

TÍTULO DEL TRABAJO FIN DE GRADO

THIBAUT LOPEZ

Trabajo fin de Grado

Supervisado por Dr. Pablo Trinidad Martín-Arroyo



Universidad de Sevilla

marzo 2021

Publicado en marzo 2021 por

Thibaut Lopez

Copyright © MMXXI

<http://www.lsi.us.es/~trinidad>
ptrinidad@us.es

Pon aquí cuestiones acerca del copyright

Yo, D. Thibaut Lopez con NIF número X9026046H,

DECLARO

mi autoría del trabajo que se presenta en la memoria de este trabajo fin de grado que tiene por título:

Título del Trabajo Fin de grado

Lo cual firmo,

Fdo. D. Thibaut Lopez
en la Universidad de Sevilla
28/03/2021

Tu dedicatoria aquí



AGRADECIMIENTOS

Agradecimientos a mi tutor por soportarme.



RESUMEN

En este proyecto se busca desarrollar el videojuego Way Too Late. Para ello se usará la herramienta Unity.

ÍNDICE GENERAL

I	Introducción	1
1.	Contexto	3
1.1.	El mundo del videojuego	4
1.2.	Subcontexto	4
1.3.	Subsubcontexto	4
1.4.	Estado del arte	4
2.	Objetivos	5
2.1.	Motivación	6
2.2.	Listado de objetivos	6
II	Organización del proyecto	7
3.	Metodología	9
3.1.	Estructura organizacional del proyecto	10
3.2.	Metodología de desarrollo	10
3.2.1.	Funcionamiento de Scrum	10
3.2.2.	Aplicación de Scrum	11
4.	Planificación	13

4.1. Gestión de alcance	14
4.1.1. Definición del alcance	14
4.1.2. Requisitos	16
4.1.3. Matriz de trazabilidad de requisitos	21
4.1.4. EDT	24
4.1.5. Diccionario de la EDT	24
4.2. Gestión de configuración	33
4.3. Gestión del cronograma	34
4.3.1. Lista y estimación de actividades	34
4.3.2. Línea base del cronograma	46
4.4. Gestión de adquisiciones	47
4.4.1. Justificación de las adquisiciones	47
4.4.2. Listado de adquisiciones	48
4.5. Gestión de costes	50
4.5.1. Estimación de costes	50
4.5.2. Presupuesto	51
4.6. Gestión de la calidad	51
4.7. Gestión de riesgos	53
4.7.1. Identificación y revaluación de riesgos	53
4.7.2. Análisis cualitativo de los riesgos	56
4.7.3. Análisis cuantitativo de los riesgos	57
4.7.4. Plan de respuesta a los riesgos	58

III	Desarrollo del proyecto	61
5.	Product Backlog	63
6.	Iteración 1	69
6.1.	Sprint Backlog	70
6.2.	Control de la gestión	72
6.2.1.	Control del alcance	72
6.2.2.	Control del cronograma	74
6.2.3.	Control de los costes	75
6.2.4.	Control de la calidad	76
6.2.5.	Control de riesgos	77
6.3.	Informes de cambio	80
6.3.1.	Documentación	80
6.3.2.	GDD	81
IV	Cierre del proyecto	83
7.	Manual de usuario	85
7.1.	Sección libre	86
8.	Conclusiones	87
8.1.	Informe post-mortem	88
8.1.1.	Lo que ha ido bien	88
8.1.2.	Lo que ha ido mal	88
8.1.3.	Discusión	88
8.2.	Trabajos futuros	88

V	Appendices	89
A.	Software Product Lines	91
A.1.	Software Product Lines	92
A.2.	Feature Models	92
A.3.	Automated Analysis of Feature Models	94
A.3.1.	Scope	94
A.4.	Dynamic Software Product Lines (DSPL)	97
A.5.	Hypothesis and Objectives	97
B.	Siglas	101
	Referencias bibliográficas	102

ÍNDICE DE FIGURAS

4.1. EDT	24
4.2. Cronograma	46
6.1. Burndown Iteración 1	73
A.1. An example of a Home Integration System	93
A.2. A different view on AAFM distinguishing between information extrac- tion and explanatory operations	95

ÍNDICE DE CUADROS

4.1. Requisitos de negocio	16
4.2. Requisitos de información	17
4.3. Requisitos funcionales	19
4.4. Requisitos de calidad	19
4.5. Requisitos de implementación	19
4.6. Requisitos no funcionales	20
4.7. Matrices de trazabilidad de requisitos	23
4.8. Paquete de trabajo 1.1.1.	25
4.9. Paquete de trabajo 1.1.2.	25
4.10. Paquete de trabajo 1.1.4.	25
4.11. Paquete de trabajo 1.1.5.	26
4.12. Paquete de trabajo 1.1.6.	26
4.13. Paquete de trabajo 1.1.7.	26
4.14. Paquete de trabajo 1.1.8.	27
4.15. Paquete de trabajo 1.2.1.	27
4.16. Paquete de trabajo 1.2.2.	27
4.17. Paquete de trabajo 1.2.3.	28
4.18. Paquete de trabajo 1.2.4.	28
4.19. Paquete de trabajo 1.3.1.	28
4.20. Paquete de trabajo 1.3.2.	29

4.21. Paquete de trabajo 1.3.3.	29
4.22. Paquete de trabajo 1.3.4.	29
4.23. Paquete de trabajo 1.3.5.	30
4.24. Paquete de trabajo 2.1.	30
4.25. Paquete de trabajo 2.2.	30
4.26. Paquete de trabajo 2.3.	31
4.27. Paquete de trabajo 2.4.	31
4.28. Paquete de trabajo 3.1.	31
4.29. Paquete de trabajo 3.2.	32
4.30. Actividades paquete 1.1.1.	34
4.31. Actividades paquete 1.1.2.	35
4.32. Actividades paquete 1.1.3.	35
4.33. Actividades paquete 1.1.4.	36
4.34. Actividades paquete 1.1.5.	36
4.35. Actividades paquete 1.1.6.	37
4.36. Actividades paquete 1.1.7.	37
4.37. Actividades paquete 1.2.1.	38
4.38. Actividades paquete 1.2.2.	38
4.39. Actividades paquete 1.2.1.	39
4.40. Actividades paquete 1.2.4.	39
4.41. Actividades paquete 1.3.1.	40
4.42. Actividades paquete 1.3.2.	40
4.43. Actividades paquete 1.3.3.	41
4.44. Actividades paquete 1.3.4.	41
4.45. Actividades paquete 1.3.5.	42

4.46. Actividades paquete 2.1.	42
4.47. Actividades paquete 2.2.	43
4.48. Actividades paquete 2.3.	43
4.49. Actividades paquete 2.4.	44
4.50. Actividades paquete 3.1.	44
4.51. Actividades paquete 3.2.	45
4.52. Listado de adquisiciones	49
4.53. Estimación de costes	50
4.54. Presupuesto	51
4.55. Análisis de la calidad	52
4.56. Lista de riesgos negativos	54
4.57. Lista de riesgos positivos	55
4.58. Relación impacto-probabilidad de los riesgos	56
4.59. Analisis cuantitativo de los riesgos	57
4.60. Plan de respuesta para los riesgos negativos	59
4.61. Plan de respuesta para los riesgos positivos	60
5.1. Product Backlog	68
6.1. Sprint Backlog Iteración 1	71
6.2. Control del alcance Iteración 1	72
6.3. Control del cronograma Iteración 1	74
6.4. Control de costes Iteración 1	75
6.5. Control de la calidad Iteración 1	76
6.6. Control de riesgos negativos Iteración 1	78
6.7. Control de riesgos positivos Iteración 1	79

A.1. Most frequently used explanatory operations and their corresponding information extraction operations	99
---	----

LISTA DE TAREAS PENDIENTES

■ To Abductive Section in 2.1	92
Figura: A feature model example	93
■ To Abductive Intro	94

PARTE I

INTRODUCCIÓN

CONTEXTO

1

2

A

continuación se introduce el entorno en el que se va a llevar a cabo el proyecto.

1.1 EL MUNDO DEL VIDEOJUEGO 1

Hay que ir poco a poco acotando el contexto donde se desarrolla el proyecto. No 2
se debe sobreentender que el evaluador de la memoria sabe del tema. Escribid el texto 3
para la abuela. 4

1.2 SUBCONTEXTO 5

1.3 SUBSUBCONTEXTO 6

1.4 ESTADO DEL ARTE 7

Cómo se encuentra la industria hoy en día a nivel económico y tecnológico. 8

OBJETIVOS

1

2

A

quí mal un breve resumen del capítulo.

2.1	MOTIVACIÓN	1
	Esta sección se rellenará cuando tengamos un producto de mercado en lugar de un	2
	proyecto en el que haya un cliente específico. Deberá justificar brevemente el problema	3
	a resolver, escenario en el que se aplica, hipótesis de partida, público objetivo, etc.	4
2.2	LISTADO DE OBJETIVOS	5
	Objetivo 1. Blabla Detalles del objetivo 1.	6
	Objetivo 2. Blabla Detalles del objetivo 2.	7

PARTE II

ORGANIZACIÓN DEL PROYECTO

METODOLOGÍA

1

2

A

continuación se explicará la metodología usada en el desarrollo del proyecto.

3.1 ESTRUCTURA ORGANIZACIONAL DEL PROYECTO

El único desarrollador de este proyecto es Thibaut Lopez, que se ocupará de todos y cada uno de los aspectos del TFG. Por tanto, es considerado tanto el Project Manager como el programador, realizando toda la documentación y el código que puedan ser necesarios.

Para ello, dispondrá de un horario de trabajo adaptable a su agenda universitaria. Es decir, podrá trabajar los días y la cantidad de horas por día que desea siempre y cuando cumpla con los plazos establecidos. Esto también significa que no se tomarán en cuenta ni fines de semana ni festivos a la hora de estimar las fechas de entrega del trabajo realizado.

Para poder llevar a cabo un seguimiento del tiempo dedicado al proyecto, se usará la herramienta Clockify para poder saber las fechas en las que trabajará y las horas que lo hará.

3.2 METODOLOGÍA DE DESARROLLO

La metodología que se va a usar para la creación del videojuego es Scrum, un proceso de desarrollo de productos software basado en la realización de entregas constantes de incrementos funcionales del producto final.

3.2.1 Funcionamiento de Scrum

El primer paso para aplicar la metodología Scrum es realizar un Product Backlog, que es un documento que recopila todos los requisitos del producto final desde el punto de vista del usuario en forma de historias de usuario. De esta forma se obtiene una visión entera del trabajo a realizar y permite repartir tareas en sprints.

Tras eso se realiza una selección de algunas historias de usuario en previsión a su realización durante un sprint. Esta selección es un documento llamado Sprint Backlog y esta más enfocado a permitir a los desarrolladores saber el trabajo que deben realizar durante un sprint en concreto. Se debe realizar uno de estos documentos al principio de cada sprint para planificarlos.

Los sprints son el tiempo que poseen los desarrolladores para satisfacer las historias de usuario del Sprint Backlog. Suelen ser de entre 2 a 4 semanas y durante su ejecución siempre se deben realizar reuniones diarias para mantener un seguimiento

1 del trabajo realizado por cada desarrollador.

2 Una vez un sprint terminado se obtiene un entregable que satisface el Sprint Bac-
3 klog y se realiza una revisión del sprint para repasar el trabajo realizado, y una retros-
4 pectiva de sprint para analizar posibles cambios en la forma de trabajar de los desarro-
5 lladores. El objetivo de esto último es poder mejorar la ejecución del siguiente sprint y
6 aprender de errores pasados.

7 Con todo, se realizan varios sprint hasta completar el Product Backlog y así obtener
8 el producto final.

9 3.2.2 Aplicación de Scrum

10 En el caso de este proyecto, se llevarán a cabo 4 sprints (o iteraciones de aquí en
11 adelante) de 3 semanas cada una. El trabajo a realizar en cada una de ellas se agrupará
12 por funcionalidades y no por relevancia en el proyecto como se haría normalmente.
13 Esto es para facilitar la realización de tareas interrelacionadas o similares y poder tener
14 fácilmente conciencia de que áreas del videojuego pueden sufrir lacunas o retrasos.

15 A pesar de crear un Sprint Backlog con todas las tareas que deberían de realizarse
16 en un sprint, el objetivo no será de realizarlas todas, sino de desarrollar las funciona-
17 lidades más relevantes. Dependiendo de la gestión del tiempo del desarrollador y del
18 tiempo disponible de sprint para implementar funcionalidades menos importantes, se
19 realizarán el mayor número posible de estas tareas sin poner en peligro la organiza-
20 ción temporal del proyecto. Es decir, si para un sprint se deben realizar una tarea muy
21 importante y cinco mucho menos relevantes, se podrán dar el caso de llevar a cabo
22 las seis tareas, la importante y dos otras, o solo la importante, sin poner en peligro el
23 producto final en ningún caso.

24 Además, puesto que el equipo de trabajo de este proyecto se compone de un único
25 desarrollador, no se llevarán a cabo las reuniones diarias típicas de Scrum. A parte de
26 esto, se respetarán las otras prácticas intrínsecas de Scrum, así como la generación de
27 un Product Backlog y un Sprint Backlog por iteración.

PLANIFICACIÓN

1

2 *E* n este capítulo se procederá a planificar el proyecto, definiendo el trabajo a realizar, la
3 forma de realizarlo y las repercusiones que tiene en el proyecto en su conjunto.

4.1 GESTIÓN DE ALCANCE

1

4.1.1 Definición del alcance

2

En este apartado se busca definir en lo que consiste el proyecto explicando lo que va a realizarse durante su desarrollo.

3

4

Enunciado del alcance

5

El proyecto consiste en la creación del videojuego Way Too Late cuyas características se encuentran descritas en su GDD (Game Design Document). Las necesidades del proyecto serán por tanto deducidas de este documento de aquí en adelante.

6

7

8

Criterios de aceptación

9

Para que el proyecto se considere cerrado, debe desarrollarse una versión ejecutable del juego que debe funcionar correctamente según los requisitos establecidos en el GDD y toda la documentación generada por el proyecto debe ser válida y útil.

10

11

12

Tanto el código del juego como la documentación deben ser entregados antes de una fecha establecida y mediante un procedimiento concreto. La fecha y el procedimiento serán descritos en otro apartado de este mismo documento (Gestión de entregables) así como las repercusiones negativas que impliquen su incumplimiento.

13

14

15

16

Entregables

17

El entregable más importante del proyecto es el código completo del juego y los elementos que lo forman (ilustraciones, animaciones, músicas ...). Todo esto debe ser entregado de forma que se pueda usar intuitivamente con Unity. Junto al código debe hallarse un ejecutable del juego que permita a cualquier jugador poder jugar a Way Too Late de forma simple y sencilla.

18

19

20

21

22

Además, deben hallarse los documentos generados por el desarrollo del proyecto. Estos son:

23

24

- Plan de gestión: Establece el funcionamiento del proyecto y dicta como va a desarrollarse.
- Procesos de ejecución, seguimiento y control: Permite llevar a cabo un seguimiento del desarrollo del proyecto, así como el control de la aplicación de lo establecido en el Plan de gestión.

25

26

27

28

29

■ Cierre del proyecto: Revisa lo realizado durante el proyecto y como se ha realizado en comparación con lo planificado en el Plan de gestión, analiza el estado final del proyecto y permite su cierre.

■ Presentación: Permite al desarrollador exponer oralmente el trabajo realizado.

Todos estos documentos deberán encontrarse juntos en un único documento, este mismo documento, que debe ser entregado en formato pdf.

Exclusiones

Una vez el proyecto cerrado, posteriores versiones de este mismo juego realizado por el mismo desarrollador no serán tomadas en cuenta como partes de este proyecto en ninguna medida. Estas versiones podrán relacionarse con otros proyectos o ser simplemente modificaciones y arreglos de errores en el código, pero no tendrán nada que ver con el proyecto al que este documento pertenece.

Restricciones

Todos los entregables deben ser entregados obligatoriamente antes del 23 de junio de 2021, a excepción de la presentación que puede ser entregada después, siempre y cuando sea antes del 7 de julio de 2021. En ambos casos, no se aceptará ningún tipo de retraso en la entrega.

El equipo de trabajo de este proyecto no podrá ser modificado bajo ningún caso una vez el proyecto iniciado. Cuando esto ocurra, deberá respectarse todo lo definido en este documento en la medida de lo posible.

Supuestos

Si no se pudiesen cumplir los plazos previamente establecidos (apartado 1.1.5), se podrá realizar una segunda entrega antes del 3 de septiembre de 2021, pudiendo entregarse la presentación no más tarde del 8 de septiembre de 2021. Una vez estas fechas pasadas, no se podrán realizar ninguna entrega más.

Fases del proyecto

El proyecto se dividirá en 4 fases: planificación, ejecución, seguimiento y control, cierre y presentación. La fase de ejecución, seguimiento y control se desarrollará en 4 iteraciones de alrededor de 3 semanas cada una. En ellas se aplicará la metodología Scrum para el desarrollo del juego. La fase de cierre solo constará de una semana. La fase de presentación constará de 4 días para realizar la presentación, y de una semana

en la que podrá realizarse la exposición.

4.1.2 Requisitos

En este apartado se enunciarán los requisitos indispensables para la correcta realización del proyecto. A cada requisito se le asignará un Id que permita su sencilla identificación más adelante. Además, se establecerá una jerarquía de prioridades en las que sus significados son:

- Baja: si el requisito no puede ser satisfecho, no afectará a la posibilidad de completar el proyecto.
- Media: este tipo de requisito es importante para el proyecto, pero puede no ser satisfecho sin afectar al proyecto.
- Alta: este tipo de requisito es el más importante y es necesario para que el proyecto pueda ser completado.

Requisitos de negocio			
Id	Descripción	Prioridad	Criterios de aceptación
RN01	El juego debe poseer un menú principal	Media	El menú principal debe ser accesible
RN02	El juego debe permitir el desarrollo de una partida	Alta	Debe poder jugarse una partida correctamente
RN03	El juego debe permitir ver las puntuaciones guardadas	Baja	Debe existir una pantalla para ver las puntuaciones y debe ser accesible
RN04	El juego debe permitir cambiar los controles	Baja	Debe existir una pantalla para poder cambiar los controles y debe ser accesible
RN05	El juego debe permitir que se deje de ejecutar	Alta	Debe poderse salir del juego

Cuadro 4.1: Requisitos de negocio

Requisitos de información			
Id	Descripción	Prioridad	Criterios de aceptación
RI01	El juego debe guardar el nombre del jugador	Baja	El jugador poder registrar su nombre y que aparezca en las puntuaciones
RI02	El juego debe guardar la puntuación del jugador	Baja	La puntuación del jugador debe poder verse desde la pantalla de puntuaciones
RI03	El juego debe guardar la asignación de controles usada por el jugador	Baja	El jugador no necesita asignar los controles cada vez que inicia el juego porque están guardados

Cuadro 4.2: Requisitos de información

Requisitos funcionales			
Id	Descripción	Prioridad	Criterios de aceptación
RF01	Se pueden ver las puntuaciones guardadas	Baja	Las puntuaciones que aparecen son correctas
RF02	Se pueden cambiar la asignación de controles	Baja	Los controles se asignan correctamente
RF03	Se puede dejar de ejecutar el juego	Alta	El juego deja de ejecutarse del todo
RF04	Se puede iniciar una partida	Alta	Una partida comienza desde el principio sin problemas
RF05	Se puede poner la partida en pausa	Media	La partida puede detenerse, volver a continuar e ir al menú principal
RF06	Se puede realizar el movimiento básico	Alta	El jugador puede desplazarse libremente
RF07	Los elementos del nivel se desplazan	Alta	El nivel se desplaza y el jugador puede recorrerlo
RF08	La mecánica de la velocidad se encuentra implementada	Media	Esta mecánica funciona correctamente
RF09	La mecánica del portátil se encuentra implementada	Media	Esta mecánica funciona correctamente
RF10	La mecánica de los objetos se encuentra implementada	Media	Esta mecánica funciona correctamente
RF11	Los obstáculos y sus funciones se encuentran implementados	Alta	Los obstáculos aparecen y el jugador puede chocarse con ellos
RF12	Los atajos y sus funciones se encuentran implementados	Media	Los atajos aparecen y pueden ser usados correctamente
RF13	Las zonas del nivel y sus funciones se encuentran implementados	Media	El nivel se divide en las tres zonas y cada una tiene sus características
RF14	La partida se termina al llegar al final del nivel	Alta	La partida termina correctamente

Requisitos funcionales			
Id	Descripción	Prioridad	Criterios de aceptación
RF15	Se calcula la puntuación del jugador al final de la partida	Alta	No hay errores en el cálculo de la puntuación
RF16	El jugador puede registrar su nombre al final de la partida	Baja	El jugador puede escribir su nombre y aparecerá al ver las puntuaciones

Cuadro 4.3: Requisitos funcionales

Requisitos de calidad			
Id	Descripción	Prioridad	Criterios de aceptación
RC01	El juego no posee ningún fallo	Alta	No se encuentran errores durante la ejecución del juego
RC02	Existe un tutorial que explica el funcionamiento del juego al jugador	Medio	El jugador comprende perfectamente como jugar una partida
RC03	Los tiempos de carga son cortos	Medio	Los tiempos de carga son cortos
RC04	Se respeta el aspecto estético descrito en el GDD	Medio	El aspecto del juego es cómico y simple

Cuadro 4.4: Requisitos de calidad

Requisitos de implementación			
Id	Descripción	Prioridad	Criterios de aceptación
RIM01	Usar Unity para el desarrollo	Alta	El código se estructura como un proyecto de Unity

Cuadro 4.5: Requisitos de implementación

Requisitos no funcionales			
Id	Descripción	Prioridad	Criterios de aceptación
RNF01	El juego puede ejecutarse sin hacer uso de Unity	Alta	Existe un archivo .exe que permite ejecutar el juego

Cuadro 4.6: Requisitos no funcionales

1 4.1.3 Matriz de trazabilidad de requisitos

Matriz de trazabilidad de requisitos					
Id	RN01	RN02	RN03	RN04	RN05
RN01	X	X	X	X	X
RN02		X			
RN03			X		
RN04				X	
RN05					X

Matriz de trazabilidad de requisitos					
Id	RN01	RN02	RN03	RN04	RN05
RI01			X		
RI02			X		
RI03				X	

Matriz de trazabilidad de requisitos					
Id	RN01	RN02	RN03	RN04	RN05
RF01	X		X		
RF02	X			X	
RF03	X				X
RF04	X	X			
RF05	X	X			
RF06		X			
RF07		X			
RF08		X			
RF09		X			
RF10		X			
RF11		X			
RF12		X			
RF13		X			
RF14		X			
RF15			X		
RF16			X		

Matriz de trazabilidad de requisitos			
Id	RI01	RI02	RI03
RF01	X	X	
RF02			X
RF03			
RF04			
RF05			
RF06			
RF07			
RF08			
RF09			
RF10			
RF11			
RF12			
RF13			
RF14			
RF15		X	
RF16	X		

Cuadro 4.7: Matrices de trazabilidad de requisitos

4.1.4 EDT

1

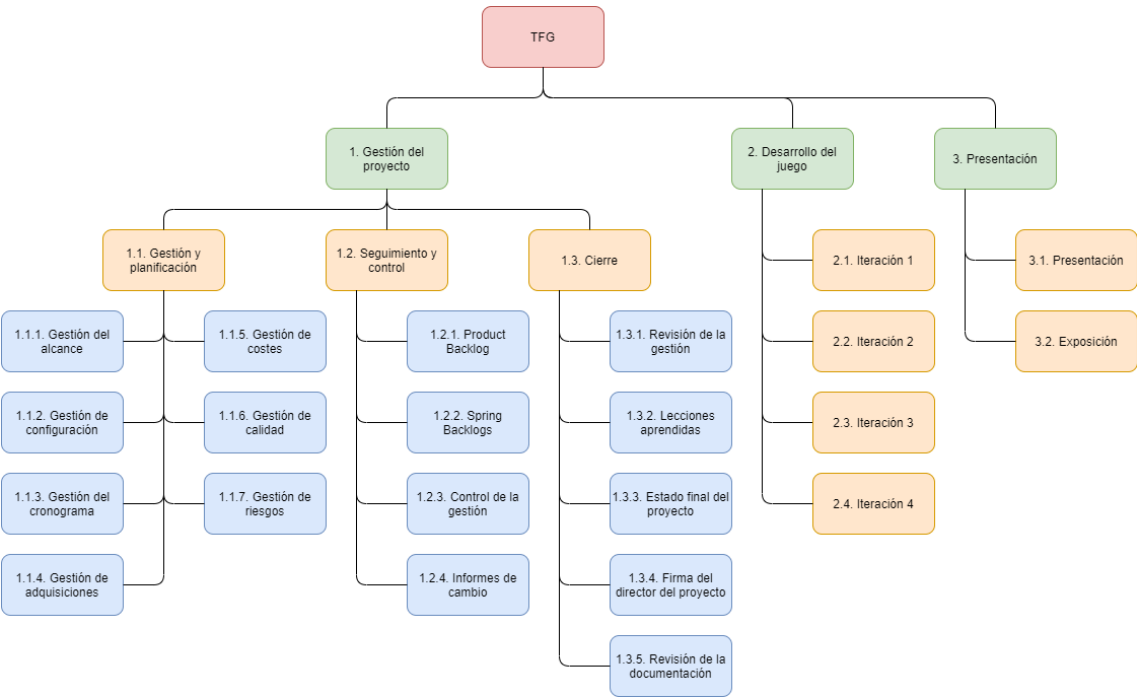


Figura 4.1: EDT

4.1.5 Diccionario de la EDT

2

1.1.1. Gestión del alcance	
Riesgos	El alcance del proyecto no se define correctamente.
Actividades	Definir el alcance. Establecer requisitos. Hacer matriz de trazabilidad. Hacer EDT. Redactar el diccionario de la EDT.
Criterios de aceptación	El alcance definido permite conocer lo que abarca el proyecto con facilidad.

Cuadro 4.8: Paquete de trabajo 1.1.1.

1.1.2. Gestión de configuración	
Riesgos	No se define correctamente la política de versionado de los entregables.
Actividades	Establecer la herramienta de versionado usada. Definir política de nombrado de versiones.
Criterios de aceptación	La política de versionado de los entregables se comprende fácilmente.

Cuadro 4.9: Paquete de trabajo 1.1.2.

1.1.3. Gestión del cronograma	
Riesgos	La cronología establecida es imposible de seguir.
Actividades	Estimar la duración de las actividades. Definir la línea base del cronograma.
Criterios de aceptación	El cronograma ha sido respetado en la medida de lo posible.

Cuadro 4.10: Paquete de trabajo 1.1.4.

1.1.4. Gestión de adquisiciones

Riesgos	No se ha realizado adecuadamente la adquisición de las herramientas.
Actividades	Obtener la documentación de las adquisiciones. Definir SOW.
Criterios de aceptación	Todas las adquisiciones se han realizado correctamente.

Cuadro 4.11: Paquete de trabajo 1.1.5.

1.1.5. Gestión de costes

Riesgos	El presupuesto ha sido calculado de forma errónea.
Actividades	Estimar los costes. Calcular el presupuesto.
Criterios de aceptación	Pueden entenderse la estimación de los costes y el cálculo del presupuesto fácilmente.

Cuadro 4.12: Paquete de trabajo 1.1.6.

1.1.6. Gestión de calidad

Riesgos	No puede juzgarse la calidad del trabajo realizado.
Actividades	Definir las métricas de usadas. Analizar los requisitos de calidad.
Criterios de aceptación	Todos los requisitos de calidad del proyecto han sido estudiados adecuadamente.

Cuadro 4.13: Paquete de trabajo 1.1.7.

1.1.7. Gestión de riesgos	
Riesgos	Los riesgos descritos no son representativos de los reales.
Actividades	Definir la metodología usada. Establecer seguimiento a realizar. Identificar y evaluar riesgos. Realizar análisis cualitativo y cuantitativo. Definir plan de respuesta.
Criterios de aceptación	Todos los riesgos han sido estudiados y se ha definido un plan de respuesta para cada uno.

Cuadro 4.14: Paquete de trabajo 1.1.8.

1.2.1. Product Backlog	
Riesgos	No todos los requisitos funcionales se encuentran correctamente representados por historias de usuarios.
Actividades	Crear todas las historias de usuario. Realizar las estimaciones.
Criterios de aceptación	El Product Backlog permite organizar el trabajo de las iteraciones.

Cuadro 4.15: Paquete de trabajo 1.2.1.

1.2.2. Sprint Backlog	
Riesgos	La organización de las tareas no permite su correcto desarrollo.
Actividades	Repartir las tareas en iteraciones. Configurar Zenhub para la iteración.
Criterios de aceptación	El Sprint Backlog permite organizar las tareas de desarrollo de una iteración.

Cuadro 4.16: Paquete de trabajo 1.2.2.

1.2.3. Control de la gestión	
Riesgos	No se ha realizado el seguimiento del cumplimiento de la gestión correctamente.
Actividades	Controlar la gestión del alcance. Controlar la gestión del cronograma. Controlar la gestión de los costes. Controlar la gestión de la calidad. Controlar la gestión de riesgos.
Criterios de aceptación	Todos los aspectos de la gestión del proyecto han sido controlados adecuadamente.

Cuadro 4.17: Paquete de trabajo 1.2.3.

1.2.4. Informes de cambio	
Riesgos	No todos los cambios han sido registrados.
Actividades	Revisar los cambios realizados. Rellenar los informes de cambio.
Criterios de aceptación	Todos los cambios están correctamente documentados.

Cuadro 4.18: Paquete de trabajo 1.2.4.

1.3.1. Revisión de la gestión	
Riesgos	No se han recopilado los controles la gestión realizados anteriormente.
Actividades	Revisar la gestión del alcance. Revisar la gestión del cronograma. Revisar la gestión de los costes. Revisar la gestión de la calidad. Revisar la gestión de riesgos.
Criterios de aceptación	Toda la gestión realizada durante el proyecto se encuentra documentado.

Cuadro 4.19: Paquete de trabajo 1.3.1.

1.3.2. Lecciones aprendidas	
Riesgos	Alguna complicación durante la ejecución del proyecto no ha sido documentada.
Actividades	Revisar el desempeño durante el proyecto. Obtener conclusiones a partir de la revisión.
Criterios de aceptación	Todo el proyecto ha sido revisado y se han sacado conclusiones provechosas para el futuro.

Cuadro 4.20: Paquete de trabajo 1.3.2.

1.3.3. Estado final del proyecto	
Riesgos	El estado del proyecto durante el cierre no se encuentra correctamente reflejado.
Actividades	Estudiar cumplimiento del alcance. Estudiar eficiencia del cronograma. Estudiar los costes finales. Estudiar la calidad.
Criterios de aceptación	El estado final del proyecto se encuentra perfectamente definido, así como las comparaciones con las expectativas iniciales.

Cuadro 4.21: Paquete de trabajo 1.3.3.

1.3.4. Firma del director del proyecto	
Riesgos	El proyecto no puede cerrarse por la falta de la firma.
Actividades	Obtener firma. Cerrar el proyecto.
Criterios de aceptación	El proyecto se ha cerrado definitivamente.

Cuadro 4.22: Paquete de trabajo 1.3.4.

1.3.5. Revisión de la documentación	
Riesgos	La documentación no se encuentra en un estado entregable.
Actividades	Definir estructura organizacional. Definir contexto del proyecto. Definir objetivos del proyecto. Definir metodologías usadas. Revisar la documentación de planificación. Revisar la documentación de seguimiento y control. Revisar la documentación de cierre.
Criterios de aceptación	La documentación es correcta y entregable.

Cuadro 4.23: Paquete de trabajo 1.3.5.

2.1. Iteración 1	
Riesgos	Se retrasa la ejecución de las siguientes iteraciones.
Actividades	Permitir el desarrollo de una partida. Implementar el movimiento básico. Implementar el movimiento del nivel.
Criterios de aceptación	Se han podido desarrollar todos los aspectos esperados durante esta iteración.

Cuadro 4.24: Paquete de trabajo 2.1.

2.2. Iteración 2	
Riesgos	Se retrasa la ejecución de las siguientes iteraciones.
Actividades	Implementar los obstáculos. Implementar los atajos. Implementar las zonas.
Criterios de aceptación	Se han podido desarrollar todos los aspectos esperados durante esta iteración.

Cuadro 4.25: Paquete de trabajo 2.2.

2.3. Iteración 3	
Riesgos	Se retrasa la ejecución de las siguientes iteraciones.
Actividades	Implementar la mecánica de la velocidad. Implementar la mecánica del portátil. Implementar la mecánica de los objetos.
Criterios de aceptación	Se han podido desarrollar todos los aspectos esperados durante esta iteración.

Cuadro 4.26: Paquete de trabajo 2.3.

2.4. Iteración 4	
Riesgos	Se retrasa el cierre del proyecto.
Actividades	Implementar menús. Implementar puntuaciones. Implementar cambio de controles.
Criterios de aceptación	Se han podido desarrollar todos los aspectos esperados durante esta iteración.

Cuadro 4.27: Paquete de trabajo 2.4.

3.1. Presentación	
Riesgos	No se representa adecuadamente el proyecto y el trabajo realizado.
Actividades	Resumir puntos más importantes del proyecto. Redactar presentación.
Criterios de aceptación	La presentación sirve de base para la realización de la exposición.

Cuadro 4.28: Paquete de trabajo 3.1.

3.2. Exposición	
Riesgos	No preparar adecuadamente la exposición y que de una mala imagen del proyecto.
Actividades	Preparar exposición. Realizar exposición.
Criterios de aceptación	La exposición permite entender a la perfección el proyecto de forma rápida y simple.

Cuadro 4.29: Paquete de trabajo 3.2.

1 4.2 GESTIÓN DE CONFIGURACIÓN

2 El versionado de los entregables del proyecto se llevará a cabo mediante el uso de
3 Github. Para ello se usa un repositorio (<https://github.com/Thiloparn/TFG>) en el
4 que se encuentran disponibles todas las versiones de los entregables. Concretamente
5 se realizará un release con la última versión disponible de cada entregable al final de
6 la planificación, de cada iteración y del cierre.

7 El método que se usará para el nombramiento de las versiones seguirá las siguien-
8 tes reglas:

- 9 ■ La primera versión completa y funcional se llamará 1.0.
- 10 ■ Cada vez que se realice un release incrementará el valor de versionando a X.0,
11 incrementándose X en una unidad respecto al valor anterior.
- 12 ■ Cada cambio sustancial realizado a raíz de una solicitud de cambio incrementará
13 el valor de versionando a X.Y, siendo Y el valor a incrementar en una unidad y X
14 manteniéndose invariable.

15 Sin embargo, el GDD seguirá un método de versionado distinto:

- 16 ■ La primera versión completa y funcional se llamará 1.0.
- 17 ■ Cada cambio sustancial realizado a raíz de una solicitud de cambio incrementará
18 el valor de versionando a X.0, incrementándose X en una unidad respecto al valor
19 anterior.
- 20 ■ Cada modificación minoritaria que no provenga de una solicitud de cambio in-
21 crementará el valor de versionando a X.Y, siendo Y el valor a incrementar una
22 unidad y X manteniéndose invariable.

23 Esta distinción entre las metodologías de versionado para cada documento es ne-
24 cesaria debido a que el versionado del GDD no debe verse afectado por las fases de
25 desarrollo del proyecto, y por tanto de la realización de release. También ha de tomar-
26 se en cuenta que la versión inicial y de partida del GDD en el que se basa el poroyecto,
27 es 3.1. Por tanto, respetando las reglas citadas anteriormente, se partirá de este versio-
28 nado cuando se realicen cambios en el documento debido a este proyecto.

Con respecto a la gestión de ramas, se distinguirán las ramas "develop" y "main". La primera es aquella que podrá modificar libremente el desarrollador y subir las versiones intermedias de los ficheros. La segunda no recibirá cambios directamente, sino que será mergeada con develop al final de cada fase del proyecto, antes de llevar a cabo los release. Esto permite aislar los errores que puedan surgir en el juego en la rama "develop", manteniendo una versión siempre estable en "main".

4.3 GESTIÓN DEL CRONOGRAMA

4.3.1 Lista y estimación de actividades

Actividades paquete 1.1.1.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.1.1.A	Definir el alcance	25/02/21 – 25/02/21	1.5h	21€
1.1.1.B	Establecer requisitos	25/02/21 – 25/02/21	2h	28€
1.1.1.C	Hacer EDT	26/02/21 – 26/02/21	1.5h	21€
1.1.1.D	Redactar el diccionario de la EDT	26/02/21 – 26/02/21	1.5h	21€
Total		25/02/21 – 26/02/21	7h	98€

Cuadro 4.30: Actividades paquete 1.1.1.

Actividades paquete 1.1.2.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.1.2.A	Establecer la herramienta de versionado usada	27/02/21 – 27/02/21	0.25h	3,50€
1.1.2.B	Definir política de nombrado de versiones	27/02/21 – 27/02/21	0.25h	3,50€
Total		27/02/21 – 27/02/21	0.5h	7€

Cuadro 4.31: Actividades paquete 1.1.2.

Actividades paquete 1.1.3.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.1.3.A	Estimar la duración de las actividades	28/02/21 – 28/02/21	5.5h	77€
1.1.3.B	Definir la línea base del cronograma	01/02/21 – 02/03/21	2.5	35€
Total		28/02/21 – 02/03/21	8h	112€

Cuadro 4.32: Actividades paquete 1.1.3.

Actividades paquete 1.1.4.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.1.4.A	Justificar las adquisiciones	03/03/21 – 03/03/21	1h	14€
1.1.4.B	Listado de adquisiciones	03/03/21 – 03/03/21	1h	14€
Total		03/03/21 – 03/03/21	2h	28€

Cuadro 4.33: Actividades paquete 1.1.4.

Actividades paquete 1.1.5.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.1.5.A	Estimar los costes	03/03/21 – 03/03/21	0.5h	7€
1.1.5.B	Calcular el presupuesto	03/03/21 – 03/03/21	0.25h	3,50€
Total		03/03/21 – 03/03/21	0.75h	10,50€

Cuadro 4.34: Actividades paquete 1.1.5.

Actividades paquete 1.1.6.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.1.6.A	Definir las métricas usadas	03/03/21 – 03/03/21	0.25h	3,50€
1.1.6.B	Analizar los requisitos de calidad	03/03/21 – 03/03/21	0.75h	10,50€
Total		03/03/21 – 03/03/21	1h	14€

Cuadro 4.35: Actividades paquete 1.1.6.

Actividades paquete 1.1.7.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.1.7.A	Identificar y evaluar riesgos	04/03/21 – 04/03/21	1h	14€
1.1.7.B	Realizar análisis cualitativo y cuantitativo	04/03/21 – 04/03/21	1h	14€
1.1.7.C	Definir plan de respuesta	04/03/21 – 04/03/21	1h	14€
Total		04/03/21 – 04/03/21	3h	42€

Cuadro 4.36: Actividades paquete 1.1.7.

Actividades paquete 1.2.1.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.2.1.A	Crear todas las historias de usuario	08/03/21 – 08/03/21	1.5h	21€
1.2.1.B	Realizar estimaciones	08/03/21 – 08/03/21	1h	14€
Total		08/03/21 – 08/03/21	2.5h	35€

Cuadro 4.37: Actividades paquete 1.2.1.

Actividades paquete 1.2.2.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.2.2.A	Repartir las tareas en iteraciones	08/03/21 – 10/05/21	0.5h	7€
1.2.2.B	Configurar Zenhub para la iteración	08/03/21 – 10/05/21	0.5h	7€
Total		08/03/21 – 10/05/21	1h	14€

Cuadro 4.38: Actividades paquete 1.2.2.

Actividades paquete 1.2.3.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.2.3.A	Controlar la gestión del alcance	08/03/21 – 25/05/21	5h	75€
1.2.3.B	Controlar la gestión del cronograma	08/03/21 – 26/05/21	5h	75€
1.2.3.C	Controlar la gestión de los costes	08/03/21 – 27/05/21	5h	75€
1.2.3.D	Controlar la gestión de la calidad	08/03/21 – 28/05/21	5h	75€
1.2.3.E	Controlar la gestión de riesgos	08/03/21 – 29/05/21	5h	75€
Total		08/03/21 – 29/05/21	25h	350€

Cuadro 4.39: Actividades paquete 1.2.1.

Actividades paquete 1.2.4.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.2.4.A	Revisar los cambios realizados	08/03/21 – 30/05/21	1h	14€
1.2.4.B	Rellenar los informes de cambio	08/03/21 – 30/05/21	3h	43€
Total		08/03/21 – 30/05/21	4h	56€

Cuadro 4.40: Actividades paquete 1.2.4.

Actividades paquete 1.3.1.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.3.1.A	Revisar la gestión del alcance	31/05/21 – 31/05/21	2h	28€
1.3.1.B	Revisar la gestión del cronograma	01/06/21 – 01/06/21	2h	28€
1.3.1.C	Revisar la gestión de los costes	02/06/21 – 02/06/21	2h	28€
1.3.1.D	Revisar la gestión de la calidad	03/06/21 – 03/06/21	2h	28€
1.3.1.E	Revisar la gestión de riesgos	04/06/21 – 04/06/21	2h	28€
Total		31/05/21 – 04/06/21	10h	140€

Cuadro 4.41: Actividades paquete 1.3.1.

Actividades paquete 1.3.2.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.3.2.A	Revisar el desempeño durante el proyecto	05/06/21 – 05/06/21	1h	14€
1.3.2.B	Obtener conclusiones a partir de la revisión	05/06/21 – 05/06/21	2h	28€
Total		05/06/21 – 05/06/21	3h	42€

Cuadro 4.42: Actividades paquete 1.3.2.

Actividades paquete 1.3.3.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.3.3.A	Estudiar cumplimiento del alcance	05/06/21 – 05/06/21	1h	14€
1.3.3.B	Estudiar eficiencia del cronograma	05/06/21 – 05/06/21	1h	14€
1.3.3.C	Estudiar los costes finales	06/06/21 – 06/06/21	1h	14€
1.3.3.D	Estudiar la calidad	06/06/21 – 06/06/21	1h	14€
Total		05/06/21 – 06/06/21	4h	56€

Cuadro 4.43: Actividades paquete 1.3.3.

Actividades paquete 1.3.4.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.3.4.A	Obtener firma	06/06/21 – 06/06/21	0.25h	3,50€
1.3.4.B	Cerrar el proyecto	06/06/21 – 06/06/21	0.25h	3,50€
Total		06/06/21 – 06/06/21	0.5h	7€

Cuadro 4.44: Actividades paquete 1.3.4.

Actividades paquete 1.3.5.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
1.3.5.A	Definir estructura organizacional	07/06/21 – 07/06/21	0.5h	7€
1.3.5.B	Definir contexto del proyecto	07/06/21 – 07/06/21	1h	14€
1.3.5.C	Definir objetivos del proyecto	07/06/21 – 07/06/21	1h	14€
1.3.5.D	Definir metodologías usadas	07/06/21 – 07/06/21	2h	28€
1.3.5.E	Revisar la documentación de planificación	08/06/21 – 08/06/21	1h	14€
1.3.5.F	Revisar la documentación de seguimiento y control	08/06/21 – 08/06/21	1h	14€
1.3.5.G	Revisar la documentación de cierre	08/06/21 – 08/06/21	1h	14€
Total		07/06/21 – 08/06/21	7.5h	105€

Cuadro 4.45: Actividades paquete 1.3.5.

Actividades paquete 2.1.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
2.1.A	Permitir el desarrollo de una partida	08/03/21 – 12/03/21	6h	84€
2.1.B	Implementar el movimiento básico	13/03/21 – 17/03/21	10h	140€
2.1.C	Implementar el movimiento del nivel	18/03/21 – 22/03/21	6h	84€
Total		08/03/21 – 28/03/21	22h	308€

Cuadro 4.46: Actividades paquete 2.1.

Actividades paquete 2.2.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
2.2.A	Implementar los obstáculos	29/03/21 – 02/04/21	9h	126€
2.2.B	Implementar los atajos	03/04/21 – 07/04/21	10h	140€
2.2.C	Implementar las zonas	08/04/21 – 12/02/21	7h	98€
Total		29/03/21 – 18/04/21	26h	369€

Cuadro 4.47: Actividades paquete 2.2.

Actividades paquete 2.3.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
2.3.A	Implementar la mecánica de la velocidad	19/04/21 – 23/04/21	12h	168€
2.3.B	Implementar la mecánica del portátil	24/04/21 – 28/04/21	7h	98€
2.3.C	Implementar la mecánica de los objetos	29/04/21 – 03/05/21	7h	98€
Total		19/04/21 – 09/05/21	26h	369€

Cuadro 4.48: Actividades paquete 2.3.

Actividades paquete 2.4.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
2.4.A	Implementar menús	10/05/21 – 14/05/21	11h	154€
2.4.B	Implementar puntuaciones	15/05/21 – 19/05/21	4h	56€
2.4.C	Implementar cambio de controles	20/05/21 – 24/05/21	5h	70€
Total		10/05/21 – 30/05/21	20h	280€

Cuadro 4.49: Actividades paquete 2.4.

Actividades paquete 3.1.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
3.1.A	Resumir puntos más importantes del proyecto	07/06/21 – 07/06/21	0.5h	7€
3.1.B	Redactar presentación	07/06/21 – 10/06/21	2h	28€
Total		07/06/21 – 10/06/21	2.5h	35€

Cuadro 4.50: Actividades paquete 3.1.

Actividades paquete 3.2.				
Id	Nombre	Fecha de inicio y fin	Estimación de duración	Coste estimado
3.2.A	Preparar exposición	11/06/21 – 11/07/21	1h	14€
3.2.B	Realizar exposición	12/07/21 – 16/07/21	0.5h	7€
Total		11/06/21 – 16/07/21	1.5h	21€

Cuadro 4.51: Actividades paquete 3.2.

4.3.2 Línea base del cronograma

Cronograma

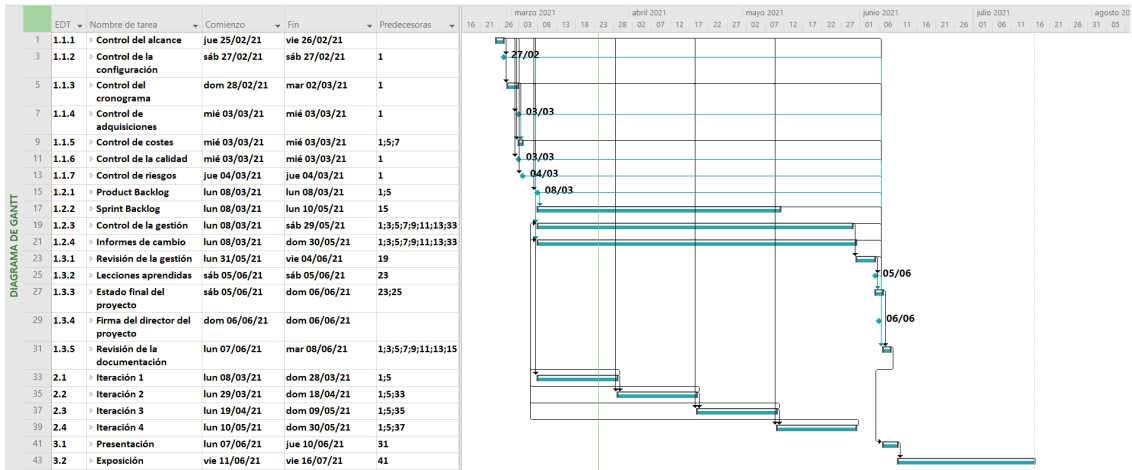


Figura 4.2: Cronograma

Hitos

A lo largo del desarrollo del proyecto se podrán encontrar 7 hitos: planificación, uno por cada iteración, cierre, y presentación. Cada hito irá acompañado de un release en el repositorio de github del proyecto (<https://github.com/Thiloparn/TFG>) formado por todos los entregables obtenidos hasta la fecha. Las fechas límites para la realización de estos hitos son:

- Planificación: 07/03/21.
- Iteración 1: 28/03/21.
- Iteración 2: 18/04/21.
- Iteración 3: 09/05/21.
- Iteración 4: 30/05/21.
- Cierre: 08/06/21.
- Presentación: 10/06/21.

1 Cadena crítica

2 A continuación se listan todas aquellas actividades cuya no realización o severo
3 retraso supondrían un gran peligro para la correcta realización del proyecto:

- 4 ■ Todas las actividades de la fase de planificación: Todas permiten entender el pro-
5 yecto y organizar el desarrollo del videojuego.
- 6 ■ Las actividades de la primera Iteración: Estas satisfacen requisitos de la más alta
7 prioridad en el proyecto y permiten el desarrollo de las siguientes Iteraciones.
- 8 ■ 2.2.A, 2.4.A: Igual que en le caso de la Iteración 1, estas actividades del resto de
9 Iteraciones satisfacen los requisitos más importantes del videojuego.
- 10 ■ Todas las actividades de la fase de cierre: Permiten obtener conocimiento sobre el
11 desarrollo del proyecto en su globalidad, aprender de errores cometidos, y cerrar
12 el proyecto definitivamente.
- 13 ■ Todas las actividades de la fase de presentación: Sin ellas no se podría exponer
14 lo transcurrido durante el proyecto, y por tanto se desconocerían los aspectos
15 positivos y negativos del trabajo realizado.

16 4.4 GESTIÓN DE ADQUISICIONES

17 4.4.1 Justificación de las adquisiciones

18 Lo primero que se requiere para llevar a cabo el proyecto es una plataforma de
19 desarrollo de videojuegos. Como se ha mencionado anteriormente, esta debe ser obli-
20 gatoriamente Unity.

21 El código generado por Unity así como toda la documentación generada durante el
22 proyecto deben ser almacenados de forma seguro y permitir un versionado adecuado.
23 Para ello se usará Github. Además, de forma privada para facilitar el trabajo del desa-
24 rrollador en distintos equipos informáticos, se usará por la misma razón One Drive.

25 Para poder generar toda la documentación, se usará Overleaf para el uso de LaTeX.
26 Puesto que algunos elementos de la documentación no pueden generarse mediante el
27 uso de esta herramienta, se usará Microsoft Office para la creación de la presentación
28 y el cronograma entre otros, y Diagramas.net para obtener esquemas como la EDT.

Finalmente, se requerirá la adquisición de dos herramientas adicionales para el correcto seguimiento del trabajo realizado durante el proyecto. Estas son Clockify, para medir el tiempo dedicado a cada actividad, y Zenhub (una extensión de Github), para la organización durante las Iteraciones y la obtención de información acerca del desempeño.

4.4.2 Listado de adquisiciones

Todas las adquisiciones que se listan a continuación tienen las siguientes características en común:

- Tipo de adquisición: Servicio.
- Tipo de contrato: Request for proposal (RFP).
- Riesgos:
 - No mantener sus funcionalidades gratuitas.
 - Caída de la plataforma online (salvo Unity y Microsoft Office).
 - La versión usada queda totalmente obsoleta en un futuro cercano (únicamente Unity).
- Coste máximo: Debido a que la mayoría de herramientas usadas son de uso gratuito y Microsoft Office también lo es por ser el desarrollador un estudiante de la Universidad de Sevilla, todos los costes de adquisición son de 0€.

Listado de adquisiciones			
Id	Nombre	Página web	Descripción
A01	Unity	https://unity.com	Motor de desarrollo de videojuegos
A02	Github	https://github.com	Plataforma de desarrollo colaborativo basado en el sistema de control de versiones Git
A03	One Drive	https://onedrive.com	Servicio de almacenamiento de archivos en la nube
A04	Overleaf	https://overleaf.com	Editor colaborativo de LaTeX en la nube
A05	Microsoft Office	https://office.com/	Paquete de aplicaciones de escritorio y servicios desarrollados por Microsoft
A06	Diagrams.net	https://draw.io	Editor online de diagramas relacionados con el desarrollo de Software
A07	Clockify	https://clockify.me	Software de seguimiento temporal centrado en el desarrollo de proyectos
A08	Zenhub	https://zenhub.com	Herramienta de administración de proyecto ágiles integrada en Github

Cuadro 4.52: Listado de adquisiciones

4.5 GESTIÓN DE COSTES

1

4.5.1 Estimación de costes

2

Para estimar los costes, se partirá de las estimaciones de tiempo y costes por paquete de trabajo realizado en la gestión del cronograma (apartado 3). Con esto, se podrá estimar la duración de cada fase del proyecto y después calcular el salario bruto del desarrollador, sabiendo que es de 14€ por hora. Una vez hecho esto, se calculará el coste del proyecto para el desarrollador, partiendo de que el coste de la seguridad social es del 30 por ciento.

3

4

5

6

7

8

Estimación de costes					
	Planificación	Iteraciones	Cierre	Presentación	Total
Duración	22.25h	126.5h	17.5h	4h	170.25h
Salario bruto	311,50€	1.771€	245€	56€	2.384,50€
Seguridad social	93,45€	531,30€	73,50€	16,80€	715,05€
Salario final	404,95€	2.302,30€	318,50€	72,80€	3.098,55€

Cuadro 4.53: Estimación de costes

Además de esto, se debe considerar las amortizaciones del equipo informático del desarrollador. Para la realización del proyecto se le ha sido entregado el modelo MSI Modern 14 B10MW-215XES, de coste 849€. Al ser un equipo nuevo, la amortización será del 20 por ciento, suponiendo un coste de 1.018,80€.

No se considerarán más costes adicionales debido a que todas las licencias de software obtenidas han resultado gratuitas. Además, al tratarse de un único desarrollador, este podrá trabajar desde su vivienda, suponiendo así un coste de alquiler de oficina nulo.

4.5.2 Presupuesto

Para el calculo del presupuesto base, se añadirá a lo calculado anteriormente un 10 por ciento del mismo como fondos de reserva, así como el 21 por ciento de IVA.

Presupuesto	
Concepto	Coste
Recurso humano	3.098,55€
Amortizaciones	1.018,80€
Presupuesto base	4.117,35€
Fondo de reserva	411,74€
Impuesto sobre valor añadido (IVA)	864,65€
Presupuesto total	5.393,74€

Cuadro 4.54: Presupuesto

4.6 GESTIÓN DE LA CALIDAD

Para analizar las métricas de calidad se usará el sistema GQM (Question, Goal, Metric), que como su nombre indica, se basa en hacer una serie de preguntas acerca de los requisitos de calidad, y obtener respuestas a estas que indiquen como gestionar estos requisitos.

Todo ello esta recogido en la tabal a continuación:

Análisis de la calidad		
Objetivo	Pregunta	Métrica
RC01: El juego no posee ningún fallo	¿Existen bugs visuales? ¿Funcionan todas las mecánicas? ¿Una partida puede desarrollarse correctamente?	Realizar partidas de prueba
RC02: Existe un tutorial que explica el funcionamiento del juego al jugador	¿Existe una explicación del funcionamiento del juego? ¿El jugador entiende como se juega? ¿El tutorial es de duración aceptable?	Pedir a testers que prueben el tutorial
RC03: Los tiempos de carga son cortos	¿El jugador espera mucho para comenzar partida? ¿El juego requiere tiempos de carga durante la partida?	Realizar partidas de prueba
RC04: Se respeta el aspecto estético descrito en el GDD	¿Se cumplen las características del estilo artístico?	Revisar la descripción de la estética del juego en el GDD

Cuadro 4.55: Análisis de la calidad

1 4.7 GESTIÓN DE RIESGOS

2 4.7.1 Identificación y revaluación de riesgos

3 A continuación se listarán todos los riesgos principales del proyecto, distinguiendo
4 las siguientes características:

- 5 ■ Implicación: Interna, si se surge debido al desarrollador, o externa, si es provoca-
6 do por elementos ajenos al proyecto.
- 7 ■ Tipo: Técnico, derivado de la gestión del proyecto, de la organización o por la
8 existencia de alguna dependencia ajena.

Lista de riesgos negativos			
Id	Descripción	Implicación	Tipo
IRN01	Desconocimiento de las tecnologías usadas	Interna	Técnica
IRN02	Planificación del proyecto errónea	Interna	Gestión del proyecto
IRN03	Control y seguimiento del proyecto erróneo	Interna	Gestión del proyecto
IRN04	Imposibilidad de cerrar el proyecto	Interna	Gestión del proyecto
IRN05	Presentación y exposición no representativos del proyecto	Interna	Gestión del proyecto
IRN06	Incapacidad de seguir el plan de las Iteraciones	Interna	Organización
IRN07	No cumplimiento de las fechas de entrega	Interna	Organización
IRN08	Mal funcionamiento de las tecnologías	Externa	Dependencia ajena
IRN09	Obligaciones del desarrollador le impiden llevar a cabo el proyecto de forma planificada	Externa	Dependencia ajena
IRN10	Imposibilidad de satisfacer el GDD	Externa	Dependencia ajena

Cuadro 4.56: Lista de riesgos negativos

Lista de riesgos positivos			
Id	Descripción	Implicación	Tipo
IRP01	Buen conocimiento de las tecnologías por parte del desarrollador	Interna	Técnica
IRP02	Planificación del proyecto permite su correcta realización	Interna	Gestión del proyecto
IRP03	Cumplimiento de las expectativas de la planificación	Interna	Organización
IRP04	Cumplimiento de las estimaciones temporales	Interna	Organización
IRP05	GDD más sencillo de satisfacer que lo esperado	Externa	Dependencia ajena

Cuadro 4.57: Lista de riesgos positivos

4.7.2 Análisis cualitativo de los riesgos

1

Para evaluar el nivel de cada uno de los riesgos que se han citado anteriormente se
estudirá el impacto y la posibilidad de que ocurra cada uno.

2

3

Relación impacto-probabilidad de los riesgos						
Impacto	Muy alto	Medio	Medio	Alto	Alto	Alto
	Alto	Medio	Medio	Medio	Alto	Alto
	Medio	Bajo	Medio	Medio	Medio	Alto
	Bajo	Bajo	Bajo	Medio	Medio	Medio
	Muy bajo	Bajo	Bajo	Bajo	Medio	Medio
		Muy baja	Baja	Media	Alta	Muy alta
		Probabilidad				

Cuadro 4.58: Relación impacto-probabilidad de los riesgos

1 4.7.3 Análisis cuantitativo de los riesgos

2 Basandose en la tabla anterior, se obtiene el nivel de cada riesgo:

Análisis cuantitativo de los riesgos			
Id	Probabilidad	Impacto	Nivel
IRN01	Muy bajo	Medio	Bajo
IRN02	Baja	Muy alto	Medio
IRN03	Baja	Bajo	Bajo
IRN04	Baja	Muy alto	Medio
IRN05	Muy baja	Alto	Medio
IRN06	Baja	Medio	Medio
IRN07	Muy bajo	Muy alto	Medio
IRN08	Muy bajo	Alto	Medio
IRN09	Medio	Alto	Medio
IRN10	Alta	Baja	Medio
IRP01	Muy alta	Alto	Alto
IRP02	Alta	Medio	Medio
IRP03	Alta	Alto	Alto
IRP04	Medio	Medio	Medio
IRP05	Baja	Muy alta	Medio

Cuadro 4.59: Análisis cuantitativo de los riesgos

4.7.4 Plan de respuesta a los riesgos 1

Por último, se establecen las acciones a tomar para reducir los riesgos y aumentar las oportunidades de completar con éxito los objetivos del proyecto. 2 3

Se definirá para cada riesgo negativo un plan de contingencia para intentar evitar dicho riesgos y un plan de mitigación para solventar el riesgo si llegase a ocurrir. 4 5

Plan de respuesta para los riesgos negativos		
Id	Plan de contingencia	Plan de mitigación
IRN01	Usar únicamente tecnología conocidas en profundidad	Retrasar uso de la tecnología para aprender su funcionamiento
IRN02	Comparar planificación con la de otro proyectos para ver posibles errores	Modificar planificación para adaptarse a la realidad
IRN03	Comparar control y seguimiento con los de otro proyectos para ver posibles errores	Modificar control y seguimiento para adaptarse a la realidad
IRN04	Seguir planificación según lo previsto mientras no acarée problemas futuros	Revisar proyecto para solventar los errores y cerrar el proyecto
IRN05	Estudiar con detenimiento el desarrollo del proyecto para realizar una buena presentación y exposición	Buscar ayuda de una persona externa al proyecto para encontrar fallos en la presentación
IRN06	Realizar las estimaciones de tiempo considerando posibles fallo e investigaciones	Realizar solo aquellas tareas de mayor prioridad
IRN07	Acabar proyecto con antelación para poder preparar entrega	Preparar próximas entregas lo antes posible
IRN08	Usar tecnologías poco susceptible de mal funcionamiento	Realizar adquisición de nuevas tecnologías que funcionen según lo requerido
IRN09	Realizar estimaciones temporales que contemplen la posible falta de tiempo del desarrollador en lo referente al proyecto	Retrasar fechas claves lo máximo posible sin peligrar el proyecto
IRN10	Estudiar el GDD en profundidad para buscar posibles errores de cara al futuro	Adaptar GDD a las posibilidades del proyecto

Cuadro 4.60: Plan de respuesta para los riesgos negativos

En cuanto a los riesgos positivos se definirán planes de potenciación para aumentar las posibilidades de que el riesgo tenga lugar, y planes de aprovechamiento para maximizar el impacto en el proyecto.

1
2
3

Plan de respuesta para los riesgos positivos		
Id	Plan de potenciación	Plan de aprovechamiento
IRP01	Usar únicamente tecnología conocidas en profundidad	Usar conocimientos avanzados de las tecnologías para facilitar el desarrollo del proyecto
IRP02	Realizar la planificación lo más detallada posible	Centrarse en el desarrollo del videojuego para mejorar sus características
IRP03	Realizar la planificación de forma realista y preventiva	Mejorar las documentaciones obtenidas para que los entregables superen las expectativas
IRP04	Realizar estimaciones temporales de forma realista y preventiva	Satisfacer obligaciones del desarrollador de forma preventiva para minimizar futuros problemas
IRP05	Planificar el proyecto tomando en cuenta errores o fallos poco probables	Mejorar al máximo el videojuego respetando el GDD y sin peligrar el proyecto

Cuadro 4.61: Plan de respuesta para los riesgos positivos

PARTE III

DESARROLLO DEL PROYECTO

PRODUCT BACKLOG

1

2 *E*n este capítulo se estructurarán las historias de usuario necesarias para el desarrollo
3 del proyecto en un Product Backlog. Este permitirá la repartición de las historias en
4 iteraciones futuras.

A continuación puede verse el Product Backlog generado para comenzar el desarrollo de las iteraciones de trabajo. Notar que las reparticiones de las histroias de usuario entre las iteraciones se encuentra sujeto a cambio y no será definitivo hasta el establecimiento de los Sprint Backlog correspondientes.

Product Backlog					
Id	Descripción	Requisito	Iteración	Estimación temporal	Puntos de HU
1	Como jugador <i>quiero</i> que exista un botón que pare la ejecución del juego <i>para</i> poder dejar de jugar	RF03	1	1h	1
2	Como jugador <i>quiero</i> que exista un botón que inicie una partida <i>para</i> poder comenzar a jugar	RF04	1	1h	1
3	Como jugador <i>quiero</i> que exista una zona al final del nivel <i>para</i> poder terminar la partida	RF14	1	4h	4
4	Como jugador <i>quiero</i> poder moverme a la derecha y a la izquierda <i>para</i> poder navegar por el nivel	RF06	1	2h	2
5	Como jugador <i>quiero</i> poder saltar <i>para</i> poder esquivar obstáculos	RF06	1	4h	4
6	Como jugador <i>quiero</i> poder deslizarme <i>para</i> poder esquivar obstáculos	RF06	1	4h	4
7	Como jugador <i>quiero</i> que los elementos del nivel se desplacen <i>para</i> poder navegar por el nivel	RF07	1	6h	6
8	Como jugador <i>quiero</i> que puedan aparecer obstáculos <i>para</i> poder esquivarlos	RF11	2	3h	3
9	Como jugador <i>quiero</i> poder chocarme con los obstáculos <i>para</i> que sea necesario esquivarlos	RF11	2	4h	4

Product Backlog					
Id	Descripción	Requisito	Iteración	Estimación temporal	Puntos de HU
10	Como jugador <i>quiero</i> que existan una gran lista de obstáculos <i>para</i> que exista diversidad	RF11	2	2h	2
11	Como jugador <i>quiero</i> que puedan aparecer atajos <i>para</i> poder usarlos	RF12	2	3h	3
12	Como jugador <i>quiero</i> poder usar los atajos <i>para</i> aprovechar su mecánica	RF12	2	2h	2
13	Como jugador <i>quiero</i> que los atajos me desplacen por el nivel <i>para</i> saltarme tramos de este	RF12	2	5h	5
14	Como jugador <i>quiero</i> que el nivel esté dividido en zonas <i>para</i> que exista diversidad	RF13	2	3h	3
15	Como jugador <i>quiero</i> que cada nivel posea su lista de obstáculos y atajos <i>para</i> que exista diversidad entre zonas	RF13	2	4h	4
16	Como jugador <i>quiero</i> poder ver los niveles de velocidad <i>para</i> saber a qué velocidad me desplazo	RF08	3	3h	3
17	Como jugador <i>quiero</i> poder aumentar mi velocidad <i>para</i> desplazarme más rápido	RF08	3	4h	4
18	Como jugador <i>quiero</i> perder velocidad al chocar con obstáculos <i>para</i> que sea necesario esquivarlos	RF08	3	5h	5

Product Backlog					
Id	Descripción	Requisito	Iteración	Estimación temporal	Puntos de HU
19	Como jugador <i>quiero</i> poder ver el nivel de integridad del portátil <i>para</i> saber cuántas veces más puedo golpear	RF09	3	4h	4
20	Como jugador <i>quiero</i> poder golpear los obstáculos <i>para</i> poder romperlos	RF09	3	4h	4
21	Como jugador <i>quiero</i> poder abrir la interfaz de objetos y navegar por ella <i>para</i> usar los objetos	RF10	3	5h	5
22	Como jugador <i>quiero</i> que cada objeto cumpla su función <i>para</i> poder ayudarme en mi partida	RF10	3	2h	2
23	Como jugador <i>quiero</i> poder ver el cálculo de mi puntuación <i>para</i> poder saber mi desempeño en la partida	RF15	4	3h	3
24	Como jugador <i>quiero</i> poder registrar mi nombre junto a mi puntuación <i>para</i> ser reconocido de forma online	RF16	4	3h	3
25	Como jugador <i>quiero</i> poner el juego en pausa <i>para</i> poder salir del juego si lo deseo	RF05	4	5h	5
26	Como jugador <i>quiero</i> ver mis puntuaciones y las de otros jugadores <i>para</i> conocer mi desempeño y el de otros jugadores	RF01	4	4h	4

Product Backlog					
Id	Descripción	Requisito	Iteración	Estimación temporal	Puntos de HU
27	Como jugador <i>quiero</i> poder ver los controles del juego y cambiar sus asignaciones <i>para</i> que sea más cómodo para mi jugar	RF02	4	5h	5

Cuadro 5.1: Product Backlog

ITERACIÓN 1

1

2 *E* n este capítulo se realizará el seguimiento y control del trabajo realizado durante la
3 primera Iteración de desarrollo.

6.1 SPRINT BACKLOG

1

A continuación se establecen las tareas que pertenecen a la primera Iteración en el Sprint Backlog de forma definitiva.

2

3

Sprint Backlog Iteración 1					
Id	Descripción	Requisito	Iteración	Estimación temporal	Puntos de HU
1	Como jugador <i>quiero</i> que exista un botón que pare la ejecución del juego <i>para</i> poder dejar de jugar	RF03	1	1h	1
2	Como jugador <i>quiero</i> que exista un botón que inicie una partida <i>para</i> poder comenzar a jugar	RF04	1	1h	1
3	Como jugador <i>quiero</i> que exista una zona al final del nivel <i>para</i> poder terminar la partida	RF14	1	4h	4
4	Como jugador <i>quiero</i> poder moverme a la derecha y a la izquierda <i>para</i> poder navegar por el nivel	RF06	1	2h	2
5	Como jugador <i>quiero</i> poder saltar <i>para</i> poder esquivar obstáculos	RF06	1	4h	4
6	Como jugador <i>quiero</i> poder deslizarme <i>para</i> poder esquivar obstáculos	RF06	1	4h	4
7	Como jugador <i>quiero</i> que los elementos del nivel se desplacen <i>para</i> poder navegar por el nivel	RF07	1	6h	6

Cuadro 6.1: Sprint Backlog Iteración 1

6.2 CONTROL DE LA GESTIÓN1

6.2.1 Control del alcance2

En este apartado se lleva a cabo un seguimiento de la realización de historias de usuarios durante la primera iteración.34

Control del alcance Iteración 1						
Id	Estado	Tiempo estimado	Tiempo Real	Coste estimado	Coste real	Comentarios
HU01	Hecho	1h	0.5h	14€	7€	-
HU02	Hecho	1h	0.4h	14€	5,60€	-
HU03	Hecho	4h	1.2h	56€	16,80€	-
HU04	Hecho	2h	1.3h	28€	18,20€	-
HU05	Hecho	4h	1.8h	56€	25,20€	-
HU06	Hecho	4h	1h	56€	14€	-
HU07	Hecho	6h	6.2h	84€	86,80€	La forma en la que ha estado programada esta funcionalidad difiere de la definida en el GDD.

Cuadro 6.2: Control del alcance Iteración 1

Gracias a la herramienta de Zenhub integrada con Github, se ha podido obtener el siguiente Burndown Report:56



Figura 6.1: Burndown Iteración 1

Con esta gráfica puede verse que se han respetado las fechas establecidas para la realización de la iteración y de sus historias de usuario.

Nótese que los puntos de historias de usuarios por realizar se quedan a zero mucho antes de finalizar la iteración. Esto es debido a que los últimos cinco días de todas las iteraciones son dedicados a la realización de documentación. También sirven de tiempo extra si una historia de usuario requiere para su realización más tiempo de lo estimado. De esta forma, un error de estimación difícilmente puede provocar el retraso de una iteración.

También ha de destacarse que las primeras historias de usuario se completaron en las fechas esperadas, pero un error de uso de Zenhub ha llevado a que en esta gráfica no salga reflejado como tal.

6.2.2 Control del cronograma

A continuación se lleva a cabo un seguimiento del cronograma focalizado en las actividades de la primera iteración.

Control del cronograma Iteración 1					
Id	Nombre	Tiempo estimado	Tiempo real	Coste estimado	Coste real
2.1.A	Permitir el desarrollo de una partida	6h	2.1h	84€	29,40€
2.1.B	Implementar el movimiento básico	10h	4.1h	140€	57,40€
2.1.C	Implementar el movimiento del nivel	6h	6.2h	84€	86,80€
Total		22h	12.4h	308€	173,60€

Cuadro 6.3: Control del cronograma Iteración 1

En esta iteración se ha realizado una gran sobreestimación en cuanto a la duración de las actividades. Esto es debido a que estas actividades resultaron mucho más sencillas de realizar de lo esperado. Sin embargo, estas desviaciones en las estimaciones no afectan a la correcta realización del proyecto.

1 6.2.3 Control de los costes

2 En este apartado se realizará una comparativa de las estimaciones de duración de
 3 la primera iteración con los valores reales, así como su repercusión en los costes del
 4 proyecto. Puesto que las únicas variaciones que se hayan sufrido son las de horas tra-
 5 bajadas del desarrollador, solo se estudiarán las desviaciones de su salario, sin tomar
 6 en cuenta las actividades de seguimiento y control.

Control de costes Iteración 1			
	Estimación	Valor real	Desvío
Duración	22h	12.4h	-9.6h
Salario bruto	308€	173,60€	-134,40€
Seguridad social (30 por ciento)	92,40€	52,08€	-39,92€
Salario final	400,40€	225,68€	-174,72€

Cuadro 6.4: Control de costes Iteración 1

Las grandes desviaciones de tiempo en llevado a una diferencia de 174.72€ en el salario del desarrollador. Puesto que la diferencia es negativa, es decir, el salario real es menor al estimado, esta desviación no supone una amenaza para el correcto desarrollo del proyecto.

6.2.4 Control de la calidad

A continuación se listan las respuestas que se han dado durante la primera iteración a las preguntas planteadas acerca de la satisfacción de los requisitos de calidad.

Control de la calidad Iteración 1		
Objetivo	Pregunta	Respuestas
RC01	¿Existen bugs visuales?	Las animaciones no parecen naturales pero no conllevan ningún bug.
	¿Funcionan todas las mecánicas?	Sí.
	¿Una partida puede desarrollarse correctamente?	Sí.
RC02	¿Existe una explicación del funcionamiento del juego?	No procede para esta iteración.
	¿El jugador entiende como se juega?	
	¿El tutorial es de duración aceptable?	
RC03	¿El jugador espera mucho para comenzar partida?	No.
	¿El juego requiere tiempos de carga durante la partida?	No.
RC04	¿Se cumplen las características del estilo artístico?	El estilo visual es simple y burdo para permitir el testing del juego. Por lo tanto no cumple las expectativas del GDD.

Cuadro 6.5: Control de la calidad Iteración 1

¹ 6.2.5 Control de riesgos

- ² En este apartado se establecerán los planes de respuestas aplicados para cada riesgo
³ que haya aparecido durante la ejecución de la primera iteración.

Control de riesgos negativos Iteración 1		
Id	Descripción	Respuesta
IRN01	Desconocimiento de las tecnologías usadas	No se ha dado el caso.
IRN02	Planificación del proyecto errónea	Ha sido necesario realizar una serie de cambios a la planificación, pero ningún ha peligrado la correcta ejecución del proyecto.
IRN03	Control y seguimiento del proyecto erróneo	No procede para esta iteración.
IRN04	Imposibilidad de cerrar el proyecto	No procede para esta iteración.
IRN05	Presentación y exposición no representativos del proyecto	No procede para esta iteración.
IRN06	Incapacidad de seguir el plan de las Iteraciones	No se ha dado el caso.
IRN07	No cumplimiento de las fechas de entrega	No se ha dado el caso.
IRN08	Mal funcionamiento de las tecnologías	No se ha dado el caso.
IRN09	Obligaciones del desarrollador le impiden llevar a cabo el proyecto de forma planificada	No se ha dado el caso.
IRN10	Imposibilidad de satisfacer el GDD	El aspecto estético aún no ha podido ser satisfecho al tener aún una versión temprana del juego. El funcionamiento del movimiento del nivel se ha desarrollado de una forma distinta a la descrita en el GDD, pero el resultado jugable es el esperado.

Cuadro 6.6: Control de riesgos negativos Iteración 1

Control de riesgos positivos Iteración 1		
Id	Descripción	Respuesta
IRP01	Buen conocimiento de las tecnologías por parte del desarrollador	El desarrollador conocía perfectamente las tecnologías, o ha adquirido los conocimientos necesarios de forma sencilla.
IRP02	Planificación del proyecto permite su correcta realización	La planificación no ha supuesto ningún problema.
IRP03	Cumplimiento de las expectativas de la planificación	Se han cumplido las expectativas.
IRP04	Cumplimiento de las estimaciones temporales	Se realizó una gran sobreestimación, que ha llevado a la imposibilidad de cumplir estas estimaciones.
IRP05	GDD más sencillo de satisfacer que lo esperado	Algunos aspectos resultaron más fáciles de desarrollar de lo esperado, lo que ha supuesto un gran ahorro de tiempo.

Cuadro 6.7: Control de riesgos positivos Iteración 1

6.3 INFORMES DE CAMBIO 1

En esta sección se establecerán los cambios realizados para cada una de las versiones de los documentos. 2 3

6.3.1 Documentación 4

v1.1 5

El paquete de trabajo 1.2.3 "Seguimiento del trabajo invertido" resultó no ser necesario puesto que esto ya se llevaría a cabo durante el control de la gestión del alcance. Por lo tanto, este paquete fue eliminado y se modificaron los apartados afectados. 6 7 8

v1.2 9

Durante la realización del Product Backlog, resultó obvio que las estimaciones temporales realizadas en la Gestión del cronograma se encontraban muy lejos de la realidad. Esto es debido a que durante la realización de estas estimaciones, se calcularon buscando aprovechar todo el tiempo disponible en lugar de buscar entender la duración real de estas estimaciones. 10 11 12 13 14

Sin embargo, al realizar las estimaciones de esta forma y organizar el proyecto basándose en ellas, se aseguró que en el futuro las actividades dispusieran del tiempo necesario para su realización; además de poseer un tiempo extra en caso del surgimiento de errores para su resolución. 15 16 17 18

Por lo tanto, al reducir las estimaciones de tiempo de las actividades para que fuesen más realistas, el tiempo extra para la resolución de fallos solo aumentó. Además, estos cambios permiten plantear en el futuro la posibilidades de añadir más actividades sin poner en peligro al proyecto. En definitiva, estos cambios fueron aplicados y se modificaron todos los apartados de la planificación repercutidos por ello (por ejemplo, la gestión de costes). 19 20 21 22 23 24

Esta versión no se ha encontrado disponible en el repositorio de Github debido a que no fue subida al repositorio antes de realizar nuevos cambios que implicasen una nueva versión. Los cambios sin embargo han sido aplicados correctamente y pueden comenzar a verse a partir de la versión siguiente. 25 26 27 28

v1.3

Al principio de la realización del seguimiento de la iteración se hizo necesaria la adición de los paquetes de trabajo del Product Backlog y Sprint Backlog en la EDT, así como la modificación de todos los apartados afectados por estos cambios. Esto es debido a que las actividades que conllevan son requeridas por la aplicación de las metodología Scrum, y a pesar de ser llevadas a cabo, no se encontraban reflejadas en la planificación.

v1.4

Durante la realización de las primeras actividades de la iteración, se hizo evidente que se necesitaba una rama que permitiese alojar las versiones intermedias del juego, para así evitar corrompir las rama principal del repositorio con errores si surgiesen. También resultó necesario cambiar la política de versionado debido a que no resultaba cómoda e intuitiva para versionar este mismo documento.

A raíz de ello, se modificó la Gestión de la configuración de la planificación inicial para adecuarse a estos cambios.

6.3.2 GDD**v4**

La forma en la que se ha programado la generación de niveles ha variado respecto a la descrita en el GDD. Puesto que la nueva forma es más sencilla de desarrollar y permite más fácilmente otras mecánicas, se ha cambiado la definición del GDD.

PARTE IV

CIERRE DEL PROYECTO

MANUAL DE USUARIO

1

2 *The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck*
3 *of his whole damn life – and one is as good as the other.*

4

Ernest Hemingway (1899–1961),

5

Novelist

6

7

8

R esumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo?

7.1 SECCIÓN LIBRE	1
Estructurar en función del proyecto.	2

CONCLUSIONES

The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck of his whole damn life – and one is as good as the other.

*Ernest Hemingway (1899–1961),
Novelist*

*R*esumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo?

8.1	INFORME POST-MORTEM	1
	Qué es un informe post-mortem	2
8.1.1	Lo que ha ido bien	3
	■ Argumento a favor 1.	4
	■ Argumento a favor 2.	5
	■ Argumento a favor 3.	6
8.1.2	Lo que ha ido mal	7
	■ Argumento en contra 1.	8
	■ Argumento en contra 2.	9
	■ Argumento en contra 3.	10
8.1.3	Discusión	11
	En función de lo anterior, qué cambiaría si empezara hoy el proyecto de nuevo.	12
8.2	TRABAJOS FUTUROS	13
	Enumera los puntos abiertos y que no se han resuelto. Indica si darían lugar a otro proyecto y de qué forma se podría acotar.	14
		15

PARTE V

APPENDICES

SOFTWARE PRODUCT LINES

The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck of his whole damn life – and one is as good as the other.

*Ernest Hemingway (1899–1961),
Novelist*

This is an example of an abstract. Multiple lines are supported. Several paragraphs. It jumps to the next page. Blau blau blau. I am introducing more text to reach the third line

A.1 SOFTWARE PRODUCT LINES

- Objective of a *Product Line (PL)* (mass production and customisation) [1]
- The focus in software derives in *Software Product Lines (SPLs)*.
- Variability management: variability models
- When and how are used VMs: FMs are described in FODA report as a key element in SPL since they represent the variability and commonality of the different products in a SPL.

A.2 FEATURE MODELS

To Abductive Section in 2.1

As the number of products to be built by a SPL may be large and the constraints among features may be complex, representing such an information in a manageable and compact manner is a must. *Feature Models (FMs)* represent the set of products a SPL may build in terms of product features. Some features are optional while others are mandatory. To indicate the relationships among features, they are hierarchically linked, forming a tree whose root is a feature representing the whole functionality of a product. The root feature is refined in child features, which increase the level of detail and reduce the scope of features. Recursively following this refinement process, a tree-like structure is obtained where three basic kinds of hierarchical relationships are used:

- **Mandatory:** a mandatory relationship affects a parent and child feature. It forces the child feature to appear in a product whenever its parent feature does.
- **Optional:** a child feature connected to a parent feature by means of an optional relationship may be optionally selected whenever its parent feature is.
- **Set-relationships:** three or more features are part of a set-relationship: a parent feature and a set of two or more child features. A set-relationship contains a cardinality that constraints the number of child features to be selected in a product whenever its parent feature is selected. If the cardinality is $[1,1]$ it is commonly remarked as an *alternative relationship* where only one child feature may be selected at the same time. If the cardinality is $[1..N]$ (where N is the number of

child features), it is also known as an *or-relationship* as any combination of child features is allowed while at least one is selected.

Although FMs can represent most of the most frequent constraints, the hierarchical nature of these models might hinder the representation of some constraints. Under this circumstance, *cross-tree constraints* can be added. The most common kinds of cross-tree constraints are:

- Dependency: a feature depends on another feature if the second one must be part of a product whenever first one is selected.
- Exclusion: two features exclude themselves if both of them cannot be part of a product at the same time.

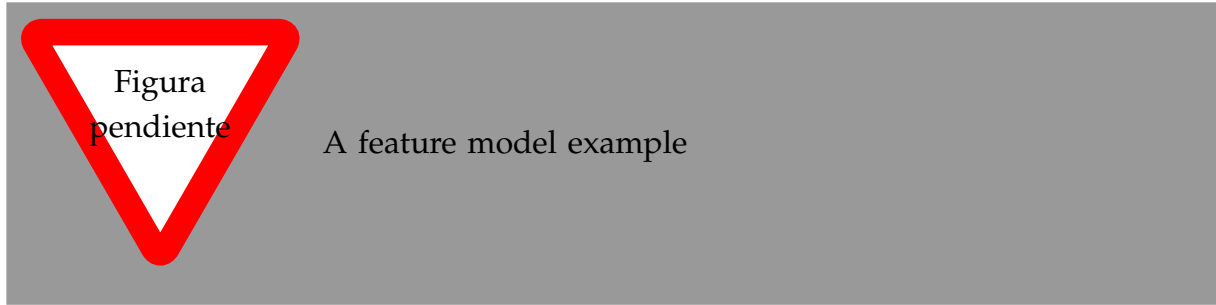


Figura A.1: An example of a Home Integration System

The example in Figure §A.1 describes a *Home Integration System* (HIS) SPL in terms of its features and the relationships among them. Leaning on this example we define some useful terms:

Partial configuration : a partial configuration is a composed by three sets of selected (S), removed(R) and undecided(U) features. A feature can only be in one of these sets and every feature in the FM (fm) must be in one of them, i.e. $S \cup R \cup U = fm$ and $S \cap R \cap U = \emptyset$. A partial configuration represents an intermediate state during the process of a customer selecting the feature for a custom product. For example, $S_P = \{\dots\}$, $R_P = \{\dots\}$ and $U_P = \{\dots\}$ define a partial configuration for the sample FM where some features are still to be decided if they are to be selector or removed in a configuration.

(Full) configuration : a full configuration or simply a configuration is a partial configuration such that the set of undecided features in empty. For example, $S_F = \{\dots\}$ and $R_F = \{\dots\}$ describe a full configuration for the example FM.

Product : a product is a representation for a full configuration such that only the selected features are remarked. For instance, $P = \{\}$ is a product for the above full configuration. A product such as A,B is a valid since all the constraints within the FM are satisfied. However, A,B and C is not a valid product since D is required.

Validation A partial configuration is *valid* if all the relationships and constraints are satisfied given the sets of selected, removed and undecided features. So the definition applies for valid full configurations and valid products. As a conclusion we can affirm that a FM represents all the valid products in a SPL.

Objetivo: Briefly expose attributes as an important asset in feature models.

It is frequent that features are not enough to represent information that is relevant to represent a SPL variability. In this case, FMs are extended with feature attributes such as cost, versions, RAM consumption, etc. in the so-called *Extended Feature Models (EFMs)* [1]. Besides relationships, an EFM contains constraints that affect attributes which reduce even more the set of products a FM describes. Above definitions remain when attributes are introduced into FMs.

A.3 AUTOMATED ANALYSIS OF FEATURE MODELS

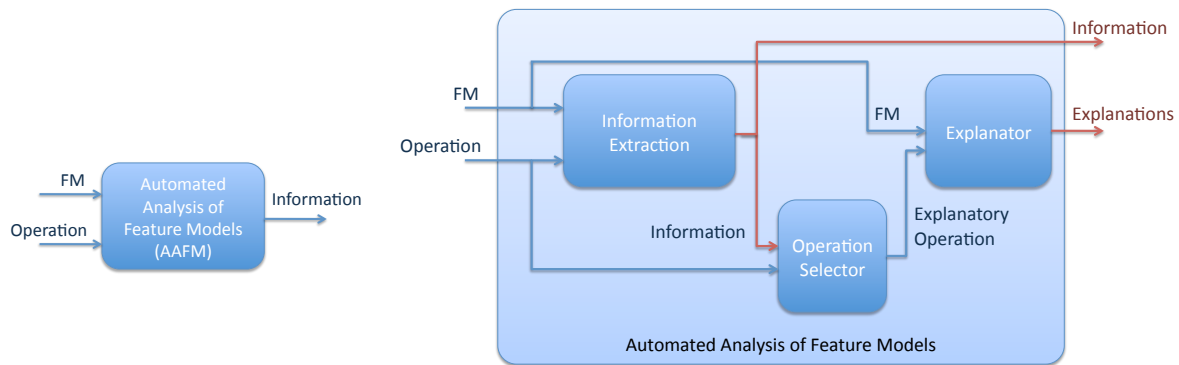
A.3.1 Scope

To Abductive Intro

FMs are used all along the SPL development as key models and many of the development decisions are taken relying on the information contained within them. Most of the times, relationships are complex and hinder the manual extraction of information. Manually obtaining information such as 'which is the product that costs the less?', 'does the feature model contain errors?' or 'why there exist no product containing certain features?' can be an unfeasible task. The complexity and compactness of FMs justify the need of an automated support of these operations. So the *Automated Analysis of Feature Models (AAFM)* arises as a topic of interest to deal with this problem in the SPL community.

The AAFM can be seen as a black-box process that receives a FM and an operation as inputs and obtains information (its kind depends on the analysis operation) as an

output (Fig. A.2(a)). There are many operations that extract information from a FM such as 'counting products' operation whose result is a natural number indicating the number of customised products that can be built; or 'list of products' operation that obtains each of those products. This vision of AAFM as a black-box is valid for a subset of analysis operations that we call *information extraction operations* (IEO) that can be seen as processes to extract information from FMs. In other words, an IEO makes explicit an implicit information within a FM.



(a) The AAFM seen as a black-box process

(b) Extending the AAFM process with explanations

Figura A.2: A different view on AAFM distinguishing between information extraction and explanatory operations

Use me to explain in a larger text than 'side-text' anything that is important to a reader not familiar with the dissertation context for example.

However, there is a subset of analysis operations known as *explanatory operations* (EO) whose objective is explaining the result obtained from a IEO. Sometimes, the result is not the expected one and the analyser needs to know which are the relationships that have caused it. For example, let us suppose that the IEO 'which are the products described in a FM that cost less than \$1000?' obtains no products as a result. If we were expecting to obtain at least one product, it is important to determine the relationships in the FM that are responsible of that behaviour, so an EO 'why there is no product costing less than \$1000?' will shed light on the relationships that avoid obtaining any product. Obtaining no result is not the only case that claims for explanations.

If we obtained only one product as a result and we were expecting to obtain at least 10 products, although an answer is obtained the result is unexpected and the discrepancy reasons have to be found. Moreover, explanatory operations are also use-

ful even when an expected result is obtained, to reinforce the certainty that the result is correct. So it can be concluded that EOs complement the information an FM analyser obtains from IEOs.

The complexity of feature modelling relies on correctly setting the relationships that describe the set of products to be built in a SPL. Relationships are the only elements responsible of the results obtained in FM analysis. So an *explanation* is a set of relationships that may have caused that result. While IEO provides for an unique response that is known for certain, an EO provides for a set of probable explanations to a result obtained from a IEO, being only one of them a valid explanation. It would be the analyser the one in charge of discriminating the correct explanation, maybe performing new analysis operations.

THIS IS A SIDE TEXT. USE TO
REMARK IMPORTANT
INFORMATION

Therefore, two kinds of operations are distinguished in AAFM: information extraction and explanatory operations. Explanatory operations have no sense without a paired information extraction operation and its result. To ensure that explanatory operations are always paired to an information extraction operation, we define a new black-box process of AAFM that incorporates explanations as an additional output (see Figure A.2(b))

1. Information extraction: the original process, which remains the same.
2. Operation selector: depending on the information extraction operation the analyser asks for and the information obtained as a result, this process provides the explanatory operation to be performed. In other words, it pairs an explanatory operation to an information extraction operation.
3. Explanatory analysis: provides a set of explanations from the FM and the explanatory operation.

The overall process can be encapsulated into a holistic black-box process which receives the FM and the information extraction operation as inputs and provides a result and explanations as outputs. It can be seen as we just add explanations as an output to the analysis process.

To realise this view on the AAFM, we need to give details on the insides of these black-boxes. Since the information extraction process is already rigourously defined in

Benavides' PhD dissertation, the purpose of this paper is defining the remaining two sub-processes. We formalise the explanatory analysis process by means of default logic and provide the criteria to implement the operation selector process.

Most Common Techniques to perform AAFM Operations.

A.4 DYNAMIC SOFTWARE PRODUCT LINES (DSPL)

What is a *Dynamic Software Product Line (DSPL)*. Different points of view. What is important is the automation of reconfiguration properties relying on SPL techniques.

We focus in the application of explanations in DSPLs as an application of our results. Specifically we have worked in MAS and smart homes providing a solution for automating product reconfiguration.

A.5 HYPOTHESIS AND OBJECTIVES

Objetivo: Justifying that explanations are a particular set of operations in AAFM that are not solvable by means of the techniques that are used up-to-date

Objetivo: Set an impacting phrase that summarises the hypothesis

Hypothesis

*Explanations cannot be solved by AI techniques used to solve AAFM.
There should exist other AI techniques to solve explanations.*

Objective of the dissertation

Defining a framework to provide solutions for explanatory analysis in FMs.

This dissertation summarises our contribution to solve some of the objectives we set in our PhD project.

- Defining a catalog of analysis operations where explanations are applied.
- Rigorously defining these operations in terms of logics.
- Proposing solutions to these operations.
- Validating our results by means of tools and projects where they are applied.

Next chapter focuses on refining how we have contributed to deal with the above objectives.

A piece of code...

```

public Map<Cardinality, CardinalValue> detectWrongCardinals() {
    // any other implementation of Map can be used instead.
    Map<Cardinality, CardinalValue> result =
        new TreeMap<Cardinality, CardinalValue>();
    for( r : relationships) {
        if (r instanceof Set) {
            Set set = (Set)r;
            Cardinality card = set.getCardinality();
            Domain dom = card.getDomain();
            for (value: dom.getValues())
                if (isWrongCardinal(card, value))
                    result.put(card, value);
        }
    }
    return result;
}

```

A coolTable. Use inside a table.

Use \TableSubtitle{n,title} to add a subtitle as the header. n is the number of columns and title is the text to place. [1]

A Catalog of FM Explanatory Operations (2009 version)		
Information Extraction Operation	FM Explanatory Operations	
	<i>Why? operation</i>	<i>Why not? operation</i>
Valid FM	-	invalid FM
Valid Configuration	valid partial conf.	invalid partial conf.
Valid Product	valid product	invalid product
Products Listing	vaild Product/Config	invalid FM/Product/Config
Products Counting	vaild Product/Config	invalid FM/Product/Config
Optimisation	vaild Product/Config	invalid FM/Product/Config
Core feature	core feature	core feature
Variant feature	variant feature	variant feature
Dead feature detection	-	dead feature
False-optional feature detection	-	false-optional feature
Wrong-cardinality detection	-	wrong cardinal
Information Extraction Operation	Configuration Explanatory Operations	
	<i>Why? operation</i>	<i>Why not? operation</i>
Valid Configuration	valid partial conf.	invalid partial conf.

Cuadro A.1: Most frequently used explanatory operations and their corresponding information extraction operations

SIGLAS

DSPL Dynamic Software Product Line.

EFM Extended Feature Model.

FM Feature Model.

PL Product Line.

SPL Software Product Line.

BIBLIOGRAFÍA

- [1] D. Benavides, A. Ruiz-Cortés, and P. Trinidad. Automated reasoning on feature models. *LNCS, Advanced Information Systems Engineering: 17th International Conference, CAiSE 2005*, 3520:491–503, 2005. ISSN 0302-9743. (pages 92, 94 y 98).