Flash Professional CS6, Adobe hizo un cambio importante en su política de descarga y distribución de productos, de forma que el software de la familia CS6 ya no está disponible para descarga en sus servidores. Desde el lanzamiento de los productos CC, cualquier usuario de aplicaciones Adobe con sistema operativo de 64 bits, sólo puede descargar productos de esta familia.

Ya que la gama CS6 y CC son perfectamente compatibles, la guía de implantación describirá la instalación del sistema en el entorno CC, el único accesible en este momento para sistemas operativos 64 bits.

5.1.2 Descarga e instalación

1. En primer lugar, se debe llevar a cabo la descarga del producto desde la página oficial: <http://www.adobe.com/es/products/flash.html>.



Figura 69. Web de descarga de Adobe Flash CC

El sistema de pago actual de Adobe para sus aplicaciones es mediante cuota mensual. En cualquier caso, todos sus productos disponen de un periodo de prueba de 30 días en los que se puede hacer pleno uso funcional del programa.

En la página de descarga de Adobe Flash Professional CC, se pincha en el botón “Probar” de la barra lateral derecha. Se mostrará la página del centro de descargas de Adobe Creative Cloud.

2. Una vez en descargas, se clic en el producto **Flash Professional CC**.

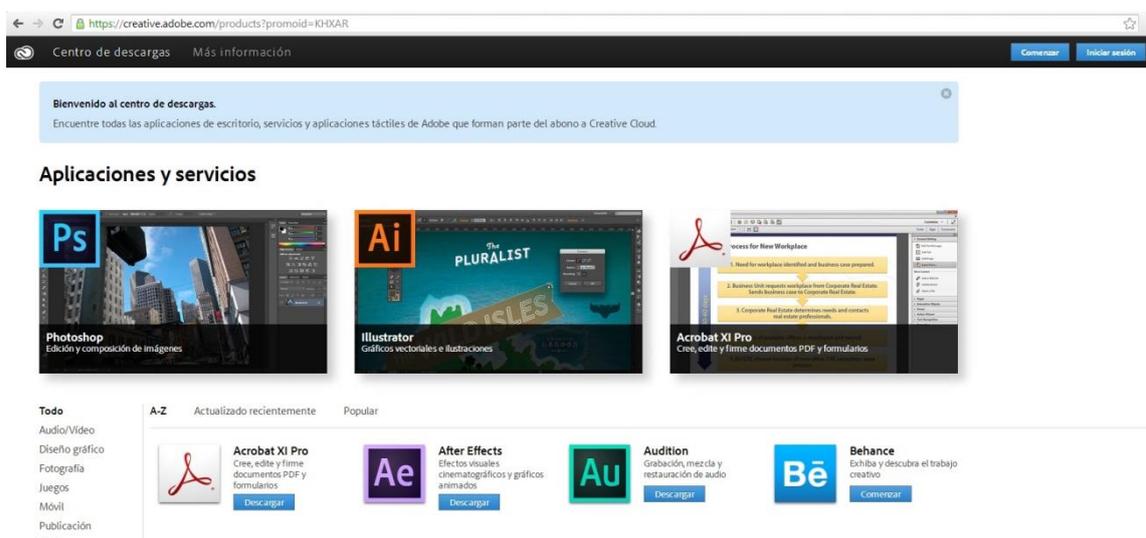


Figura 70. Centro de descargas de Adobe Creative Cloud

3. A continuación se clic en “Descargar la versión de prueba”. La descarga de la aplicación está sujeta a la creación de un usuario de adobe. Si no se dispone de uno, bastará con seguir los pasos especificados al pulsar el botón.

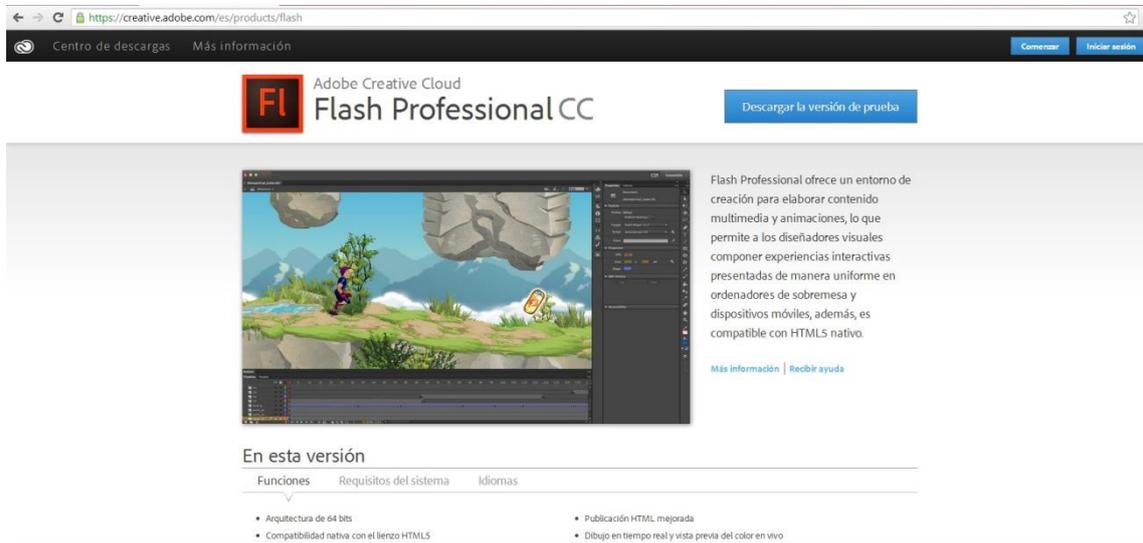


Figura 71. Descarga de Flash Professional CC

4. Se descargará en el ordenador un ejecutable de Creative Cloud que se abrirá para su instalación, tras la cual se abrirá Creative Cloud y se iniciará la descarga de Adobe Flash CC. Si no se inicia automáticamente, bastará con clicar en el botón “Probar” correspondiente a Flash CC de la pestaña “Apps”.

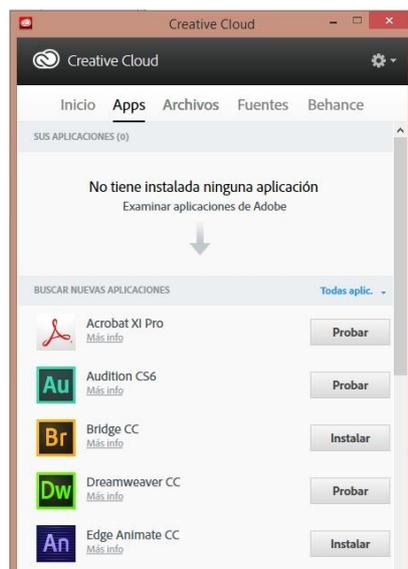


Figura 72. Ventana de Creative Cloud

5. Flash CC se descargará e instalará.

5.2 Instalación de Adobe AIR

Las aplicaciones y juegos desarrollados en el entorno Flash Professional CC, requieren del añadido de una extensión adicional para su compilación si el desarrollador desea publicar en plataformas móviles Android o Apple. Esta extensión se conoce como Adobe AIR sdk, y es de descarga gratuita, así como el motor necesario para la pre-visualización de este tipo de aplicaciones.

5.2.1 Adobe AIR

Está disponible para descarga en la página oficial de Adobe:

<http://get.adobe.com/es/air/>

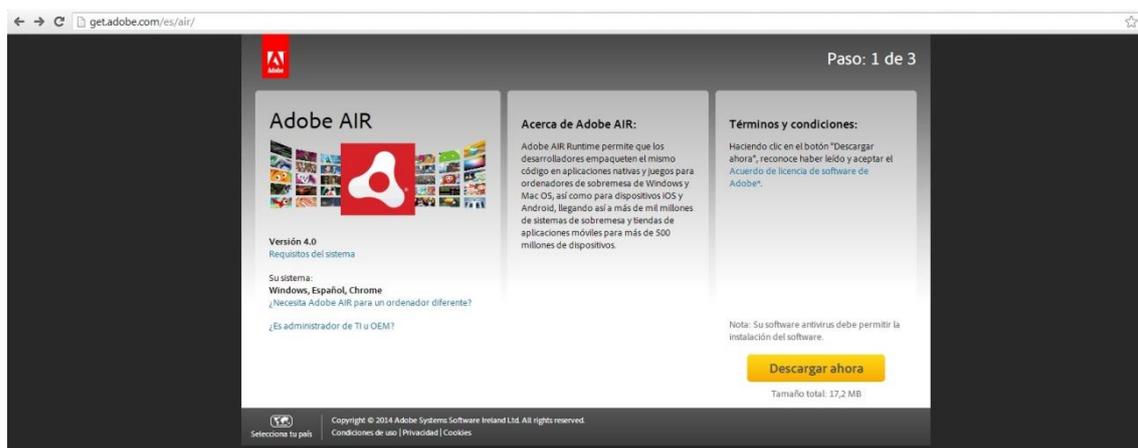


Figura 73. Descarga de Adobe AIR

Pulsar el botón “Descargar ahora” iniciará la descarga de un archivo instalador, que al ejecutarse guiará paso a paso la instalación del programa.

5.2.2 Adobe AIR sdk

La última versión disponible de la extensión se encuentra siempre disponible en el siguiente enlace: <http://www.adobe.com/devnet/air/air-sdk-download.html>.

1. Se ha de pulsar el botón “Download Now” de la versión correspondiente al sistema operativo Windows.

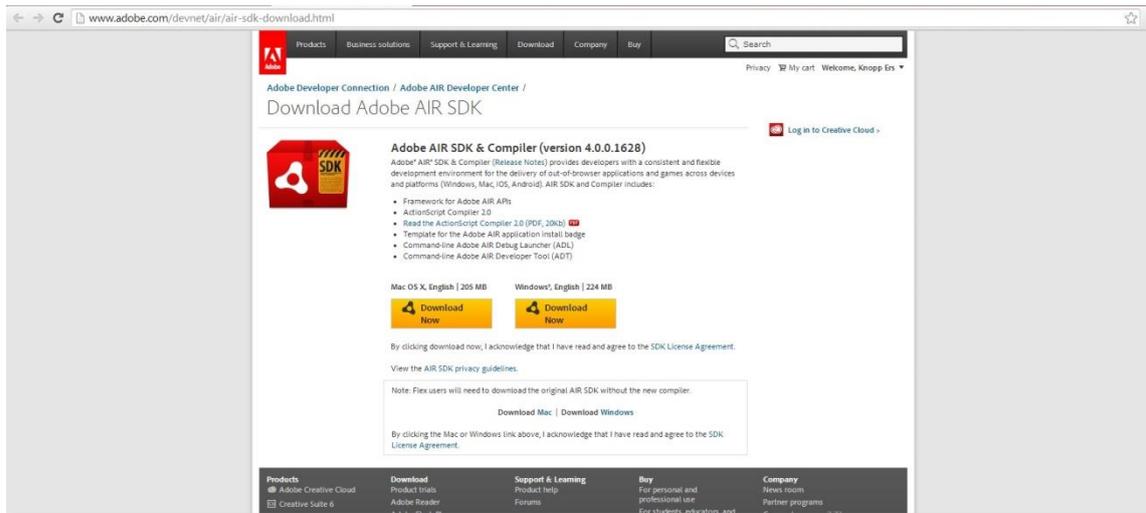


Figura 74. Descarga de Adobe AIR sdk

Se descargará un archivo comprimido en zip.

2. Se descomprime el archivo descargado en una carpeta con el nombre correspondiente a la versión de la sdk descargada para su fácil reconocimiento, en este caso AIR4.0.

3. Se mueve la carpeta recién creada al directorio de Adobe Flash CC, por defecto en: **C:/Archivos de programa/Adobe/Adobe Flash CC/**.

4. Se abre el programa Flash CC.

5. En la pestaña “Ayuda” se pincha en “Administrar SDK de Adobe AIR”.

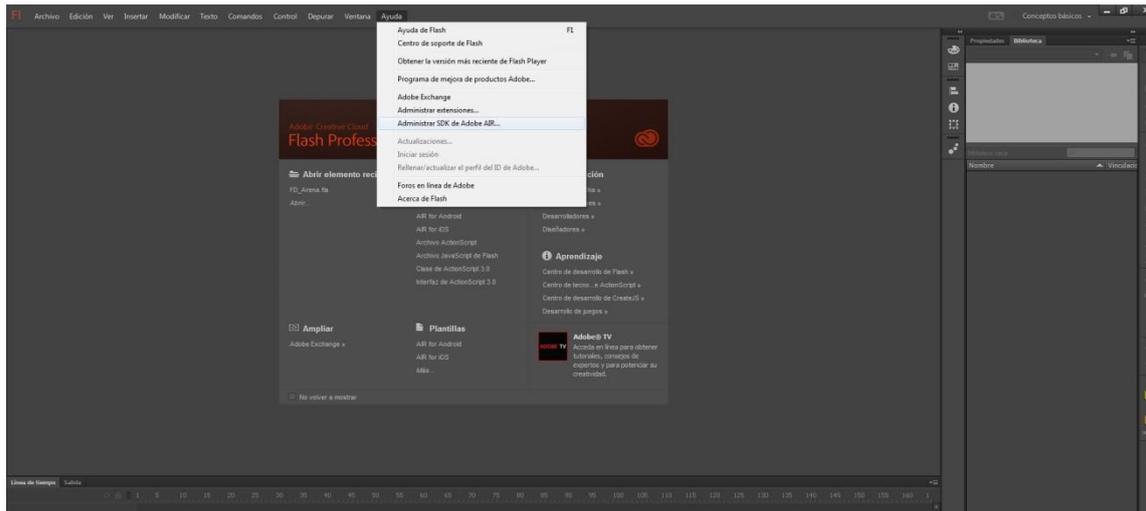


Figura 75. Cuadro de ayuda de Flash Pro CC

6. En la ventana que se muestra, se clic en el símbolo “+” y se añade la recién movida carpeta AIR4.0. Se mostrará en el cuadro una vez abierta.

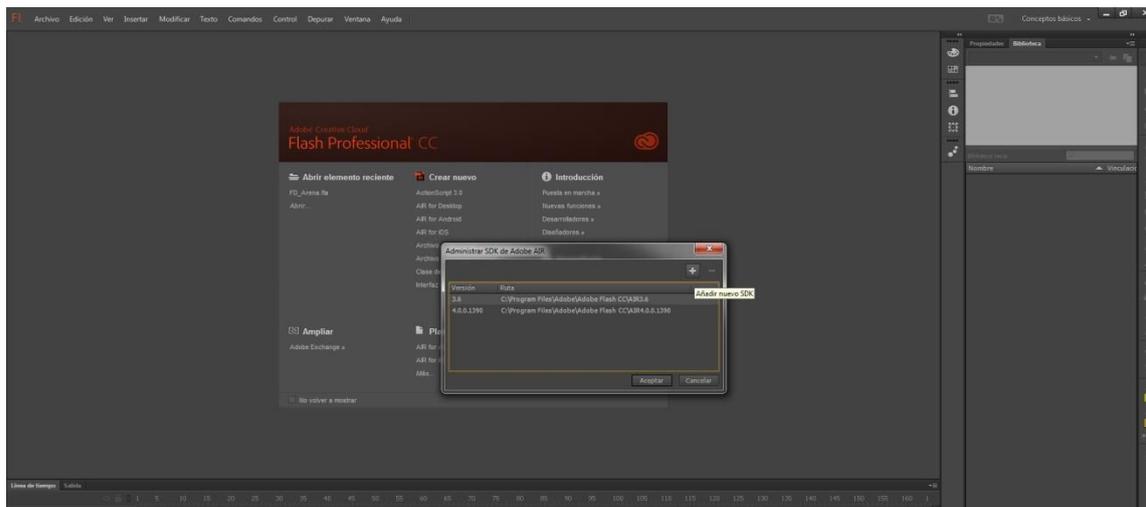


Figura 76. Administrador de sdk de AIR de Flash CC

5.3 Configuración de publicación

Para la correcta pre-visualización de la aplicación en Adobe AIR, hay una serie de parámetros que se deben configurar. Estos parámetros varían si se trata de una publicación final para los mercados de Apple o Android. En el caso de este proyecto, basta con comprobar el funcionamiento del juego en el simulador de Adobe AIR, para lo que emplearemos la pre-visualización de la aplicación en formato Android.

5.3.1 Configuración de parámetros

1. Con la aplicación abierta, en el cuadro de propiedades de la columna derecha se muestran las opciones genéricas de publicación. En el desplegable “Destino”, se selecciona la opción “AIR 4.0.0.1390 for Android” o la equivalente a la última versión de la sdk de AIR descargada.

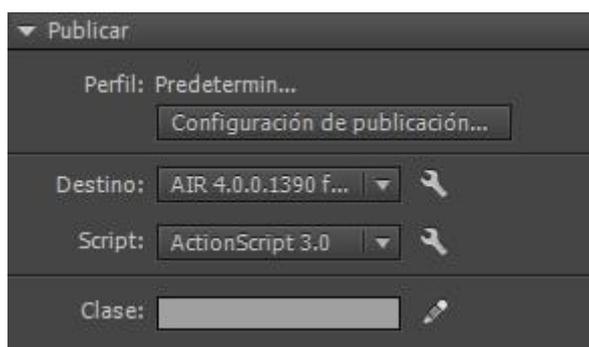


Figura 77. Configuración de publicación de Flash Pro CC

Una vez seleccionada, se clica en el icono de la llave inglesa a la derecha de la elección realizada.

2. En la pestaña “General”, para la pre-visualización, bastará con definir nombres genéricos de Archivo de salida, nombre e ID de la aplicación. La proporción de aspecto debe marcarse como “Horizontal” y el Modo de procesamiento en CPU.

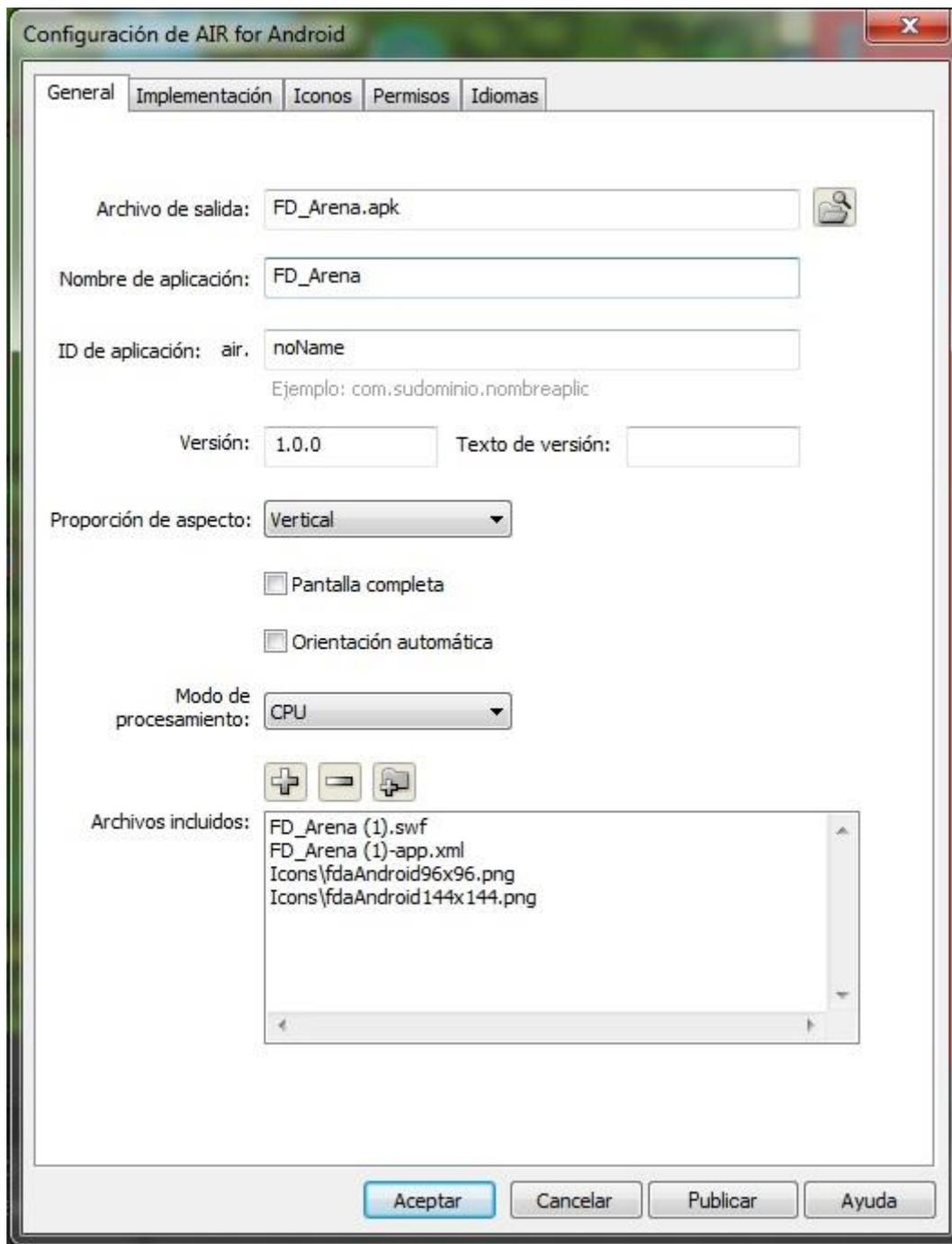


Figura 78. Configuración de Air para aplicación Android: General

3. En la pestaña “Implementación” se añadirá un certificado estándar para que se permita publicar la aplicación. Este archivo está incluido en la carpeta de la aplicación, dentro de la subcarpeta “Certificate”, con el nombre “CertfAndroid.p12”.

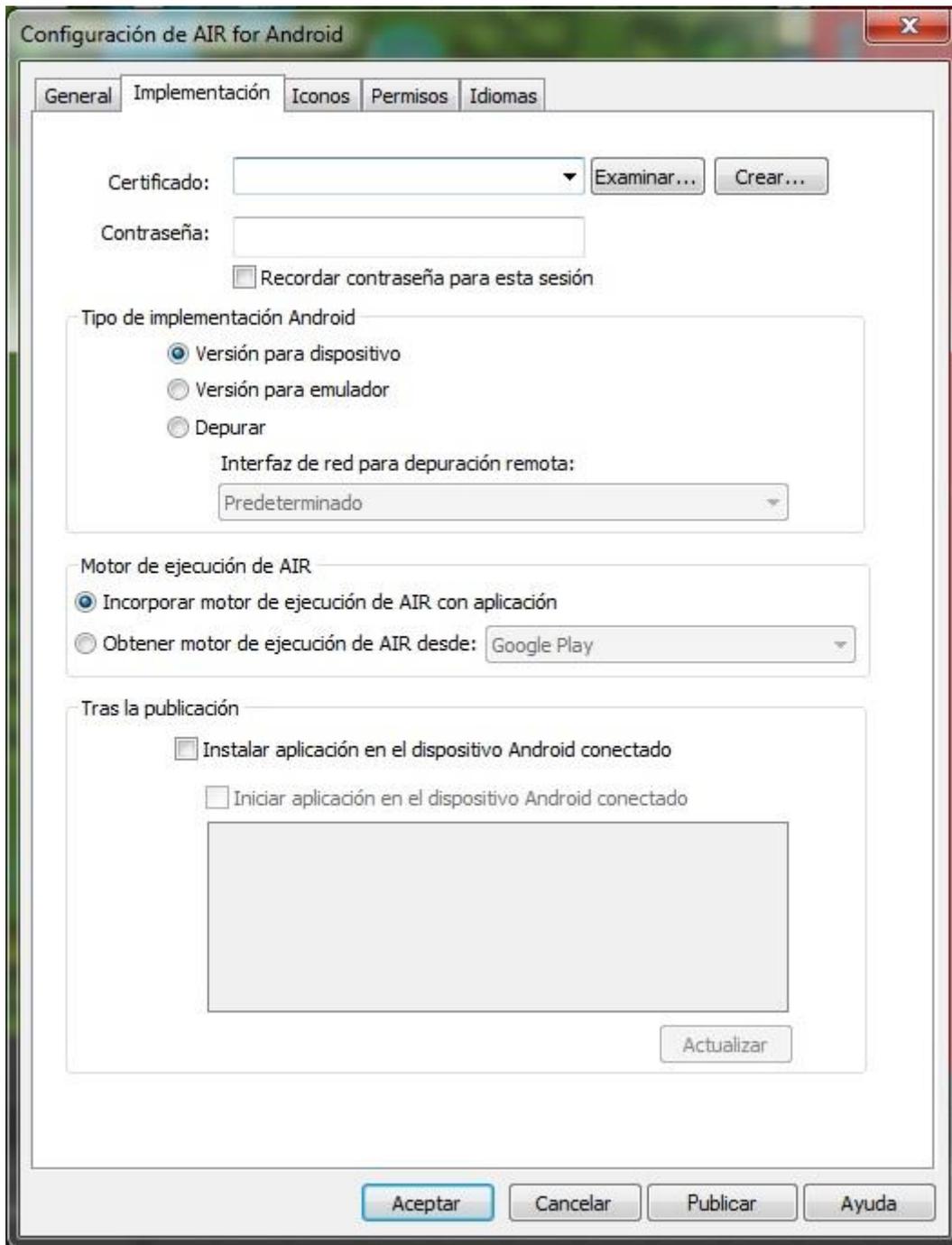


Figura 79. Configuración de Air para aplicación Android: Implementación

4. Se clic en "Aceptar" y la aplicación ya está lista para ser testada ejecutando el comando de teclado "Ctrl+Enter".

5.3.2 Configuración del simulador

Una vez se ejecuta la pre-visualización de la aplicación, para que el simulador interprete los clics del ratón como eventos táctiles de tocar pantalla, tal como se producirían en una tableta, se debe marcar un último parámetro.

En la ventana anexa a la aplicación que se abrirá en el modo simulador, se pulsa en la pestaña “Touch and gesture”, y dentro de ella se activa la opción “Touch layer”. A partir de ese momento, se podrá manejar la aplicación como si se hiciera desde una tableta.

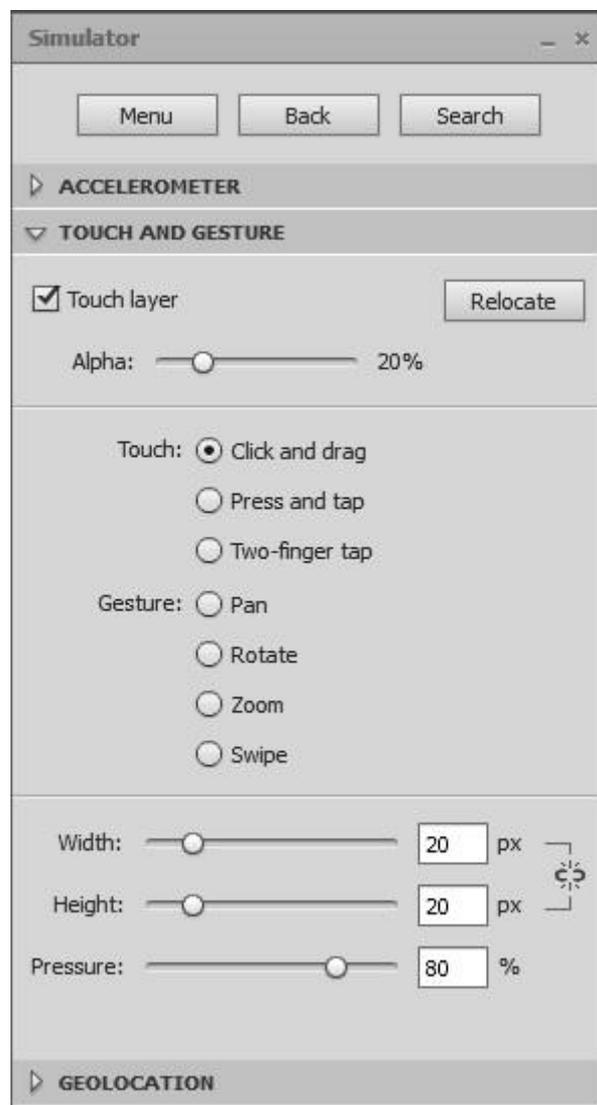


Figura 80. Configuración del Simulador de aplicaciones Android de AIR

5.4 Publicación en mercados

Una vez se ha comprobado que la aplicación es estable, no genera errores y no incumple de ninguna forma las normativas de aplicaciones de Apple o Google, se puede compilar el archivo final en .apk o .ipa (para Google o Apple) y hacerlo público, subiéndolo a los respectivos servidores de aplicaciones de las citadas compañías para que esté disponible en el Play Store y/o el App Store.

5.4.1 Cuentas de desarrollador

Para disponer de estos servicios de publicación, se ha de disponer de una cuenta de desarrollador tanto en Google como en Apple. Cada cual dispone de características específicas, y el precio de disponer de ellas varía de entre los 30€ totales de Google, a los 100€ anuales de Apple, pero su funcionamiento y propósito es similar.

Google Play



Se trata de un mercado más abierto, con menos restricciones a la hora de aprobar aplicaciones y libertad para subir y actualizar rápidamente nuevas compilaciones.

Su ventaja principal es la comodidad y rapidez con la que se puede publicar una nueva aplicación o actualizarla.

App Store



Apple es más restrictiva a la hora de decidir qué aplicaciones pueden aparecer en su mercado. Para cualquier nueva aplicación o actualización se ha de pasar por un proceso previo de comprobación y validación que lleva a cabo una persona de su equipo. Además, hay que crear un archivo de licencia general de desarrollador y otro específico para cada aplicación, que sólo pueden editarse en ordenadores Mac.

Esta clase de restricciones complican y retrasan el proceso de publicación, pero gracias a ellas, Apple consigue que la calidad general de las aplicaciones de su App Store sea muy alta y que se trate de un mercado fiable para el usuario final, que como respuesta es más receptivo y con más disposición a invertir dinero.

Por tanto, que un juego esté publicado en el App Store siempre indicará que se trata de un producto que cumple unos mínimos de calidad.

5.4.2 Creación de las cuentas

El proceso de creación de las cuentas de desarrollador es sencillo y válido para cualquier persona. El hecho de que la aplicación que se desea publicar sea de descarga gratuita evita además la necesidad de firmar acuerdos y contratos específicos para aplicaciones de pago.

Google Play

Para poder comprar el servicio, se ha de disponer de una cuenta básica gratuita de Google o Gmail. Con ella, se accede a la plataforma de desarrolladores en:

<https://play.google.com/apps/publish/signup/>

Una vez en esa pantalla, sólo hay que completar el proceso de pago y validación de datos personales y se conseguirá acceso instantáneo.

App Store

Al igual que en Google, se ha de disponer de un identificador válido de usuario estándar gratuito de Apple. Con él se accede a:

<https://developer.apple.com/devcenter/ios/index.action>

Siguiendo los pasos especificados y completando el pago, se conseguirá acceso a la plataforma de desarrollo.

5.4.3 Creación y subida de la aplicación

Antes de subir los archivos correspondientes de la aplicación, ha de crearse un registro de la misma en la página de desarrollador de cada servicio.

En este apartado se omite la muestra de enlaces o pantallazos, por disponer de identificadores reales de un usuario de pago que desean mostrarse de forma pública.

Google Play

En la consola de desarrollador de Google, basta con pinchar en “Nueva aplicación” y seguir los pasos que se muestran, enviando información sobre el título, descripción y características de la aplicación, así como el material de marketing correspondiente (capturas, imágenes promocionales e iconos).

También se puede restringir la publicación de la aplicación en países o dispositivos específicos.

Una vez creado el registro de la aplicación, para publicar el archivo .apk final firmado, se accede a la ventana de publicación de aplicación Android de Flash Professional, y en la pestaña certificado se clica en “Crear certificado”. Rellenando una información básica, se creará un archivo en formato .p12 que servirá para identificar al desarrollador de la aplicación.

Con el archivo final publicado, en la consola de desarrollo de Google Play, se clica en la sección “APK” dentro del juego creado. Esta sección ofrece una interfaz de subida para el archivo final exportado, que incluye información técnica sobre el producto y su compatibilidad con los diferentes dispositivos.

Una vez se ha subido el archivo, basta con “publicar”, y la aplicación estará disponible en el mercado tras unas horas de espera.

App Store

Dentro de la consola de desarrollo de aplicaciones iOS de Apple, se ha de empezar creando los certificados correspondientes.

Se accede a la sección “Certificados” y se siguen, en un ordenador Mac, los pasos indicados para crear tanto un Certificado de desarrollo, como un identificador de aplicación específico para la aplicación que se va a subir.

Una vez creados los certificados, se puede publicar la versión final firmada a través de la interfaz de publicación de Flash Professional, incluyendo los certificados en la pestaña específica para ello.

Con la versión final firmada publicada, se ha de acceder a la plataforma específica de creación de nuevas aplicaciones iOS de Apple: iTunes Connect:

<https://itunesconnect.apple.com/WebObjects/iTunesConnect.woa>

Con el identificador de desarrollador de Apple, se accede a una interfaz de subida de aplicaciones, en la que se introduce información similar a la del caso de Google, como es el título de la aplicación, descripción, material promocional, etc.

Una vez se ha creado el registro de la aplicación, se vuelve a necesitar acceso a un ordenador Mac, en el que ha de descargarse la aplicación gratuita "Application Loader" (no disponible para PC), cuyo enlace de descarga se muestra en la propia interfaz de iTunes Connect.

Esta aplicación permite subir desde un ordenador Mac, el archivo final firmado exportado desde Flash Professional y vincularlo al registro de aplicación ya creado.

Si se ha completado correctamente toda la información correspondiente a la aplicación y la subida del archivo final se ha finalizado con éxito, la consola de iTunes Connect permitirá la publicación de la aplicación, que no será instantánea porque pasará previamente por el proceso de validación antes citado, en el que se evaluarán la calidad y características del juego para definir si es apto para publicarse en el App Store.

Apple envía un correo al desarrollador una vez que la aplicación ha pasado este proceso y ha sido aprobada o rechazada, ofreciendo, en este último caso, información específica sobre el motivo de su rechazo y tareas a cumplir para que pueda ser aprobada en una posterior evaluación.

6 Manual de usuario

En este apartado se muestra la guía de utilización del juego. En ella se describen, ilustradas con imágenes, todas las funciones y características del producto para que cualquier usuario pueda aprender a manejarlo.

6.1 Introducción

6.1.1 Género

El juego es un Tower Defense de dos jugadores que se enfrentan en batallas uno contra uno, en las que cada jugador se encarga de enviar enemigos a su rival y defenderse de los que él le envía, con el objetivo final de conseguir que 10 de los enemigos que se envían al rival, recorran el camino que hay entre la salida y la meta de su porción del escenario.

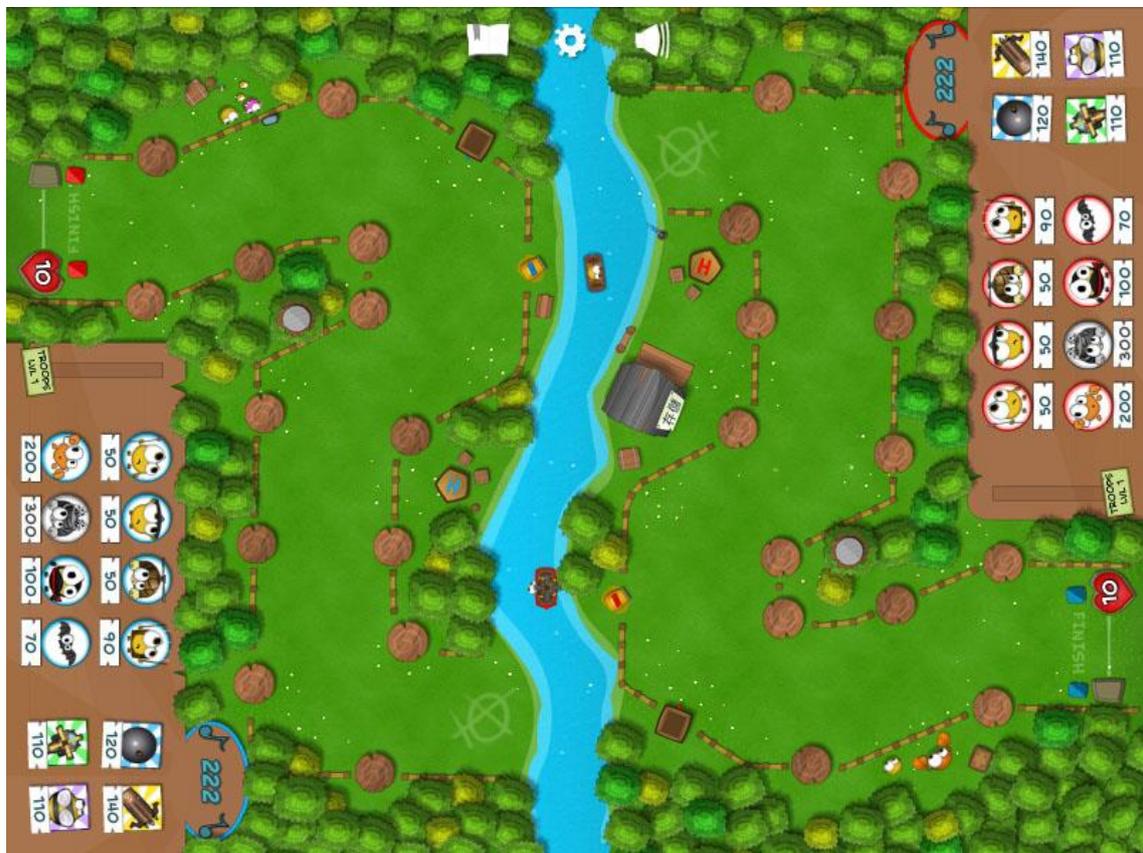


Figura 81. Vista general de la pantalla de batalla

Como en cualquier Tower Defense, cada jugador ha de fortificar su escenario para eliminar a los progresivamente más enérgicos enemigos que se acercan a él. En este juego la mecánica es básicamente la misma, con el valor adicional de que no todos los enemigos son generados por el juego, sino por el rival, al que además, habrá que derrotar gestionando los enemigos que van a atacarle a él.

6.1.2 Pantalla

El juego está planteado para dos jugadores, que pueden manejarlo pulsando en la pantalla en sus respectivas interfaces de usuario de forma simultánea.

En el transcurso de la partida, cada jugador dispondrá de la mitad de la pantalla del dispositivo para gestionar sus maniobras de ataque y defensa. La porción de la pantalla del rival se respeta como su zona de manejo y no se interactúa con ella, pero sí puede visualizarse para controlar su estado.



Figura 82. Posición de los jugadores

Posición de los jugadores

Para mayor comodidad de los jugadores, el dispositivo ha de situarse físicamente en posición vertical enfrentando a un jugador con el otro.

Menús

Durante las pantallas de menú, es el jugador principal el que maneja la aplicación y gestiona cuándo iniciar una partida o cambiar de menú.

Guía

La guía y opciones de juego accesibles desde el modo batalla se muestran en posición horizontal para que ambos jugadores puedan leerla con facilidad.

6.1.3 Elementos del escenario

El escenario de batalla está compuesto de multitud de elementos, fácilmente identificables. Pueden ser decorativos, informativos o dinámicos. A continuación se enumeran los que disponen de especial importancia durante la batalla.

Árboles y elementos decorativos



La decoración principal del escenario está ambientada en un bosque arbolado, cuya estructura condiciona el movimiento de los enemigos, que siguen un camino determinado por la vegetación. El escenario también está decorado con elementos como vallas o cajas.

El río



El escenario, para identificar claramente la porción de la pantalla dedicada a cada jugador, está dividido por un río, que representa el límite entre el área del jugador 1, y la del jugador 2.

Las barcas



Además de como elemento decorativo de ayuda de identificación de las áreas de juego, el río muestra la aproximación de los enemigos que se generan de forma automática cada 30 segundos. Estos enemigos se aproximan al punto de salida de cada jugador navegando en barcasas, en las que se les puede identificar gráficamente.

El Camino



Delimitado por árboles, vallas y otros objetos de decoración, el camino es la zona verde desarbolada de cada porción del mapa por la que avanzan los enemigos.

Puntos de salida



Se identifican con una forma abstracta pintada en la parte del camino cercana al río. En ellos se generan tanto los enemigos automáticos como los enviados por el rival.

Puntos de meta



Se sitúan en la parte inferior izquierda de la pantalla de cada jugador, representados con piedras de color y una línea con la palabra “Finish”.

Troncos



Se ubican en diferentes puntos de los bordes del camino. Son los elementos interactivos en los que el jugador puede construir torres. Al pulsar sobre ellos, se abre el menú de compra de torres.

Interfaz de envío de enemigos



Se muestra en la parte inferior derecha del área de juego de cada jugador. En ella se encuentran los botones de envío de cada tipo de enemigo y la barra informativa del nivel de los mismos, que se incrementa a medida que se envían.

Puntos de salida de habilidades especiales



Son los puntos en los que aparecerán los elementos especiales al ser comprados por el jugador. Aunque se posición se anuncia con una interfaz específica cuando aparecen, estos puntos decorativos anticipan su lugar de salida.

6.2 Manejo básico

En esta sección se explican con detalle cada una de las acciones disponibles para los jugadores durante el transcurso de la partida. Cada jugador tiene acceso a ellas en la porción de pantalla destinada a su escenario, con una interfaz intuitiva y accesible. El acceso a ellas es a través de acciones táctiles en el dispositivo.

6.2.1 Construir torre

Cuando un jugador pulsa sobre un tronco, la interfaz de construcción de torre aparece en torno a ese elemento. Esta interfaz muestra iconos con los cuatro tipos de torres disponibles para comprar y el precio de cada una de ellas. Cuando el jugador pulsa en uno de los iconos, si dispone de energía suficiente, el tronco es sustituido por la torre representada por ese icono, haciendo desaparecer el menú de compra de torre. Mientras el menú de compra de torre está abierto, cualquier pulsación en otro lugar de la pantalla lo volverá a ocultar.



Figura 83. Interfaz de construcción de torre

6.2.2 Mejorar y vender torre

Un tronco que ha sido sustituido por una torre, adquiere el comportamiento de torre. Si el jugador pulsa en ella, se le muestran los iconos de mejora y venta de torre, que permiten respectivamente evolucionar las capacidades de la torre o venderla a cambio de un 65% de la energía que costó construirla y mejorarla, esto último siempre y cuando haya sido mejorada anteriormente.



Figura 84. Interfaz de mejora de torre

6.2.3 Enviar enemigo

Todos los enemigos disponibles para enviar al rival se muestran en iconos en la interfaz de envío de enemigos, cada uno de ellos con su precio asociado. Cada vez que un jugador pulsa en un icono de enemigo de su interfaz, si dispone de energía suficiente para comprarlo, el enemigo aparece inmediatamente en el punto de salida del escenario rival.



Figura 85. Interfaz de envío de enemigos

Tiempo de espera entre envíos

Cada enemigo enviado, tiene un tiempo de espera posterior a su envío asociado a su tipo. Durante este tiempo, el envío de enemigos queda bloqueado para evitar la excesiva reutilización de esta capacidad de ataque. Es un tiempo variable según el tipo de enemigo que se envía y se representa gráficamente encima de la interfaz de envío.



Figura 86. Tiempo de espera entre enemigos

6.2.4 Comprar habilidad especial

En la parte derecha de la interfaz de envío de enemigos se ofrecen iconos de las posibles habilidades especiales para comprar junto con su precio en energía. Si el jugador pulsa en uno de ellos y dispone de energía suficiente, el objeto correspondiente a la habilidad pulsada aparece en el escenario y puede ser utilizado por él durante el resto de la partida.



Figura 87. Habilidades especiales

6.2.5 Mejorar habilidad especial

Cuando una habilidad ha sido comprada, su icono evoluciona mostrando que su posterior pulsación ofrecerá una mejora al objeto especial correspondiente, siempre y cuando se disponga de la energía necesaria.



Figura 88. Mejora de habilidades especiales

6.2.6 Manejo de habilidades

Las habilidades especiales del juego responden a eventos táctiles, la mayoría de ellos del tipo “drag and drop” (arrastrar y soltar). A continuación se explica el funcionamiento de todas ellas.

Gatling



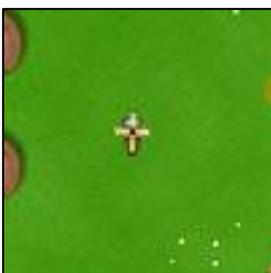
La torreta Gatling se sitúa automáticamente en una posición concreta del escenario específica para ella y al igual que las torres, dispone de un radio de acción determinado. Una vez construida, dispara por sí misma sin requerir acción del jugador.

Bomba



La bomba puede ser lanzada manualmente por el jugador en cualquier parte de su área de juego. Para ello debe pulsar en ella y arrastrar el dedo a la posición en la que desea lanzarla. Cuando explota, vuelve a estar disponible pasados 20 segundos.

Helicóptero



El helicóptero se desplaza con velocidad asociada a su nivel de evolución hacia un punto variable definido por el usuario. Una vez se sitúa en este punto, permanece inmóvil pero activo hasta que se le vuelve a proporcionar un nuevo punto de movimiento. Para mover este punto, el jugador ha de pulsar en el helicóptero y arrastrar el dedo hasta el nuevo destino hacia el que desea que vuele.

Avispa



La avispa se maneja pulsando en ella y arrastrando el dedo hasta donde se la desea situar. Al contrario que el helicóptero, su posición se actualiza instantáneamente siguiendo al dedo, así que el jugador puede desplazarla rápidamente a cualquier punto del mapa.

6.2.7 Opciones y guía de batalla

En el centro de la pantalla, en el lado izquierdo del jugador principal, se encuentran los botones comunes de guía de juego, opciones y mute. Son accesibles para cualquier jugador.



Figura 89. Botones de configuración

6.3 Elementos de juego

En este apartado se enumeran y describen los elementos principales con los que ambos jugadores van a manejarse durante la batalla. Su conocimiento y la manera en la que se ponen en juego determinan el transcurso de la partida en favor de un jugador o de otro.

6.3.1 Energía

La energía es el único recurso económico del juego. Todos los elementos del juego, tanto torres u habilidades especiales, como enemigos, tienen un coste en energía, que supone el precio que un jugador debe pagar para disponer de ellos.



Figura 90. Energía

La energía se muestra en la parte superior derecha de la interfaz de envío de enemigos y puede ascender de 3 maneras:

- **Automáticamente.** Aumenta 3 puntos cada 2 segundos para cada jugador, o 5 puntos si su nivel de enemigos ha llegado a 10.
- **Al destruir enemigos.** Cada enemigo otorga al jugador que lo elimina, un 60% del precio que le costó a su rival enviarlo.
- **Vendiendo torres.** La venta de una torre la destruye, otorgando al jugador un 65% del precio que costó construirla si está en nivel 1, o un 65% del precio que costó construirla y mejorarla si se encuentra en nivel mayor que 1.

6.3.2 Estrellas

Durante el transcurso de la batalla, dos tipos de estrellas caen en el escenario de los dos jugadores y se mantienen activas durante unos segundos hasta que desaparecen. Si durante ese tiempo el jugador pulsa sobre ellas, consigue beneficios especiales.



Estrellas azules

Las estrellas azules aparecen transcurridos intervalos de tiempo aleatorios en cualquier parte del escenario. Al pulsar una estrella azul caída en su área del mapa, el jugador obtiene un incremento extra de 25 puntos de energía.



Estrellas amarillas

Las estrellas amarillas tienen un 15% de probabilidades de aparecer en un punto aleatorio del escenario cada vez que un enemigo es destruido por cualquiera de los jugadores. Al pulsar en una estrella amarilla caída en su área del mapa, el jugador envía a su rival un enemigo estrella, explicado más adelante en el apartado “Unidades”.

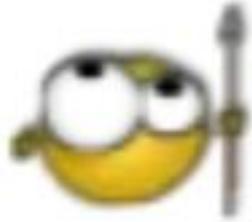
6.3.3 Unidades

Las unidades son el elemento de ataque principal. Se muestran en la interfaz de envío de unidades. Su nivel (de 0-10), que modifica progresivamente su capacidad de aguante, aumenta a medida que un jugador envía enemigos a su rival, de forma que cada 4 unidades enviadas, el nivel general de las siguientes unidades que se envían aumenta un punto. El incremento de nivel es definitivo, no puede reducirse.

Además de los enemigos enviados por el jugador, automáticamente se genera en la pantalla de cada jugador un enemigo aleatorio cada 30 segundos.

A continuación se describen todos los tipos de unidades disponibles para ser enviadas por un jugador y sus características principales.

Unidad básica



Es la unidad genérica. Con atributos intermedios de velocidad de movimiento y resistencia, y vulnerable especialmente a torres de flechas. Las torres anti aéreas no pueden dispararlas.

Unidad armada



Es la unidad especial con armadura, que muestra una resistencia adicional a las torres de flechas, pero es vulnerable a las de bombas. Al igual que a la unidad básica, las torres anti aéreas no pueden disparar a esta unidad.

Tortuga voladora



Es la unidad básica aérea, un poco más veloz que las terrestres. Es inalcanzable para las torres de bombas pero especialmente vulnerable a las anti aéreas, que son torres específicas contra este tipo de unidad.

Paracaidista



Es similar en características a la unidad básica, con la peculiaridad de que es lanzado al escenario en paracaídas. En lugar de comenzar el recorrido en el punto de salida, lo hace en mitad del camino.

Cangrejo



Unidad de alta resistencia. Su velocidad es algo más lenta que la de las unidades normales y es especialmente vulnerable a las torres de flechas, pero es inmune a las avispas y su aguante es mucho mayor que el de las unidades básicas.

Jinetes



Los jinetes montan una tortuga terrestre gigante. Es la unidad más poderosa y cara. Es muy lenta y recibe daño extra de bombas y avispas, pero su resistencia es muy elevada y es difícil de destruir.

Ninja



Unidad de ataque sorpresa. Su velocidad de movimiento es muy alta y es muy resistente a las torres normales. Su punto débil son las torres de avispas, que les hacen daño extra.

Murciélago



Unidad especial de ataque. En lugar de recorrer el camino hacia la meta, esta unidad sobrevuela el área enemiga hasta ser destruida. Mientras está viva, otorga a todas sus unidades aliadas una bonificación de velocidad de movimiento del 10%. Además, mientras haya al menos un murciélago sobrevolando uno de los escenarios, el jugador afectado no puede enviar unidades al rival.

6.3.4 Unidades Extra

Además de las unidades de ataque que los jugadores pueden enviar a su rival, hay dos unidades adicionales que aparecen en eventos específicos del juego.



Unidad estrella

Este tipo de unidad aparece cuando un jugador pulsa sobre una estrella amarilla caída del mapa. Es un tipo de unidad estándar terrestre con algo más de resistencia que la unidad genérica, pero sin restricción de daño a ningún tipo de torre.



Unidad corazón

Estas unidades tienen las mismas características que las unidades estrella. Hay un 5% de probabilidad de que cada barcaza de enemigos automáticos generada envíe una unidad corazón en lugar de la unidad que hubiera correspondido. Sólo en ese caso aparecen y su eliminación otorga al jugador un punto de vida adicional, siempre y cuando no se encuentre en el tope de 10.

6.3.5 Torres

Las torres son el elemento defensivo principal. Se construyen en los troncos protegiendo una zona específica del mapa alrededor de ellas. Cada una está especializada en destruir ciertos tipos de enemigos y todas ellas disponen de 5 niveles de mejora que aumentan progresivamente sus capacidades de daño, rango de ataque y cadencia de disparo.

Torre de flechas



Torre principal. Ataca a todos los tipos de enemigos haciendo un daño estándar en la mayoría de ellos. Especialmente efectiva contra las unidades básicas y los cangrejos.

Torre de bombas



Torre específica de daño a unidades terrestres. Lanza bombas que al explotar, producen daño al enemigo afectado y a los que se encuentren en un área alrededor de él.

Torre anti aérea



Sólo ataca a unidades aéreas, pero tiene un daño adicional muy elevado contra ellas. Es difícil para el jugador resistir ataques de unidades voladoras sin disponer de estas torres.

Torre de avispas



El ataque de esta torre no es muy efectivo en cuanto a daño se refiere excepto para los ninjas. Su principal característica es que el enemigo afectado por la picadura de una de sus avispas, ve ralentizada su velocidad de movimiento en un intervalo de tiempo variable según el nivel de la torre.

6.3.6 Atributos de torres

Todas las torres, además de las características específicas que las diferencian, poseen tres atributos comunes principales.

Daño

El daño define la cantidad de puntos de vida que reduce a una unidad cuando la golpea. El daño tiene un modificador asociado al tipo de enemigo al que dispara, de forma que aunque el daño de una torre en un nivel determinado siempre es el mismo, sólo un porcentaje de éste afectará a la unidad impactada, en función del tipo de unidad del que se trate.

Rango

Es la distancia de acción de la que dispone una torre, un área circular alrededor de la ella dentro de la cual puede localizar enemigos y dispararlos. Para visualizar ésta zona en una torre específica, el jugador debe pulsar en la torre.

Cadencia

Intervalo de tiempo entre cada disparo de una torre y el siguiente. En la mayoría de las torres es el atributo que más evoluciona con las mejoras. Cuanto menor sea este intervalo de tiempo, más rápida será la torre disparando.

6.3.7 Habilidades especiales

Además de las torres, el jugador dispone de cuatro elementos especiales de defensa. Estos elementos se compran por separado y ofrecen nuevas posibles estrategias de defensa.

Gatling



La gatling es una metralleta de alta velocidad de ataque que dispara multitud de flechas. Al comprarse, se coloca automáticamente en un punto predefinido del mapa y dispara a los enemigos con un comportamiento similar al de las torres. Su posición es perfecta para que su rango abarque un área muy grande del camino.

Bomba



La bomba es de lanzamiento manual. Tiene un efecto explosivo de área en unidades terrestres en un área amplia desde la zona de impacto. Una vez ha explotado, se regenera en 20 segundos y puede volverse a lanzar.

Helicóptero



Unidad especial de ataque aéreo. Puede moverse manualmente por el mapa y dispara a las unidades aéreas que entren en su zona de visión.

Avispa



Con un comportamiento similar a las torres de avispas, la avispa reina lanza pequeñas avispas a los enemigos en su rango con un poder de ralentización muy alto. La avispa se también se puede mover manualmente por todo el escenario.

6.4 Menú de opciones y mute

Como se explicó en apartados anteriores, ambos jugadores tienen acceso a tres botones de configuración situados en el centro de la pantalla, en el lado izquierdo del jugador principal. Dos de ellos sirven para modificar opciones del sistema y el tercero da acceso a la guía de juego.

6.4.1 Menú de opciones

El menú de opciones se abre en forma de pop up y muestra las opciones principales de configuración del sistema.



Sonido

En el apartado de sonido, se permite a los jugadores habilitar o deshabilitar los diferentes tipos de sonidos del juego. La música, el sonido ambiente y los efectos.

Gráficos

En el apartado gráfico se permite la opción de habilitar o deshabilitar la proyección de sombras de las unidades y sus animaciones. Esto permite a los usuarios con dispositivos antiguos o de escasa potencia, hacer que el juego se ejecute con menor consumo de recursos.

6.4.2 Mute

El botón mute permite detener la reproducción de cualquier sonido en el juego sin necesidad de entrar en el menú de opciones. Pulsando en él una vez el sonido se detendrá, y volviendo a pulsar en él, se reanudará la reproducción.

6.4.3 Guía de juego

La guía de juego es una referencia de acceso rápido a las principales características del juego. Se muestra y define en el próximo apartado “Guía en el dispositivo”.

6.5 Guía en el dispositivo

La guía de juego ofrecida en el propio dispositivo se muestra en un pop up que ocupa horizontalmente casi la mayor parte de la pantalla. Su objetivo es proporcionar información básica sobre el manejo del juego, y específica sobre unidades de ataque y defensa. Está disponible en cualquier momento durante la batalla y ofrece su información dividida en pestañas.

Overview

Información cómoda y sencilla de la relación entre los objetos defensivos y las unidades de ataque. Muestra la efectividad de cada elemento frente a enemigos específicos.



Figura 91. Menú Overview

Troops

Listado de todas las unidades disponibles con una descripción de sus características principales. Cada una se muestra con su gráfico asociado.



Figura 92. Menú Troops

Defenses

Muestra la lista de torres defensivas disponibles, así como las habilidades especiales, con una descripción breve de cada una de ellas.



Figura 93. Menú Defenses

Graphics

Esta pestaña es para jugadores más avanzados. Muestra gráficos con el daño específico que cada torre y cada elemento especial hacen a cada unidad.



Figura 94. Menú Graphics

7 Conclusiones

7.1 Conclusiones del Proyecto

Teniendo en cuenta que este proyecto se centraba en el desarrollo de un producto orientado a su distribución en el mercado para un cliente que exigía determinados requisitos, se ha de analizar si los objetivos en los que se asentaron las bases y necesidades del proyecto se han cumplido y en qué medida.

7.1.1 Análisis de objetivos

Como se ha transmitido durante todo el documento, el objetivo principal para el cliente era crear un juego estilo Tower Defense con un diseño simple pero con una calidad equiparable a la de los principales títulos del mercado, añadiendo la novedad de ofrecer un sistema de dos jugadores simultáneos que aún no se ha visto en el mercado.

Con la finalización del desarrollo del producto, se concluye que el objetivo principal se ha conseguido en el plazo deseado y empleando los recursos que se estimaron necesarios en la planificación.

Dispositivos

El videojuego es perfectamente compatible con sistemas Android e iOS y ofrece un rendimiento aceptable para poder ser ejecutado en dispositivos de capacidades inferiores a las de los requeridos.

Migración

Se ha conseguido con éxito adaptar un sistema manejado con ratón a uno nuevo manejable en entornos multitáctiles por dos usuarios al mismo tiempo.

Entorno

Se ha empleado con eficiencia el entorno establecido como óptimo para la aplicación, Adobe Flash Professional CS6, empleando la última versión de AIR.

Compatibilidad

La misma versión del juego es compatible para sistemas tanto Android como iOS, tan solo cambiando los parámetros necesarios en la configuración de la publicación. La aplicación se adapta a los distintos tamaños de pantalla de los dispositivos en los que se jugará.

Jugabilidad

Se ha conseguido ofrecer con éxito el sistema de dos jugadores simultáneos en el mismo dispositivo en modo batalla, de forma divertida y original.

Idioma

Todos los textos de la aplicación se muestran en inglés.

7.1.2 Adaptaciones de juegos para dispositivos móviles

Aunque existen herramientas cada vez más avanzadas para facilitar la correcta conversión de un juego o herramienta ideada para sistemas con interfaces de usuario basadas en teclado y ratón, a nuevas aplicaciones compatibles en dispositivos móviles con pantallas táctiles, el trabajo que supone realizar la migración de manera que el resultado final sea de calidad es enorme.

Existen infinidad de problemas asociados a la diversidad de formatos de pantalla en toda la gama de dispositivos móviles ofertada en el mercado, además de los relacionados a la gran diferencia de potencia y fiabilidad que hay entre los diferentes móviles y tabletas disponibles.

Ofertar un producto que funcione correctamente en todos ellos de forma que ofrezca un buen rendimiento en los dispositivos de menor potencia, y a la vez explote las características y capacidades de los más modernos es un trabajo que conlleva esfuerzo y creatividad desde el momento mismo en el que se plantea el diseño.

Se ha de mantener en todo momento una mentalidad abierta a explorar cualquier nueva forma de mejorar la comunicación entre usuario y sistema,

aprovechando las pantallas multitáctiles y teniendo en cuenta que es una característica moderna aún por explotar completamente.

Ha supuesto un reto y una fuente de aprendizaje muy amplia el enfrentarse a todas las complicaciones que surgen en este proceso.

7.1.3 Mercado

Aunque sólo era necesario desarrollar el producto para la fase previa a su difusión en el mercado, se ha estudiado si su lanzamiento podía ser viable. La aplicación cumple todos los requisitos necesarios para ser publicada tanto en Google Play como en el Apple Store y además, ofrece un nivel de calidad muy elevado para competir con cualquier producto de su mismo género.

A falta de una inversión en marketing, el producto está preparado para una publicación libre de riesgos.

7.2 Conclusiones personales

Tras la conclusión del desarrollo de la aplicación y de esta memoria, es conveniente analizar desde una perspectiva más amplia lo que ha supuesto a nivel práctico y personal llevarla a cabo.

El proyecto de fin de grado tiene como fin hacer que el alumno ponga en práctica sus conocimientos para recrear una situación más parecida a lo que encontrará en su vida profesional. En el caso de este proyecto, se ha intentado simular con la mayor fidelidad posible una situación real de trabajo para un cliente con cierto nivel de exigencia.

Desarrollar un videojuego a este nivel supone un reto personal muy grande, ya que exige trabajo, inteligencia y creatividad, no ya sólo en la parte relativa al diseño funcional de la aplicación, sino en la propia codificación del juego, en la que el programador se enfrenta a multitud de problemas para los que la teoría no le ha preparado completamente.

En este sentido, se puede concluir que personalmente este proyecto ha satisfecho su función en todos los sentidos, afianzando en el autor los ya bien

adquiridos conocimientos en sus años de carrera y otorgándole seguridad y confianza para su carrera profesional.

7.3 Líneas futuras

Gracias a la sencillez del concepto, el juego puede evolucionar de distintas formas una vez haya sido publicado. A continuación se estudian diferentes posibilidades que se consideran viables siguiendo la dinámica de sencillez y originalidad en la que se ha trabajado durante todo el proyecto.

Nuevos elementos

Por la estructura del código, sería sencillo añadir nuevos elementos de ataque y defensa para los jugadores, tanto unidades, como torres y objetos especiales.

Simplificación de interfaz

Aunque se consiguió el objetivo secundario de integrar la interfaz en el escenario de batalla, ésta ocupa un porcentaje muy grande de la pantalla, limitando el tamaño del escenario. Hacer que la interfaz permanezca oculta mientras no se esté utilizando resolvería el problema y permitiría crear escenarios más grandes.

Nuevas pantallas

Manteniendo la mecánica y elementos establecidos para el juego, la creación de nuevas pantallas resultaría cómodo y rápido, y aportaría mucho valor al juego.

8 Referencias

[1] **Newzoo**, *Placing Mobile Games in Perspective of the Total Games Market: Free Mobile Trend Report*, <http://www.newzoo.com/insights/placing-mobile-games-in-perspective-of-the-total-games-market-free-mobile-trend-report/>

[2] **Deloitte University Press**, *Exploring pathways to growth in the mobile semiconductor industry*, <http://dupress.com/articles/rising-tide-exploring-pathways-to-growth-in-the-mobile-semiconductor-industry/>

[3] **FanStudio.co.uk**, *How Much Do Average Smartphone Apps Make?*, <http://www.fanstudio.co.uk/iphone-game-development/09/18/how-much-do-average-smartphone-apps-make/>

[4] **Dan Laughlin**, *The Gamification of Mobile Games*, <http://www.flurry.com/bid/92377/The-Gamification-of-Mobile-Games>

[5] **Business World TI**, *Requerimiento, Trazabilidad, requisito, Matriz de Trazabilidad de Requisitos, Elicitación, Patrones de Seguridad, Herramienta Case*, <http://businessworldti.wordpress.com/2012/11/19/requerimiento-trazabilidad-requisito/>

[6] **René Ávila**, *Objective-C, el lenguaje nativo de iOS*, <http://rubiko.mx/objective-c-el-lenguaje-nativo-de-ios/>

[7] **Apple Inc**, *Approval guidelines*, <https://developer.apple.com/appstore/resources/approval/guidelines.html>

[8] **Google Play**, *Políticas del Programa para Desarrolladores de Google Play*, <https://play.google.com/about/developer-content-policy.html>

[9] **Y. Huang, P. Jalote and C. Kintala**, *Two Techniques for Transient Software Error Recovery. Lecture Notes in Computer Science*, Vol. 774, pp. 159-170, 1994.

[10] Trygve Reenskaug and James O. Coplien, *The DCI Architecture: A New Vision of Object-Oriented Programming*,
http://www.artima.com/articles/dci_vision.html

[11] Tomás Bradanovich, *MVC, Diseño en cascada y Prototipos*,
<http://sigesun.blogspot.com.es/2014/03/mvc-diseno-en-cascada-y-prototipos.html>

[12] Adobe, *ActionScript Technology Center*,
<http://www.adobe.com/devnet/actionscript.html>

[13] Guillermo Díaz, *UML - Casos de Uso y Diagramas de Clase*,
<http://www.slideshare.net/guillermmods/uml-casos-de-uso-y-diagramas-de-clase>

[14] Adobe, *Adobe Air*, <http://www.adobe.com/products/air.html>

[15] Jeff Ward, *What is a Game Engine?*,
http://www.gamecareerguide.com/features/529/what_is_a_game_.php

[a.1] **Adaptación a móvil de Earthworm Jim**
http://www.pcworld.idg.com.au/slideshow/334182/23_console_games_iphone/?image=4

[a.2] **Muestra de dispositivo de pantalla multitáctil gigante**
http://101.wacom.com/sp/bamboo/multi_touch.php

[a.3] **Diferencias entre formato de escritorio y táctil de Kingdom Rush**
<http://www.kingdomrush.com/>

[a.4] **Modelos en baja poligonización**
<http://www.deviantart.com/morelikethis/228763540>

[a.5] Sprite Sheet de New Super Mario Bros Wii

<http://sandbox.yoyogames.com/games/79272-platfor-beta>

[a.6] Diferentes tipos de eventos táctiles

<http://adventurgraphics.blogspot.com.es/2012/08/15-tendencias-dentro-del-diseno-web-de.html>

[a.7] Emulación de mando de consola en un dispositivo móvil

<http://www.engadget.com/2011/12/07/onlive-intros-apps-for-ios-and-android-refreshes-its-wireless-c/>

[a.8] Aproximación de cámara en versión móvil de Fieldrunners

<http://www.androidtapp.com/fieldrunners-hd/>

[a.9] Ejemplo de juego estilo Tower Defense (Tower Defense: Lost Earth)

<http://www.tecnoinforme.com/2014/05/26/los-60-mejores-juegos-android/>

[a.10] Captura de pantalla del editor de niveles de Warcraft III <http://game-spot.iimdo.com/cl%C3%A1sicos/warcraft-3/>

[a.11] Captura de pantalla de Kingdom Rush.

<http://armorgames.com/play/12141/kingdom-rush>

[a.12] Captura de pantalla de Plants vs Zombies

<http://support.popcap.com/agegate.php?site=PopCap.com&dst=http%3A%2F%2Fsupport.popcap.com%2Fmobile%2Fandroid>

[a.13] Captura de pantalla de Jelly Defense

<https://play.google.com/store/apps/details?id=pl.idreams.jellydefense&hl=es>

[a.14] Captura de pantalla de Fieldrunners

<https://chrome.google.com/webstore/detail/fieldrunners/lkpikhjbfbfdblahfidklcohlaeabak>

[a.15] Captura de pantalla de Bloons Tower Defense 5

<http://ninjakiwi.com/Espanol/Games/Tower-Defense/Play/Bloons-Tower-Defense-5.html>

[a.16] Captura de pantalla de Pirate Legends HD

<https://itunes.apple.com/es/app/pirate-legends-td/id626884809?mt=8>

[a.17] Xcode para Mac OS X

<https://itunes.apple.com/es/app/xcode/id497799835?mt=12>

[a.18] Flash CS6 <http://helpx.adobe.com/es/x-productkb/policy-pricing/cs6-product-downloads.html>

[a.19] Unity3d <http://unity3d.com/es>

[a.20] Pre-visualización de filtro de sombreado en Flash Professional

<http://www.recursoesweb.es/16-simples-tutoriales-para-adobe-flash/>

[a.21] Diferencia de pérdida de calidad en el escalado en bitmaps con respecto al escalado en imágenes en formato vector <http://manonmw.wordpress.com/>

[a.22] Evento estándar "Zoom" que requiere capacidad multitáctil del dispositivo

<http://www.iosbit.com/anadir-gestos-multitactiles-en-el-ipad-2/>