



GRADO EN INGENIERÍA INFORMÁTICA DE GESTIÓN Y SISTEMAS DE INFORMACIÓN

TRABAJO FIN DE GRADO

2015 / 2016

Control the Qí!

Memoria del proyecto

DATOS DE LA ALUMNA O DEL ALUMNO

NOMBRE: **JONATAN**
APELLIDOS: **PÉREZ LÓPEZ**

FDO:

FECHA: **13-06-2016**

DATOS DEL DIRECTOR O DE LA DIRECTORA

NOMBRE: **JUAN ANTONIO**
APELLIDOS: **PEREIRA VARELA**
DEPARTAMENTO: **LENGUAJES Y SISTEMAS INFORMÁTICOS**

FDO:

FECHA: **13-06-2016**

Índice

1	Introducción	9
1.1	Origen del proyecto	10
1.2	Motivaciones para la elección del proyecto	10
2	Planteamiento inicial	11
2.1	Descripción	11
2.2	Objetivos	11
2.3	¿Qué es la realidad aumentada?	12
2.4	Definiciones, acrónimos y abreviaturas	13
2.5	Herramientas	14
2.6	Arquitectura	16
2.6.1	Arquitectura interna del juego	16
2.6.2	Arquitectura externa del juego	18
2.7	Alcance	18
2.7.1	Organización	20
2.7.2	Aprendizaje	21
2.7.3	Análisis y diseño	22
2.7.4	Implementación y desarrollo	23
2.7.5	Pruebas	25
2.7.6	Documentación	26
2.8	Planificación temporal	30
2.9	Evaluación económica	32
2.9.1	Mano de obra	32
2.9.2	Gasto de software	32
2.9.3	Gasto de hardware	33
2.9.4	Gastos indirectos	34
2.9.5	Gastos totales	35
2.9.6	Posibilidades de negocio	35
2.10	Riesgos	37
2.10.1	Problemas con eduroam	38
2.10.2	Fallos en el ordenador	39
2.10.3	Problemas con C#, Unity, Vuforia, Blender o GIMP	40
2.10.4	Falta de inspiración	41
3	Antecedentes	43
3.1	Aprende: números y letras	43
3.2	Simon	44
4	Captura de requisitos	45

4.1	Jerarquía de actores	45
4.2	Casos de uso	45
4.3	Modelo de dominio	46
5	Elección de lenguajes y tecnologías	49
5.1	Unity y el lenguaje de programación C#	49
5.2	SDK de Vuforia	49
5.3	Blender	50
5.4	GIMP 2	50
6	Análisis y diseño	51
6.1	Análisis de los elementos del juego	51
6.1.1	Escenas	51
6.1.2	<i>Ranking</i>	51
6.1.3	<i>Scripts: Start(), Awake(), Update()</i>	52
6.1.4	Objetos virtuales	52
6.1.5	Animaciones	53
6.1.6	Imagen objetivo y botones virtuales	53
6.2	Diagrama de estados	54
6.3	Diagrama de clases	56
6.4	Diagramas de secuencia	58
7	Desarrollo	61
7.1	Desarrollo inicial de la aplicación	61
7.2	Desarrollo final de la aplicación	62
8	Pruebas	67
8.1	Pruebas del menú	67
8.2	Pruebas del juego	69
8.3	Pruebas del <i>ranking</i>	72
8.4	Pruebas de dispositivos	73
9	Conclusiones	75
9.1	Análisis entre planificación estimada y real	75
9.1.1	Cambios respecto al alcance	75
9.1.2	Reevaluación económica	77
9.1.3	Rediseño de los diagramas de planificación	77
9.2	Lineas futuras para Control the Qí!	81
9.3	Licencias	82
9.3.1	Recursos utilizados	82
9.3.2	Control the Qí!	82

9.4 Reflexión personal	83
Bibliografía	85
A Anexo 1: Manual de usuario	87
A.1 Requisitos mínimos	87
A.2 Pantallas de Control the Qí!	88
A.3 Componentes y temática del juego	90
A.4 Cómo jugar una partida	94
A.5 Posición del cuerpo con respecto al tablero	97
B Anexo 2: Gafas de realidad aumentada: BilboFranky	101
B.1 Materiales	101
B.2 Instrucciones	102

Índice de figuras

1	Imagen de la aplicación <i>Streetmuseum</i>	12
2	Arquitectura interna del juego	16
3	Arquitectura externa del juego	18
4	Diagrama EDT por bloques.	19
5	Diagrama EDT por tareas.	19
6	Diagrama Gantt	31
7	Imagen del juego Simon	44
8	Jerarquía de actores	45
9	Diagrama de casos de uso	46
10	Modelo de dominio	47
11	Puntos referencia en una imagen objetivo	54
12	Diagrama de estados	54
13	Diagrama de clases	57
14	Diagrama de secuencia de la partida (1)	59
15	Diagrama de secuencia de la partida (2)	60
16	Diseño inicial del menú	61
17	Diseño inicial del juego	62
18	Diseño final del menú	62
19	Tablero final de Control the Qí!	63
20	Puntos referencia del tablero	63
21	Localización botones virtuales en el tablero	64
22	Imagen de Control the Qí!	65
23	Diagrama final EDT por tareas	78
24	Comparativa horas estimadas y reales	79
25	Diagrama Gantt final	80

Índice de tablas

1	Dependencias y duración de las tareas (1).	29
2	Dependencias y duración de las tareas (2).	29
3	Costes totales de la aplicación	35

1. Introducción

Antes de nada, me gustaría que pienses un momento en la siguiente pregunta: ¿qué es para ti un juego?. Me imagino que no contestaría lo mismo un niño que una persona en plena madurez. El niño te respondería seguramente haciendo alusión a los videojuegos y la persona adulta te nombraría algún juego de mesa con el que haya pasado buenos ratos con los amigos. Es difícil que haya una respuesta unánime, y más cuando la RAE¹ tiene 19 definiciones para la palabra juego sin contar sus locuciones.

La diferencia entre cómo juegan los niños actualmente y cómo lo hacían hace 50 años es abismal. Antes, los chavales echaban partidos de fútbol en la calle, ahora, también siguen echando partidos, pero meten los goles con los botones del mando de la PlayStation 4², jugando cada uno desde sus casas.

Si nos centramos en los videojuegos, observamos que han experimentado una gran evolución gracias a los constantes avances tecnológicos que han ido surgiendo en los últimos años. No tiene sentido comparar la calidad de imagen, procesador, memoria... de las aplicaciones y consolas de finales del siglo XX a las actuales.

Una revolución surgió con la aparición de la consola Wii³ en 2006 por parte de Nintendo, dado que otorgó una nueva forma para jugar a videojuegos, realizando movimientos de manos y brazos mediante un mando inalámbrico de infrarrojos, unido por una correa de seguridad a la muñeca del usuario. En el breve plazo de 10 años, se ha avanzado de tal forma, que ahora se utilizan sensores de voz y cámaras de alta resolución para captar al detalle cualquier actividad que realice el usuario, como Kinect de Xbox One⁴.

En un mundo de comida rápida, sobrepeso y tecnológicamente alienado se hace indispensable pensar en aplicaciones que fomenten la actividad motriz de las personas para evitar tener una sociedad propensa a enfermedades cardiovasculares.

El objetivo de este Trabajo de Fin de Grado (TFG) es crear un juego de realidad aumentada que sirva para ejercitar manos, brazos y hombros. El usuario deberá disponer de unas gafas adaptadas para insertar su dispositivo móvil y de esa forma poder tener libertad en la manos. En el Anexo B, aparece una guía para crear unas gafas de realidad aumentada personalizadas.

¹Definición juego en RAE: <http://dle.rae.es/?id=MaS6XPk>

²Web oficial de PlayStation 4: <https://www.playstation.com/es-es/explore/ps4/>

³Web Oficial de Wii: <http://wii.com/>

⁴Página de Kinect: <http://www.xbox.com/es-ES/xbox-one/accessories/kinect-for-xbox-one>

1.1. Origen del proyecto

En un principio, mi intención era realizar el TFG sobre realidad virtual en Holanda, dado que me habían concedido la beca Erasmus+ para ir a Eindhoven. A falta de unos meses, me dijeron que finalmente no iba a poder realizar dicha especialización, y sumado a otros problemas surgidos, al final tuve que renunciar a la plaza y quedarme en Bilbao para acabar el grado.

Es por ello, que se me quedó esa espina clavada respecto a la realidad virtual. Empecé a buscar información y fue entonces cuando descubrí la realidad aumentada, la cual fusiona el mundo virtual con el mundo real.

Unos compañeros de clase estaban haciendo prácticas relacionadas con la realidad aumentada dentro de la propia facultad, con la empresa Creativitic, por lo que me surgió la posibilidad de realizar mi TFG con ellos y así, poder contar con su asesoría en una disciplina en la que yo no había tenido contacto previo.

1.2. Motivaciones para la elección del proyecto

Para hablar de las motivaciones que impulsaron la elección de este proyecto, tenemos que echar la vista atrás unos 7 años, hasta llegar a mi primer año de Bachiller científico-tecnológico en el colegio Cooperativa Basauri. Durante esos años, mis asignaturas preferidas fueron Matemáticas, Dibujo Técnico y Diseño Asistido por Ordenador.

De la primera asignatura me han nutrido muy bien de información en el grado, pero he echado en falta durante estos años universitarios asignaturas de modelados de objetos 3D o similares, para su posterior incorporación a juegos o aplicaciones.

Mi primera intención era acabar el grado mediante un Erasmus orientado a la realidad virtual, pero la cancelación del mismo hizo que me planteara realizar el TFG en Bilbao manteniendo dicha especialidad.

Una vez empecé a indagar, descubrí la realidad aumentada, y ví el amplio abanico de posibilidades que ofrecía con respecto a la realidad virtual, que era la idea inicial en un primer momento.

Me cautivó la utilización de la cámara del móvil para proyectar en la pantalla objetos en 3D sobre el mundo real, o la posible utilización de las manos para tocar botones colocados virtualmente. Esto, sumado a mi gran afición por el deporte, hizo que me decantara por realizar un juego de realidad aumentada que fomentara la actividad física.

2. Planteamiento inicial

En esta sección se presenta la descripción, los objetivos, el alcance y las herramientas del proyecto, así como la evaluación económica y de riesgos.

2.1. Descripción

Se pretende desarrollar un juego de realidad aumentada para dispositivos móviles que fomente el ejercicio físico. El juego dispondrá de tres niveles de dificultad y se podrá jugar en un tablero predeterminado. También tendrá un *ranking* de mejores puntuaciones conseguidas por el usuario en su dispositivo.

2.2. Objetivos

El objetivo principal es la creación de la propia aplicación de realidad aumentada para fomentar la actividad física en forma de juego. A partir de este objetivo, surgen otros objetivos secundarios los cuales se exponen a continuación:

- **Tipo de juego y franja de edad:** dado que es un juego de actividad deportiva, se tendrá en cuenta la dificultad y velocidad de los movimientos a la hora de crear la aplicación, para que lo puedan usar desde niños, hasta personas mayores. Se buscará una temática simple y visual para el juego.
- **Interfaz clara e intuitiva:** la forma de moverse a través de la aplicación debe ser lo más clara posible para todo tipo de usuarios.
- **Diseño y creación de tablero:** el juego llevará consigo un tablero acorde con la temática que tenga. Se crearán dos medidas para este tablero, una tamaño A4(297x210 mm) y otra tamaño A2(594x420 mm)⁵, así como un soporte para colgarlo en la pared.
- **Gafas de realidad aumentada:** se crearán, a partir de materiales reciclados, unas gafas de realidad aumentada para dispositivos móviles. Con ellas, el usuario podrá tener las manos libres mientras juega con la aplicación.

⁵Formatos de papel : <http://ladinamo.com/blog/entrada/medidas-de-paginas-a0-a1-a2-a3-a4-a5-a6-a7-y-a8>

2.3. ¿Qué es la realidad aumentada?

Se trata de una buena pregunta, y más, dado el auge que está generando esta tecnología en los últimos meses. Una definición simple sería la fusión del mundo real con el mundo virtual. Pero, para ilustrarlo mejor, se me ocurre hacer alusión a una aplicación real que se está utilizando actualmente por las calles de Londres.

Se trata del *Streetmuseum* [2], creado por el *Museum of London*. "*Streetmuseum gives you a unique perspective of old and new London whether you are discovering the capital for the first time or revisiting favourite haunts. Hundreds of images from the Museum of London's extensive collections showcase both everyday and momentous occasions in London's history, from the Great Fire of 1666 to the swinging sixties*".

En la Figura 1⁶ se observa cómo se combina la imagen a tiempo real con una fotografía antigua de la calle *Covent Garden* de Londres.



Figura 1: Imagen de la aplicación *Streetmuseum*

Esta aplicación tiene guardadas una serie de imágenes antiguas de la ciudad de Londres. Utilizando el GPS del dispositivo móvil, localiza dónde se encuentra el usuario, le guía a través de la ciudad de Londres y superpone esas imágenes sobre las calles que está enfocando la cámara del dispositivo en tiempo real.

Este es uno de los tipos de realidad aumentada que podemos encontrar. En el curso de formación FOPU [1], impartido al profesorado de la Universidad del País Vasco (UPV/EHU), nos hablan de los 5 tipos distintos de realidad aumentada que existen:

⁶Más imágenes en: <http://blog.museumoflondon.org.uk/streetmuseum-2-0/>

1. **Coordenadas GPS:** visto en el ejemplo anterior, utiliza las coordenadas GPS, brújula y acelerómetro para localizar y rastrear la posición exacta del usuario, incluyendo el punto cardinal al que está mirando. De esta forma, dependiendo hacia dónde esté mirando el usuario, podrá ver unos objetos virtuales u otros desde la misma posición.
2. **Marcadores en blanco y negro:** se utilizan imágenes muy simples similares a códigos QR⁷ en blanco y negro, para proyectar objetos virtuales sobre ellos.
3. **Marcadores de imagen:** utiliza la cámara del dispositivo móvil para localizar imágenes o marcadores predefinidos y entonces, proyectar sobre ellos objetos virtuales.
4. **Marcadores en tiempo real:** el propio usuario crea y define sus propias imágenes con la cámara del móvil para proyectar cualquier objeto virtual sobre ellas.
5. **Reconocimiento facial:** capta mediante cámaras los movimientos realizados por el usuario con su rostro. Cada expresión del usuario, va asociada a una determinada acción o movimiento sobre la aplicación.

2.4. Definiciones, acrónimos y abreviaturas

A continuación, se enumeran los términos relacionados con la aplicación que se va a realizar.

- **Android:** sistema operativo para móviles y *tablets* creado por Google. La última versión estable es la 6.0.1 y fue lanzada el 3 de Mayo de 2016.
- **GPS:** siglas de *Global Positioning System* o Sistema de Posicionamiento Global. Sistema usado mediante satélites que permite localizar o rastrear a persona o objetos por toda la Tierra.
- **Juego *survival*:** juego que consiste en aguantar el máximo tiempo posible realizando alguna acción determinada.
- **App:** abreviatura de aplicación.
- **3D:** abreviatura de 3 Dimensiones. Los objetos son tridimensionales si tienen longitud, anchura y profundidad representado por los ejes x,y,z.

⁷Página oficial de QR: <http://www.qrcode.com/>

- **SDK**: siglas de *Software Development Kit* (Kit de Desarrollo de Software). En este proyecto se utilizara el SDK de Vuforia en Unity para añadir las funcionalidad de realidad aumentada.
- **GameObject**: contenedor que se utiliza en Unity para guardar y manipular objetos, animaciones, luces... que se vayan a utilizar en la aplicación.
- **Prefabs**: son *GameObjects* reutilizables con una funcionalidad específica. En el caso de este proyecto se usarán los Prefabs procedentes del SDK de Vuforia.
- **Drag&Drop**: es un mecanismo con el cual se permite mover un archivo a otra carpeta o sección con el botón derecho del ratón.
- **PlayerPrefs**: clase procedente de las librerías de Unity la cual permite almacenar valores indefinidamente dentro de la aplicación. Con ella se guardarán las mejores puntuaciones.
- **Imagen objetivo**: nombre por el cual se conoce a la imagen que utiliza el programa como tablero.
- **Eduroam**: red WiFi utilizada por la Universidad del País Vasco (UPV/EHU)

2.5. Herramientas

En este apartado, se enumeran las diferentes herramientas que se utilizarán para el desarrollo del proyecto.

- **Unity v5.3.1**: motor gráfico multiplataforma de creación de videojuegos y aplicaciones. Permite una fácil integración con el SDK de Vuforia.
- **MonoDevelop v5.9.1**: es el entorno de desarrollo que utiliza Unity por defecto. Está especialmente diseñado para lenguajes como C#, JavaScript o Python. Será usado para programar el código del juego en lenguaje C#.
- **Blender v2.76**: programa de código abierto que sirve para el modelado de objetos en 3D. Con él, se generarán todas las figuras que aparecerán contenidas en el juego.
- **LibreOffice**: herramienta de código abierto de edición de textos, presentaciones, hojas de cálculo... Se utilizará para realizar algunas tablas de la memoria.

- **Vuforia v5.5.9:** SDK que permite la creación de contenidos de Realidad Aumentada en Unity.
- **GIMP 2:** programa de código abierto de edición de imágenes. Se utilizará para diseñar el tablero de la aplicación así como el logotipo de la misma.
- **pixabay.com:** página web con imágenes gratuitas de alta calidad. Se escogerán imágenes para diseñar el tablero de la aplicación así como el logotipo de la misma.
- **Dropbox:** espacio de almacenamiento de archivos en un servidor web. Se utilizará para guardar copias de seguridad de la aplicación, documentación y material utilizado.
- **Google Drive:** espacio de almacenamiento en la nube al igual que Dropbox pero permite crear y editar documentos presentaciones... desde el navegador. Se utilizará para la generación de la defensa y como almacenamiento alternativo a Dropbox.
- **TeXstudio:** programa para la edición de textos de carácter profesional mediante código LaTeX. Se usará para la realización de la memoria del proyecto.
- **Cacoo.com:** herramienta online para la realización y gestión de diagramas, prototipos, diseños, esquemas... Se utilizará en este trabajo para generar los diagramas de clases y secuencia correspondientes a la aplicación.
- **GanttProject:** herramienta para la creación de diagramas de planificación temporal. Se creará con ella el diagrama Gantt perteneciente a este proyecto.

2.6. Arquitectura

Para analizar la arquitectura del juego, de una forma simple e intuitiva, se va a dividir esta sección en dos apartados independientes, en el primero, se hablará de la arquitectura interna del juego, y en el segundo, de la arquitectura necesaria para que un usuario pueda jugar con él.

2.6.1. Arquitectura interna del juego

En la Figura 2, aparece reflejada la arquitectura interna de la aplicación, la cual viene detallada a continuación:

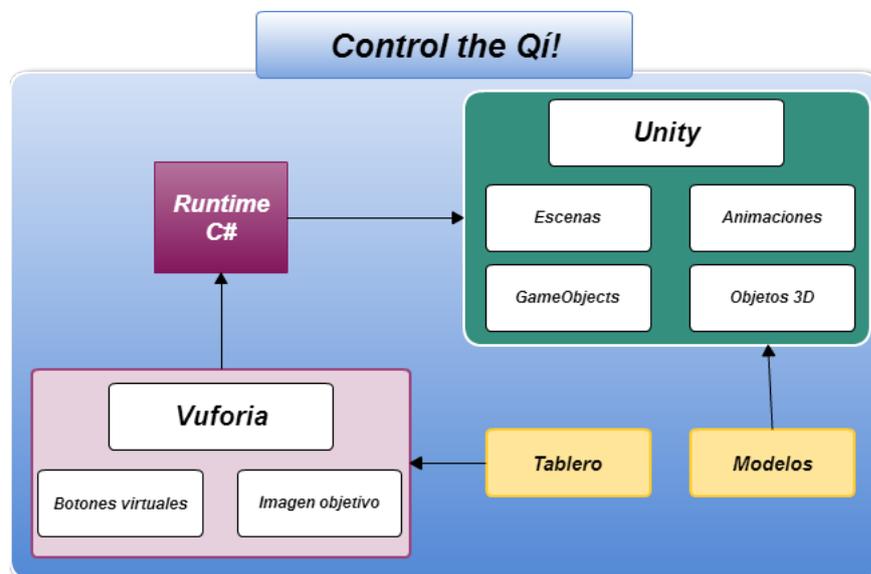


Figura 2: Arquitectura interna del juego

Tablero

El tablero de la aplicación deberá ser subido a la pagina oficial de Vuforia⁸ (sección *Develop ->TargetManager*), previa creación de una cuenta de usuario en la propia página.

Vuforia evalúa la calidad de resolución de las imágenes sobre una calificación de 5 estrellas. Una vez que la imagen sea evaluada, se podrá descargar en un formato compatible con Unity, para su posterior incorporación al proyecto.

⁸Página de Vuforia: <https://developer.vuforia.com/>

Vuforia

Lo primero es incorporar a Unity el SDK de Vuforia. En él, se encuentran las librerías necesarias para introducir la realidad aumentada a una aplicación. También dispone de un número amplio de *Prefabs*, con funcionalidades como: reconocimiento de objetos, marcadores, imágenes objetivos... pero para este proyecto se utilizarán únicamente los botones virtuales y la imagen objetivo.

Runtime C#

Se ha elegido el lenguaje de programación C# para la implementación del juego, dado que es el lenguaje estándar de Unity junto con JavaScript y además, porque las librerías del SDK de Vuforia vienen implementadas en ese lenguaje, lo que hace la vida más fácil al desarrollador evitando incompatibilidades entre lenguajes.

Modelos

Los modelos de objetos 3D se generarán desde la herramienta gratuita de modelado llamada Blender, y se importarán a Unity en un formato compatible (.blend), para su posterior incorporación al juego.

Unity

Se encargará de recopilar todos los elementos generados en los pasos anteriores, para fusionarlos en forma de juego. Se utilizará MonoDevelop como entorno de desarrollo y el lenguaje de programación C#. Se crearán los *scripts* del juego y se extraerá el código necesario de las librerías del SDK de Vuforia, para añadirlo a los *scripts* creados, y así, incorporar la realidad aumentada al juego.

2.6.2. Arquitectura externa del juego



Figura 3: Arquitectura externa del juego

Como se puede observar en la Figura 3, la arquitectura necesaria para jugar con la aplicación es muy sencilla. El usuario deberá disponer de un dispositivo Android con una **versión 3.2 (HoneyComb)** o superior y haber instalado la aplicación previamente.

A la hora de jugar, se utilizará la cámara del dispositivo para localizar la imagen del tablero y así, proyectar el juego sobre ella pudiéndose ver tanto el mundo real como el virtual simultáneamente en la pantalla del mismo.

2.7. Alcance

Para definir el alcance de este proyecto se ha dividido la Estructura de Descomposición de Trabajo o EDT en 6 bloques: organización, aprendizaje, análisis y diseño, implementación y desarrollo, pruebas, y por último, documentación.

A continuación se detalla cada bloque:

1. **Organización:** comprende las tareas de planificación y instalación de software, además de la organización del proyecto.
2. **Aprendizaje:** tareas relacionadas con el estudio de las diferentes herramientas a usar en la aplicación.
3. **Análisis y diseño:** en este bloque se realizan tareas relacionadas con el análisis y diseño de la aplicación mediante diagramas de clase, secuencia...
4. **Implementación y desarrollo:** aquí se albergarán todas las tareas relacionadas con la implementación del juego.

5. **Pruebas:** las pruebas necesarias para el correcto funcionamiento del juego.
6. **Documentación:** tareas relacionadas a la creación y edición del DOP, memoria del proyecto y defensa.

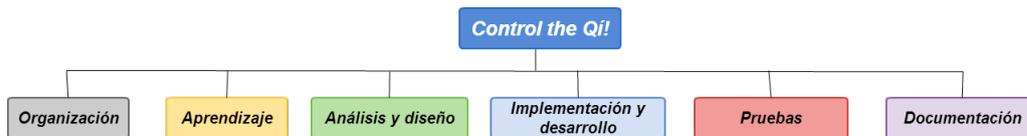


Figura 4: Diagrama EDT por bloques.

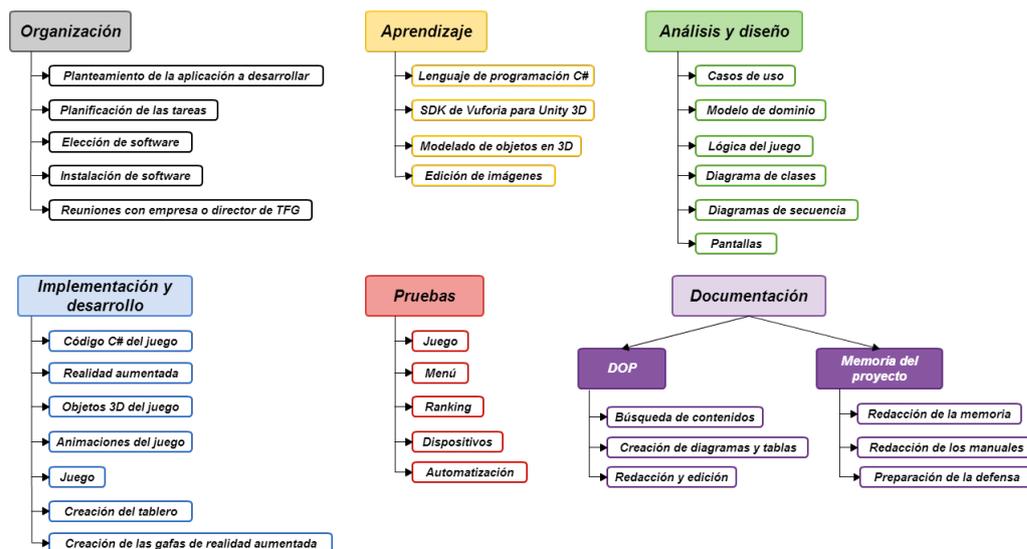


Figura 5: Diagrama EDT por tareas.

En las Figuras 4 y 5, se muestra la división por bloques mencionada previamente, así como la descomposición en tareas de cada bloque.

Todos los bloques son precedentes al siguiente, excepto por una excepción en el bloque de la documentación. Como se puede apreciar, en la Figura 5 se ha subdividido el bloque de documentación en otros dos bloques: DOP y memoria del proyecto. Las tareas relacionadas con el DOP se realizarán en paralelo a la organización y aprendizaje, mientras que las tareas de la memoria se realizarán tras el bloque de pruebas.

2.7.1. Organización

En este primer apartado, se exponen las tareas relacionadas con la organización y puesta a punto de todo lo necesario para la correcta elaboración del proyecto.

<i>Planteamiento de la aplicación a desarrollar</i>
Paquete de trabajo: Organización.
Duración: 15 horas.
Descripción: en esta tarea se realizará una lluvia de ideas para decidir qué tipo de aplicación se va a desarrollar en torno a la realidad aumentada.
Salidas/Entregables: herramientas a utilizar.
Recursos necesarios: papel, lápiz y ordenador con conexión a Internet.
<i>Planificación de las tareas</i>
Paquete de trabajo: Organización.
Duración: 5 horas.
Descripción: planificación de todas las tareas a realizar durante el proyecto, la duración de cada una, predecesores...
Salidas/Entregables: tareas a realizar durante el proyecto.
Recursos necesarios: Cacao.com y hoja de cálculo de LibreOffice.
<i>Elección de software</i>
Paquete de trabajo: Organización.
Duración: 5 horas.
Descripción: elegir qué herramientas se van a utilizar para la elaboración del proyecto. Se escogerá la versión de la herramienta que sea más apropiada, dado que pueden darse incompatibilidades entre versiones.
Salidas/Entregables: herramientas a utilizar.
Recursos necesarios: ordenador de trabajo y conexión a Internet.
<i>Instalación de software</i>
Paquete de trabajo: Organización.
Duración: 5 horas.
Descripción: instalación del software específico necesario para la realización del proyecto.
Salidas/Entregables: correcta instalación del software.
Recursos necesarios: ordenador de trabajo y conexión a Internet.

<i>Reuniones con la empresa o el director del TFG</i>
Paquete de trabajo: Organización.
Duración: 25 horas.
Descripción: reuniones periódicas con el director del TFG o con la empresa, para llevar un control sobre la planificación y realización del proyecto, así como para establecer alguna modificación al mismo si se considera oportuno.
Salidas/Entregables: cambios a realizar en la aplicación o gestión del proyecto.
Recursos necesarios: lugar tranquilo en el que poder reunirse.

2.7.2. Aprendizaje

A continuación, se exponen las tareas centradas en el aprendizaje de los programas que serán utilizados para la realización del proyecto.

<i>Aprendizaje del lenguaje C#</i>
Paquete de trabajo: Aprendizaje.
Duración: 20 horas.
Descripción: estudio de lenguaje C#, su funcionamiento en MonoDevelop y su incorporación a Unity.
Salidas/Entregables: N/A.
Recursos necesarios: Unity, MonoDevelop, foros especializados.

<i>Aprendizaje del SDK de Vuforia</i>
Paquete de trabajo: Aprendizaje.
Duración: 15 horas.
Descripción: estudio y análisis del SDK de Vuforia, su implementación en el lenguaje C# y su funcionamiento en Unity.
Salidas/Entregables: N/A.
Recursos necesarios: Unity, MonoDevelop, SDK de Vuforia y foros especializados

<i>Aprendizaje de modelado de objetos en 3D</i>
Paquete de trabajo: Aprendizaje.
Duración: 10 horas.
Descripción: aprendizaje sobre la creación y modelado de objetos en 3D.
Salidas/Entregables: N/A.
Recursos necesarios: Blender, foros especializados

<i>Aprendizaje de edición de imágenes.</i>
Paquete de trabajo: Aprendizaje.
Duración: 6 horas.
Descripción: aprendizaje sobre la edición de imágenes en GIMP 2.
Salidas/Entregables: N/A.
Recursos necesarios: GIMP 2

2.7.3. Análisis y diseño

En este apartado aparecen las tareas relacionadas a la creación de los diferentes diseños de la aplicación.

<i>Análisis y diseño de casos de uso</i>
Paquete de trabajo: Análisis y diseño.
Duración: 6 horas.
Descripción: realizar los diferentes casos de uso correspondientes a la aplicación.
Salidas/Entregables: casos de uso.
Recursos necesarios: Cacao.com.

<i>Análisis y diseño del modelo de dominio</i>
Paquete de trabajo: Análisis y diseño.
Duración: 6 horas.
Descripción: realizar el modelo de dominio correspondiente a la aplicación.
Salidas/Entregables: modelo de dominio.
Recursos necesarios: Cacao.com.

<i>Análisis y diseño de la lógica del juego</i>
Paquete de trabajo: Análisis y diseño.
Duración: 6 horas.
Descripción: realizar la lógica del juego mediante un diagrama de estados.
Salidas/Entregables: diagrama de estados.
Recursos necesarios: Cacao.com.

<i>Análisis y diseño del diagrama de clases</i>
Paquete de trabajo: Análisis y diseño.
Duración: 6 horas.
Descripción: realizar el diagrama de clases correspondiente a la aplicación teniendo como referencia la lógica implementada.
Salidas/Entregables: diagrama de clases.
Recursos necesarios: Cacao.com.
<i>Análisis y diseño de los diagramas de secuencia</i>
Paquete de trabajo: Análisis y diseño.
Duración: 16 horas.
Descripción: realizar los diagramas de secuencia correspondientes a la aplicación teniendo como referencia la lógica implementada y el diagrama de clases generado .
Salidas/Entregables: diagramas de secuencia.
Recursos necesarios: Cacao.com.
<i>Análisis y diseño de pantallas</i>
Paquete de trabajo: Análisis y diseño.
Duración: 6 horas.
Descripción: diseñar las diferentes pantallas del menú que tendrá el juego.
Salidas/Entregables: pantallas del juego.
Recursos necesarios: Cacao.com o Unity.

2.7.4. Implementación y desarrollo

A continuación, se exponen las diferentes tareas que forman parte de la implementación y desarrollo del proyecto.

<i>Implementación del código C# del juego</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 30 horas.
Descripción: se desarrollará la programación de la aplicación en lenguaje C# siguiendo los diagramas de clases y secuencia realizados previamente.
Salidas/Entregables: código del juego en lenguaje C#.
Recursos necesarios: MonoDevelop.

<i>Implementación de la realidad aumentada</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 20 horas.
Descripción: se implementará la realidad aumentada del juego en lenguaje C#, así como la subida de imágenes al servidor de Vuforia para su evaluación y posterior incorporación al juego.
Salidas/Entregables: código del juego en lenguaje C# y la imagen evaluada desde el servidor de Vuforia.
Recursos necesarios: Unity, MonoDevelop, Vuforia.
<i>Creación de objetos 3D del juego</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 20 horas.
Descripción: se generarán los objetos 3D que se vayan a utilizar en la aplicación.
Salidas/Entregables: objetos 3D de la aplicación.
Recursos necesarios: Blender.
<i>Implementación de las animaciones del juego</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 15 horas.
Descripción: se implementarán las animaciones que vayan a utilizarse en el juego.
Salidas/Entregables: animaciones del juego.
Recursos necesarios: Blender.
<i>Implementación del juego</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 20 horas.
Descripción: en esta tarea, se irán introduciendo las animaciones, los objetos 3D, el código C# a la aplicación y se creará el menú y el <i>ranking</i> .
Salidas/Entregables: aplicación realizada.
Recursos necesarios: Unity, animaciones, modelos y tablero.
<i>Creación del tablero</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 10 horas.
Descripción: se creará el tablero del juego .
Salidas/Entregables: tablero del juego.
Recursos necesarios: GIMP 2.

<i>Creación de las gafas de realidad aumentada</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 15 horas.
Descripción: se crearán las gafas de realidad aumentada para el juego .
Salidas/Entregables: gafas de realidad aumentada.
Recursos necesarios: materiales reciclados, pegamento, pinceles...

2.7.5. Pruebas

En esta sección, se detallan las diferentes pruebas que se ejecutarán una vez realizada la implementación del juego.

<i>Pruebas del juego</i>
Paquete de trabajo: Pruebas.
Duración: 10 horas.
Descripción: realización de las pruebas pertinentes para comprobar el correcto funcionamiento del juego con la imagen objetivo.
Salidas/Entregables: posibles fallos surgidos.
Recursos necesarios: Unity o dispositivo Android.

<i>Pruebas del menú</i>
Paquete de trabajo: Pruebas.
Duración: 6 horas.
Descripción: comprobar el correcto funcionamiento del menú en la aplicación, selección nivel, actualización de parámetros...
Salidas/Entregables: posibles fallos surgidos.
Recursos necesarios: Unity o dispositivo Android.

<i>Pruebas del ranking</i>
Paquete de trabajo: Pruebas.
Duración: 6 horas.
Descripción: comprobar si el <i>ranking</i> se actualiza y se muestra por pantalla correctamente.
Salidas/Entregables: posibles fallos surgidos.
Recursos necesarios: Unity o dispositivo Android.

<i>Pruebas de dispositivos</i>	
Paquete de trabajo:	Pruebas.
Duración:	6 horas.
Descripción:	jugar con la aplicación en diferentes dispositivos Android para comprobar calidad de imagen, velocidad de procesamiento u otros aspectos técnicos.
Salidas/Entregables:	informe de evaluación.
Recursos necesarios:	dispositivos Android.
<i>Pruebas de automatización</i>	
Paquete de trabajo:	Pruebas.
Duración:	16 horas.
Descripción:	realizar pruebas unitarias de cada apartado de la aplicación con el fin de comprobar su robustez y facilitar su mantenimiento.
Salidas/Entregables:	N/A.
Recursos necesarios:	dispositivos Android.

2.7.6. Documentación

Por último, se detallan las tareas relacionadas con la documentación del proyecto. Se ha querido dividir en 2 secciones, una dedicada al documento de objetivos del proyecto, y la otra, a la memoria. De esta forma, se consigue una mejor organización de este paquete de trabajo.

2.7.6.1. Documento de Objetivos del Proyecto

También conocido como DOP, es un documento muy importante a la hora de desarrollar un proyecto, en él, se detallan los objetivos del mismo, así como la planificación temporal, herramientas a utilizar u otros aspectos técnicos.

<i>Búsqueda de contenidos para el DOP</i>	
Paquete de trabajo:	Documentación - DOP.
Duración:	6 horas.
Descripción:	búsqueda de información necesaria para la elaboración del DOP acerca de las herramientas a utilizar y la tecnología a implementar.
Salidas/Entregables:	enlaces de interés.
Recursos necesarios:	ordenador con conexión a Internet, herramientas que se van a utilizar.

<i>Creación de diagramas y tablas del DOP</i>	
Paquete de trabajo: Documentación - DOP.	
Duración: 10 horas.	
Descripción: generar aquellas tablas y diagramas referentes a la planificación temporal u otros aspectos incluidos en el DOP. Los diagramas referidos al diseño de la aplicación no se incluyen en esta tarea.	
Salidas/Entregables: tablas y diagramas del DOP.	
Recursos necesarios: Cacao.com, LibreOffice y GanttProject.	

<i>Redacción y edición del DOP</i>	
Paquete de trabajo: Documentación - DOP.	
Duración: 20 horas.	
Descripción: redacción del documento de objetivos del proyecto, así como la incorporación de los diagramas y tablas elaborados en la anterior tarea.	
Salidas/Entregables: documento de objetivos del proyecto.	
Recursos necesarios: TeXstudio.	

2.7.6.2. Memoria del proyecto

La memoria del proyecto recoge los diseños de la aplicación, los diferentes diagramas creados, la implementación y desarrollo, problemas surgidos durante la realización, las conclusiones y líneas futuras que se pretendan implementar. También se incluyen los manuales necesarios para una correcta utilización de la aplicación.

<i>Redacción de la memoria</i>
Paquete de trabajo: Documentación - Memoria.
Duración: 30 horas.
Descripción: redacción y edición de la memoria del proyecto.
Salidas/Entregables: memoria del proyecto.
Recursos necesarios: Latex, diagramas generados, aplicación desarrollada, herramientas utilizadas.
<i>Redacción de los manuales</i>
Paquete de trabajo: Documentación - Memoria.
Duración: 15 horas.
Descripción: redacción y edición del manual de usuario, además de una guía sobre como realizar las gafas de realidad aumentada.
Salidas/Entregables: manuales de la aplicación.
Recursos necesarios: Latex, imágenes de la aplicación, gafas y tablero.
<i>Preparación de la defensa</i>
Paquete de trabajo: Documentación - Memoria.
Duración: 30 horas.
Descripción: preparación de la defensa del proyecto para su posterior presentación.
Salidas/Entregables: diapositivas presentación, esquema para presentar.
Recursos necesarios: proyecto, memoria, Google Drive.

En la siguiente tabla, se muestran las tareas, su duración y las dependencias de cada una. Como se puede observar en la Tabla 1, el número de horas totales del proyecto asciende a 437 horas.

N.º tarea	Nombre de la tarea	Predecesoras	Duración (horas)
0	Control de Q!		437
1	Organización		55
2	Planteamiento de la aplicación		15
3	Planificación de las tareas	2	5
4	Elección de software	3	5
5	Instalación de software	4	5
6	Reuniones con empresa o director	2	25
7	Aprendizaje		51
8	Lenguaje de programación C#	5	20
9	SDK de Vuforia para Unity 3D	5;8	15
10	Modelado de objetos en 3D	5;9	10
11	Edición de imágenes	5;10	6
12	Análisis y diseño		46
13	Casos de uso	8;9;11	6
14	Modelo de dominio	13	6
15	Lógica del juego	14	6
16	Diagrama de clases	8;9;15	6
17	Diagramas de secuencia	8;9;16	16
18	Pantallas	11;17	6

Tabla 1: Dependencias y duración de las tareas (1).

19	Implementación y desarrollo		130
20	Código C# del juego	8;16;17;18	30
21	Realidad aumentada	8;9;16;17;18	20
22	Objetos 3D del juego	10;18	20
23	Animaciones del juego	10;18	15
24	Juego	20;21;22;23;25	20
25	Creación del tablero	11;18	10
26	Gafas de realidad aumentada		15
27	Pruebas		44
28	Juego	24;25	10
29	Menú	28	6
30	Ranking	29	6
31	Dispositivos	30	6
32	Automatización	31	16
33	Documentación		111
34	DOP		36
35	Búsqueda de contenidos	3	6
36	Creación de diagramas y tablas	35	10
37	Redacción y edición	36	20
38	Memoria del proyecto		75
39	Redacción de la memoria	24;32;37	30
40	Redacción de los manuales	39	15
41	Preparación de la defensa	40	30

Tabla 2: Dependencias y duración de las tareas (2).

2.8. Planificación temporal

Una vez que se han definido las tareas que se realizarán en el transcurso del proyecto, queda situarlas en el tiempo mediante la creación de un diagrama Gantt.

Se ha establecido la jornada laboral en 6 horas al día, dando un total de 30 horas semanales. Los fines de semana se descansará, siempre y cuando los plazos lo permitan y no haya tareas pendientes de terminar.

En la Figura 6, aparece representado el diagrama Gantt del proyecto. Como se puede observar, se ha decidido dejar una semana aproximadamente entre los bloques de tareas y entre la preparación de la memoria y la defensa. De esta forma, se consigue un margen de espacio, para prevenir posibles problemas que pudieran surgir durante la realización del proyecto.

También cabe destacar el bloque de implementación y desarrollo. Las tareas de este bloque se realizarán de forma paralela a excepción de la tarea juego, que se realizará después, y de las gafas de realidad aumentada. Para realizar estas últimas, se aprovecharán las vacaciones de Pascua (20 Marzo - 1 Abril) mientras se va avanzando con el resto de tareas del mismo bloque.

Por último, y como se ha mencionado anteriormente, el bloque de la documentación se ha dividido en dos secciones. Una es el DOP, que se realiza al principio del proyecto, y otra la memoria, la cual se empieza a realizar una vez acabadas las pruebas.

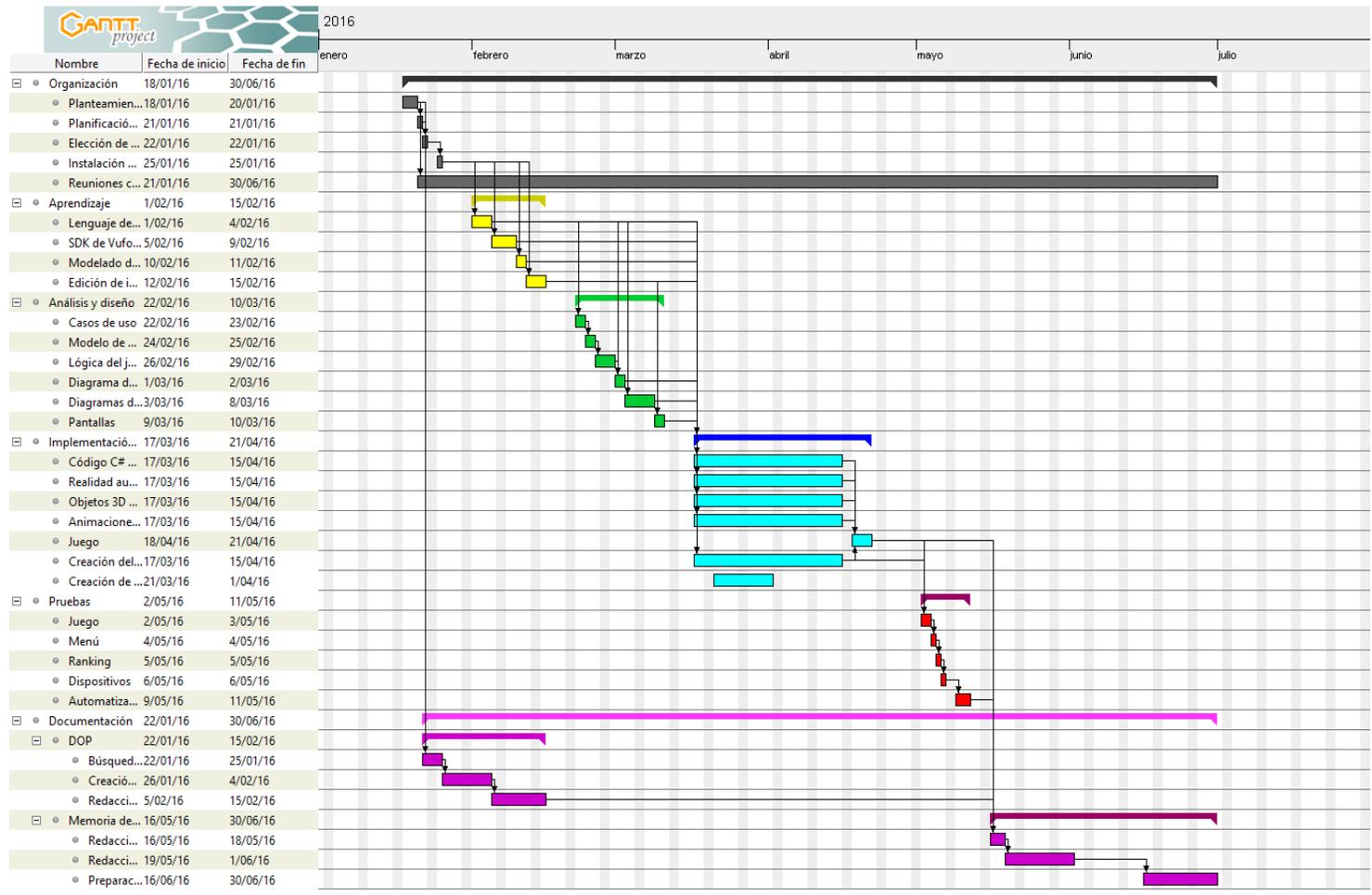


Figura 6: Diagrama Gantt

2.9. Evaluación económica

Al tratarse de un trabajo de fin de grado no se espera reporte económico con su realización. No obstante, se puede calcular el valor económico de la aplicación mediante sus gastos directos e indirectos por si en un futuro se pretendiera comercializar.

2.9.1. Mano de obra

Según lo publicado en el BOE [3], el Miércoles 10 de Febrero de 2016 (Sec III - Anexo 1, Anexo 2), se define el sueldo mensual a 12 pagas de un Programador en 1.384,09 €. El salario por hora de trabajo se puede obtener mediante la siguiente ecuación:

$$Sueldo/hora = \frac{Sueldo\ mes}{Semanas/mes * Días\ trabajo/semana * Horas\ trabajo/día} \quad (1)$$

Un mes tiene, por lo general, 4 semanas de duración, y la jornada laboral está estipulada a 5 días de trabajo a la semana y 8 horas al día. Despejando la ecuación quedaría:

$$Sueldo/hora = \frac{1384,09}{4 * 5 * 8} = \frac{1384,09}{160} = 8,65 \quad (2)$$

Dando un total de 8,65 € por cada hora trabajada.

La duración estimada del proyecto asciende a 437 horas, de las cuales 51 son dedicadas al aprendizaje de las herramientas a utilizar. Esas horas, no se van a tener en cuenta a la hora de calcular los costes de mano de obra por lo que:

$$Horas\ totales - Horas\ aprendizaje = 437 - 51 = 386\ horas \quad (3)$$

Y multiplicado por el sueldo/hora establecido anteriormente dá un total de:

$$Sueldo = Horas\ trabajo * Sueldo/hora = 386 * 8,65 = 3338,9 \quad (4)$$

El gasto económico en lo referente a la mano de obra sería de **3338,9 €**.

2.9.2. Gasto de software

Al tratarse de un proyecto de fin de grado y poder contar con las licencias de estudiante en las herramientas que lo requerían, o bien por haber usado software libre, el gasto en este apartado es **0 €**.

En caso de que la aplicación se acabe usando con fines comerciales, Unity⁹ exige, que si se generan unos 2670 € o más con la aplicación, se deberá subir la categoría de la cuenta, y empezar a pagar desde 66,75 € al mes. Los precios en las web aparecen en dólares americanos pero se ha realizado la conversión a euros.

Vuforia¹⁰ se puede usar de forma gratuita incluso con fines comerciales pero se le añadirá una marca de agua en la esquina inferior izquierda de la aplicación.

2.9.3. Gasto de hardware

En este apartado, se tratarán los aspectos económicos relacionados con la amortización del material electrónico que se usará en la realización del proyecto.

2.9.3.1. Ordenador portátil Asus F554L

La vida útil que se estima para este portátil es de unos 4 años (48 meses) y su precio fue unos 700 € aproximadamente.

Su amortización mensual es:

$$\text{Amortización mensual} = \frac{\text{Precio}}{\text{Vida útil}} = \frac{700}{48} = 14,58 \quad (5)$$

El resultado es 14,58 €/mes y como el ordenador va a ser usado durante todo el proyecto (6 meses):

$$\text{Amortización total} = \text{Amort. mensual} * \text{Meses} = 14,58 * 6 = 87,5 \quad (6)$$

Dando un total de **87,5 €** en lo referente a la amortización del ordenador.

2.9.3.2. Dispositivo móvil Motorola Moto G 4G

El precio de este dispositivo fue de 200 € y su vida útil es de 2 años (24 meses) por lo que la amortización mensual quedaría:

$$\text{Amortización mensual} = \frac{\text{Precio}}{\text{Vida útil}} = \frac{200}{24} = 8,33 \quad (7)$$

⁹Tipos de cuentas en Unity: <https://unity3d.com/es/get-unity>

¹⁰Tipos de cuentas en Vuforia: <https://developer.vuforia.com/pricing>

El resultado es 8,33€/mes, y el dispositivo móvil será usado durante el desarrollo y las pruebas de la aplicación que ocupan un periodo de unos 2 meses. Por tanto la amortización total del móvil sería:

$$\text{Amortización total} = \text{Amort. mensual} * \text{Meses} = 8,33 * 2 = 16,66 \quad (8)$$

La amortización del móvil relativa a este proyecto es de **16,66 €**.

2.9.3.3. Suma de gastos

La suma de todos los gastos relacionados con la amortización de *hardware* es:

$$\text{Gastos} = \text{Amort.ordenador} + \text{Amort.móvil} = 87,5 + 16,66 = 104,16 \quad (9)$$

Dando un total de **104,16 €** en este apartado.

2.9.4. Gastos indirectos

En este bloque, se exponen los gastos derivados del proyecto que no tienen una relación directa con el desarrollo del mismo.

2.9.4.1. Luz e Internet

El aula de empresa, en la que se va a realizar el proyecto, está dentro de la escuela, se utilizarán las tomas de corrientes de la propia facultad para conectar el ordenador, y la red eduroam para la conexión a Internet, por lo que los gastos en este apartado son **0 €**.

2.9.4.2. Bono mensual de metro Bilbao

Para llegar a la escuela, se tomará el metro Bilbao¹¹ tanto para la ida como para la vuelta. Se ha optado por coger el bono mensual, dado que resulta más económico, junto con el descuento Gazte¹² por ser menor de 26 años que son 34€ al mes y viajes ilimitados entre 2 zonas.

El proyecto se realizará desde Enero de 2016 hasta Junio del mismo año dando un total de 6 meses así que:

$$\text{Gasto en Bonos de Metro} = \text{Meses} * \text{Precio/mes} = 6 * 34 = 204 \quad (10)$$

Los gastos derivados en este apartado son de **204 €**.

¹¹Página oficial metro Bilbao: <https://www.metrobilbao.eus/>

¹²Más info: <https://www.metrobilbao.eus/utilizando-el-metro/billetes/21/billete/141>

2.9.4.3. Dietas

Con objeto de perder el menor tiempo posible, con viajes de ida y vuelta entre la universidad y el domicilio particular para comer, se llevará la comida previamente preparada en casa a la universidad. Para un correcto transporte de la comida, se comprará un porta alimentos cuyo precio es **30 €**.

2.9.4.4. Suma de gastos

La suma de todos los gastos indirectos es:

$$Gastos = Luz/Internet + Bono Metro + Dietas = 0 + 204 + 30 = 234 \quad (11)$$

Dando un total de **234 €** en esta sección.

2.9.5. Gastos totales

En este último apartado de la evaluación económica, se calcula el gasto total que origina la realización del proyecto.

En la Tabla 2.9.5, aparece reflejada la suma de todos los gastos mencionados previamente.

Tipo de Gasto	Gasto ocasionado (en €)
Mano de obra	3338,9 €
Gasto en software	0 €
Gasto en hardware	104,16 €
Gastos indirectos	234 €
Total	3577,03 €

Tabla 3: Costes totales de la aplicación

Dando un total de **3577,03 €** por el completo desarrollo de la aplicación.

2.9.6. Posibilidades de negocio

Esta aplicación, en un principio, está orientada a un Trabajo Fin de Grado sin fines lucrativos. Aun así, en este apartado, se analizarán los diferentes contextos de negocio que se podrían usar para esta aplicación de realidad aumentada, con el objetivo de generar un beneficio económico.

2.9.6.1. Google Play Store

Al tratarse de un aplicación para dispositivos Android, se hace indispensable pensar en publicar dicha app en Google Play Store¹³.

Crearse una cuenta de desarrollador en Google cuesta unos 22,25 € en un único pago y permite subir cuantas aplicaciones se quieran, actualizar sus versiones y ponerles el precio de venta que se desee. Google se llevará una comisión por cada venta que se ejecute, siempre y cuando la aplicación no sea gratuita al público. Para conocer todos los derechos y obligaciones de los desarrolladores conviene echarle un vistazo al Acuerdo de distribución para desarrolladores de Google Play [4].

A la hora de publicar aplicaciones se suele seguir unos de estos 3 modelos de negocios:

- **Aplicación gratuita:** cualquier usuario se puede bajar la aplicación gratis sin reportar ningún beneficio al desarrollador.
- **Aplicación semi-gratuita:** al igual que la gratuita cualquiera se puede bajar la aplicación a coste cero pero con unas funcionalidades limitadas. El usuario que quiera disfrutar de todas las funcionalidades tendrá que pagar una suma concreta de dinero.
- **Aplicación de pago:** el usuario deberá de pagar cierta cantidad de dinero establecida por el desarrollador para disponer de dicha aplicación.

En el caso de esta aplicación lo mejor sería optar por la segunda opción. Lanzar la aplicación a coste cero conseguiría llegar al máximo número de personas posibles, para luego, añadir alguna funcionalidad adicional de pago en una actualización de la versión y empezar a generar beneficios.

2.9.6.2. Publicidad

En caso de que la aplicación tenga un rotundo éxito, traducido en miles y miles de descargas, se podría estudiar la situación de añadir publicidad dentro de la aplicación mediante Google AdSense¹⁴ o alguna aplicación similar.

El problema que se encuentra, en el caso de que la aplicación sea de pago, es el impacto negativo que puede causar a los usuarios la introducción

¹³Más información: <https://play.google.com/store?hl=es>

¹⁴Más información: <https://www.google.com/adsense/start/?hl=es>

de publicidad en la aplicación. A los usuarios, por lo general, una vez que han pagado por la aplicación, no les hace ninguna gracia que les aparezca publicidad en ella.

Esto originará malas valoraciones y peores comentarios acerca de la aplicación, lo que puede suponer perder posiciones en los *rankings* y reducir considerablemente el número de descargas de la misma.

Por eso, en caso de que se la aplicación tenga un precio de descarga, hay que ver una vez que se haya asentado en el mercado, qué beneficios económicos otorga la aplicación sin publicidad, y calcular su posible impacto tanto positivo como negativo si se introdujera publicidad. Es importante saber que se suele generar más dinero con una descarga de una aplicación cuyo coste sea menor de 1 €, que con miles de *clicks* sobre un marcador de publicidad.

En caso de que se publique la aplicación a coste cero en Google App Store, es más normal añadir publicidad para generar beneficio, y los usuarios, al contrario que en el caso anterior, lo suelen ver con buenos ojos dado que la aplicación les ha resultado gratuita.

2.10. Riesgos

En este apartado, se hablará acerca de los posibles riesgos que pueden surgir durante la realización del proyecto, su plan de prevención y la probabilidad de que sucedan.

Se tendrá en cuenta que hay diferentes tipos de riesgos que dependiendo de su gravedad, puedan afectar de una mayor o menor manera al desarrollo del proyecto.

Se han definido 3 tipos de impacto dependiendo del tiempo invertido en solventar el problema.

- **Impacto bajo:** se resuelve el problema en menos de 1 día laboral (<6 horas).
- **Impacto medio:** se resuelve el problema en menos de 3 días laborables (6 - 24 horas).
- **Impacto alto:** se tarda en resolver el problema más de 3 días laborables (>24 horas).

2.10.1. Problemas con eduroam

Algunos días, la conexión a eduroam de la universidad es limitada o nula, por lo que no sería posible avanzar en algunos aspectos del proyecto tales como, la subida de imágenes al servidor de Vuforia para su evaluación de contraste o, no disponer del trabajo actualizado porque la última versión se encuentra en Dropbox.

Prevención

- Tener el trabajo sincronizado en el ordenador y en Dropbox.

Plan de contingencia

- Realizar la parte del trabajo que no requiera conexión a Internet.
- En caso de que el problema persista durante varios días se irá avanzando con esa parte del proyecto en el domicilio particular.

Probabilidad

- Quedarse sin conexión a Internet en la universidad. Baja: 5%.

Impacto

- Caída del servidor eduroam. Impacto bajo.

2.10.2. Fallos en el ordenador

Puede darse el caso de que el ordenador empiece a dar fallos, tanto de *software*, como de *hardware*, poniendo en peligro la información del trabajo que se encuentre en ese momento en el ordenador sin copia de seguridad.

Prevención

- Disponer de un buen antivirus en el ordenador.
- Ser cauto a la hora de instalar programas o descargarse cierto tipo de archivos.
- Conservar el ordenador en las mejores condiciones posibles.

Plan de contingencia

- Si es un problema de *software*, se intentará buscar una solución por Internet. En caso de que el fallo sea imposible de arreglar, se volverá a dejar el ordenador como salido de fábrica salvando la máxima información posible.
- Si resulta un fallo de *hardware*, se llevara el ordenador a un experto para su arreglo. Se deberá buscar un ordenador sustituto para perder el menor tiempo posible.

Probabilidad

- Problema de *software*. Baja: 15 %.
- Problema de *hardware*. Baja: 5 %.

Impacto

- Fallo de *software*. Impacto medio-bajo.
- Fallo de *hardware*. Impacto alto.

2.10.3. Problemas con C#, Unity, Vuforia, Blender o GIMP

Es complicado predecir los problemas que pueden surgir con los programas mientras se está realizando el juego, y más aún, cuando no se está familiarizado con ninguna de estas aplicaciones.

Prevención

- Tener una buena curva de aprendizaje con los programas.
- Conocer las mejores paginas y foros especializados para consultar y extraer información.

Plan de contingencia

- Para un problema sencillo, se procederá a buscar en Internet el código de error que arroja el programa y coger la mejor respuesta para poder arreglar el fallo.
- Si el problema resulta un poco más complicado, se avanzará el proyecto por otra vía y se volverá a intentar solventarlo en otro momento.
- En caso de que sea un problema de difícil solución, se procederá a pedir ayuda a los miembros de la empresa *Creativitic* con objeto de perder el menor tiempo posible.

Probabilidad

- Problema fácil solución. Media: 50 %.
- Problema solución intermedia . Media: 25 %.
- Problema difícil solución. Baja: 5 %.

Impacto

- Solución fácil. Impacto bajo.
- Solución intermedia. Impacto medio.
- Solución difícil. Impacto alto.

2.10.4. Falta de inspiración

La productividad de un trabajador no es la misma todos los días. Esto puede deberse a diversos factores tales como problemas emocionales, cansancio o la incapacidad de solventar un problema surgido en un determinado momento.

Prevención

- Dormir 8 horas al día e intentar llevar un estilo de vida tranquilo.
- Ser benévolo a la hora de establecer los plazos del proyecto.

Plan de contingencia

- Si resulta ser un problema emocional o cansancio, se cogerá la tarde libre para relajarse, y así coger fuerzas para proseguir al día siguiente con el proyecto.
- Para el caso de bloqueo en algún punto concreto del proyecto, se procederá a empezar a hacer otra parte del proyecto independiente a la anterior con objeto de no perder tiempo.

Probabilidad

- Problema emocional o cansancio. Baja: 15 %.
- Bloqueo en proyecto. Media: 25 %.

Impacto

- Problema emocional o cansancio. Impacto bajo.
- Bloqueo en proyecto. Impacto bajo.

3. Antecedentes

En este bloque, se van a detallar los programas que comparten características con el proyecto a desarrollar. Para ello, se ha querido dividir esta sección en dos apartados independientes. Por un lado, se hablará del proyecto de fin de grado de un compañero utilizando realidad aumentada, y por el otro, se mencionará un conocido juego electrónico el cual guarda una pequeña relación con la aplicación que se va a desarrollar.

3.1. Aprende: números y letras

Aplicación de realidad aumentada creada por Iñigo Allende [5], en colaboración con la empresa Creativitic, al igual que este proyecto. Se trata de una app enfocada a niños pequeños, para ayudarles con el aprendizaje de números y palabras, utilizando un dispositivo Android con cámara incorporada. Al igual que el juego a desarrollar, este programa se ha realizado con Unity y el SDK de Vuforia.

La aplicación se compone de una serie de marcadores físicos en los cuales aparecen números, objetos, palabras y dibujos. Estos marcadores tienen en los bordes una serie de cuadrados blancos, negros y grises, los cuales utiliza el programa para diferenciar a cada marcador.

Por ejemplo, si el usuario selecciona la opción cuentas desde el menú, deberá enfocar a un marcador que tenga un número y se verá en la pantalla del dispositivo el mismo número de bolas virtuales sobre el marcador.

Otro caso, es la combinación de marcadores, como en la opción competencia lingüística. Aquí, el usuario deberá combinar una palabra con la imagen asociada. Si los marcadores son los correctos, entonces se reproducirá su correspondiente audio.

En el proyecto a desarrollar, se va a utilizar una única imagen objetivo como tablero en detrimento de los marcadores. De todas formas, no hubiera sido posible utilizar marcadores dado que se quieren usar botones virtuales en la aplicación y estos deben ir asociados a una imagen objetivo y no a un marcador.

3.2. Simon

Conocido juego electrónico de memorizar patrones que tuvo un gran éxito en los años 80. Se trata de un tablero con 4 botones de colores que se iluminan y generan diferentes sonidos.

El funcionamiento del juego es el siguiente, Simon crea una secuencia iluminando los botones en un orden concreto, a continuación, el usuario tiene que ir pulsando correctamente la secuencia de los botones para avanzar de nivel. Según vas avanzando de nivel, se va aumentando la longitud de la secuencia y la velocidad con la que se iluminan los botones. Si el usuario falla, pierde.

Es propiedad de Hasbro y el diseño actual del juego se puede ver en la Figura 7¹⁵.



Figura 7: Imagen del juego Simon

El juego a realizar pretende usar la realidad aumentada para simular, mediante botones virtuales, botones físicos del estilo de Simon, pero con una lógica interna distinta, eliminando las secuencias y usando un temporizador. Esto hace que el usuario tenga un tiempo límite para pulsar los botones que se le vayan activando en el tablero.

El juego Simon aumenta la capacidad de memorizar del usuario, mientras que, con la aplicación a desarrollar, se quieren potenciar los reflejos, la velocidad y la precisión del usuario.

¹⁵Más información: <http://www.hasbro.com/es-es/product/simon-swipe-game:FA45E76C-5056-9047-F541-122FDDEF7288>

4. Captura de requisitos

En esta sección, se van a tratar los diferentes requisitos a tener en cuenta para un correcto desarrollo y funcionamiento de la aplicación. Se tratará de satisfacer, en lo máximo posible, las necesidades de los usuarios.

4.1. Jerarquía de actores

La aplicación no necesita registro previo y todas las funcionalidades están accesibles para cualquier usuario. Por ello, sólo se ha definido un tipo de actor, que es el propio usuario, como se puede ver en la Figura 8.

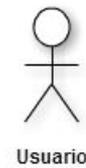


Figura 8: Jerarquía de actores

4.2. Casos de uso

Los casos de uso se utilizan para describir los pasos que deben hacer los actores para realizar acciones dentro del sistema. En este caso, solo hay un tipo de actor y su diagrama de casos de uso correspondiente se muestra en la Figura 9.

- **Seleccionar nivel:** el usuario selecciona el nivel de dificultad con el que quiere jugar a la aplicación. Una vez seleccionado, se le permite al usuario jugar una partida en esa dificultad.
- **Ranking:** se le permite al usuario visualizar el *ranking* de mejores puntuaciones por dificultad en la pantalla del dispositivo.
- **Salir:** se le permite al usuario salir de la aplicación.

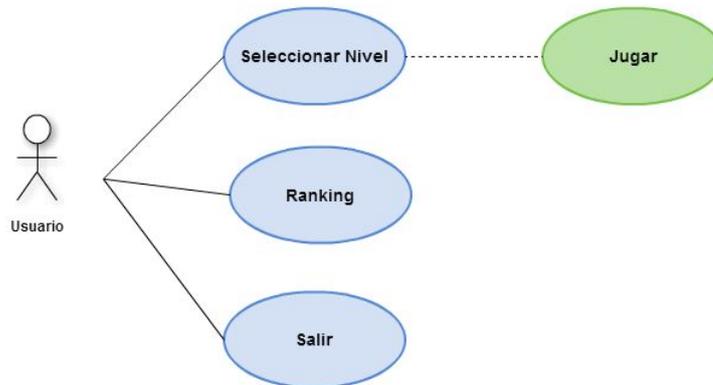


Figura 9: Diagrama de casos de uso

4.3. Modelo de dominio

El modelo de dominio representa de una forma conceptual aquellas entidades que forman parte de la aplicación y su relación entre ellas.

La descripción de cada entidad viene detallada a continuación:

- **Cerrojo:** entidad que representa el objeto virtual que se utiliza para dar la sensación al usuario de estar introduciendo la mano en un cerrojo real. Cada cerrojo lleva asociado una animación y 4 botones virtuales, 3 de frontera y 1 de pulsación correcta.
- **Animación:** entidad que representa el efecto especial correspondiente a cada cerrojo. Cada animación va asociada a un cerrojo.
- **Partida:** entidad que representa una partida del juego. Cada partida tiene asociados 5 cerrojos y la puntuación obtenida en ella se actualiza en el *ranking*, dependiendo de su atributo dificultad.
- **Imagen Objetivo:** entidad que representa el tablero de la aplicación. Se usará la misma imagen objetivo en todas las partidas del juego. En esa imagen estarán situados todos los botones virtuales de la aplicación.
- **Botón Virtual:** entidad que representa los botones virtuales de la partida y su posición con respecto a la imagen objetivo.
- **Ranking:** entidad que representa las puntuaciones obtenidas en el dispositivo. El *ranking* se actualiza si la puntuación obtenida en ese nivel ha sido mayor que las actuales.

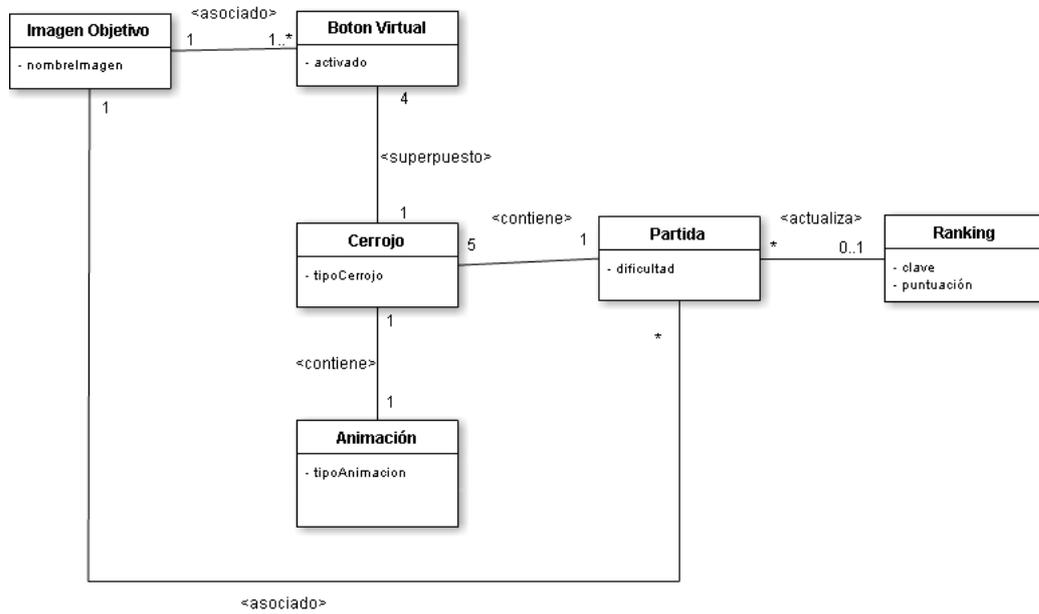


Figura 10: Modelo de dominio

Algunos nombre de las entidades del modelo de dominio no coinciden con el diagrama de clases (Figura 13) implementado. Las entidades de Imagen Objetivo y Botones Virtuales, por ejemplo, son scripts predefinidos de Vuforia, que no se han representado en dicho diagrama. Además, se han añadido otras clases buscando hacer más modular el código a implementar.

5. Elección de lenguajes y tecnologías

Una correcta selección de las herramientas a utilizar se antoja fundamental para un correcto desarrollo del proyecto. Una mala elección, puede suponer una gran pérdida de tiempo, por posibles incompatibilidades de formatos a la hora de importar archivos a otras herramientas, o por resultar la herramienta mucho más compleja de utilizar de lo que se pensó en un primer momento.

A continuación, se exponen las diferentes tecnologías usadas en el proyecto y los motivos de su utilización.

5.1. Unity y el lenguaje de programación C#

El motor gráfico se trataba de una herramienta fundamental a la hora de desarrollar el proyecto. Con Unity, se abría la posibilidad de utilizar todas sus características y poder distribuir el producto con una licencia gratuita, algo que antes estaba limitado a usuarios de pago.

Finalmente, se eligió Unity, porque ofrecía una gran cantidad de recursos online para consultar dudas y tiene una de las mayores comunidades especializadas. Además, es multiplataforma, capaz de exportar a 21 plataformas diferentes entre las que se encuentra Android, PlayStation o Xbox entre otras.

Unity utiliza 2 lenguajes de programación para la creación de juegos, C# y JavaScript. Se decidió elegir el primero, puesto que es el lenguaje en el que vienen programados los scripts del SDK de Vuforia del cual se habla en la siguiente sección.

5.2. SDK de Vuforia

El motivo principal por el que se ha elegido esta herramienta de realidad aumentada ha sido por su compatibilidad con Unity mediante su SDK. Además, al venir programado en C#, que es el mismo lenguaje que usa Unity, hace la vida más sencilla al desarrollador debido a que sólo requiere del aprendizaje de ese lenguaje de programación para la implementación de todo el juego.

5.3. Blender

A la hora de elegir el software para crear los objetos en 3D, se ha buscado que tenga una curva relativamente rápida de aprendizaje, que sea de código abierto y que no tenga problemas de incompatibilidades con Unity. Al final, se ha elegido el programa de modelado de objetos Blender porque era el que mejor cumplía esos requisitos.

5.4. GIMP 2

Se ha seguido una búsqueda similar a la del apartado anterior para elegir el software de diseño de imágenes. Una vez vistas las posibles alternativas, se decidió optar por GIMP 2 para la creación del logotipo y del tablero, por la facilidad de uso que ofrecían sus herramientas de diseño y la posibilidad de diseñar por capas la imagen. Además, es software de código abierto.

6. Análisis y diseño

Por un lado, se va a realizar un análisis de los elementos que requieren una mención especial dentro de la aplicación, con objeto de entender mejor su funcionamiento. Por el otro, se van a explicar los diagramas de estados, clases y secuencias que se han realizado para su posterior desarrollo.

6.1. Análisis de los elementos del juego

En este apartado, se va a intentar explicar de una forma sencilla, aquellos elementos internos de la aplicación que no se han detallado a lo largo de la memoria, pero que son indispensables para un correcto funcionamiento de la aplicación.

6.1.1. Escenas

En Unity, las escenas son independientes entre sí. Cada escena posee diferentes `GameObjects` que se cargan cuando se lanza esa escena, y se eliminan de memoria una vez se cambia de escena. Para moverse a través de las diferentes escenas del juego, se utilizará el método `LoadScene()` de la clase `SceneManager` propia de Unity.

6.1.2. *Ranking*

Para el almacenamiento de las mejores puntuaciones, se va a utilizar la clase `PlayerPrefs` procedente de las librerías propias de Unity. Con ella, se pueden almacenar valores que no se quieran perder cuando el usuario cierre y vuelva a abrir la aplicación. Las puntuaciones serán guardadas con una estructura de tipo (clave, puntuación).

6.1.3. *Scripts: Start(), Awake(), Update()*

Los scripts van asociados a los `GameObject` dentro de la escena, por lo que se cargan a la par que estos. Tienen una serie de funciones reservadas que se han utilizado en la implementación del juego, que son las siguientes:

- ***Start()***: esta función se ejecuta cuando se carga el script. Se va a utilizar para inicializar variables y lanzar los métodos necesarios para que empiece a funcionar dicho *script*.
- ***Awake()***: funcionamiento idéntico al *Start()* pero con una orden de prioridad mayor. Se utilizará para inicializar ciertas variables en primera instancia dentro del propio script y también, con respecto al resto de *scripts* de la escena que no tengan implementada esta función.
- ***Update()***: función que se ejecuta a cada *frame* mientras la escena se encuentra activada. Se usará para el control del temporizador, activar/desactivar los botones y, en general, para la gestión del juego en tiempo real.

En algunos *scripts* las variables solo serán declaradas en MonoDevelop para luego ser inicializadas en Unity, dado que éste permite realizar *Drag&Drop* de los *GameObjects* sobre las variables de los *scripts* previamente declaradas.

6.1.4. Objetos virtuales

Se va a utilizar la herramienta de modelado de objetos Blender para realizar los cerrojos y el panel de información de la partida.

Con respecto a los cerrojos se creará una única pieza a la que luego se le realizarán ciertos cortes con la herramienta *Knife Project*¹⁶ de Blender, para separar las piezas y así poder abrir y cerrar el cerrojo desde Unity activando y desactivando dichas piezas.

¹⁶Más información: https://www.blender.org/manual/modeling/meshes/editing/subdividing/knife_subdivide.html

6.1.5. Animaciones

Para la creación de las animaciones correspondientes a los cinco elementos de la temática, se van a utilizar diversas herramientas que vienen por defecto en Unity las cuales se exponen a continuación:

- ***ParticleSystem***: es un editor de efectos especiales. Con él, se pueden crear desde explosiones hasta lluvias de estrellas. Se va a utilizar para crear las animaciones referentes al agua, fuego y tierra. Dispone de una gran variedad de parámetros internos que se irán variando para conseguir dichos efectos.
- **Paquete *Tree***: este paquete de Unity es un generador de árboles. Se pueden crear todo tipo de árboles con el. Se recreará la animación de la madera variando los parámetros internos al igual que con la animación anterior.
- **Generador Objetos Básicos**: en Unity se pueden crear objetos tridimensionales básicos como cubos, esferas o cilindros para luego cambiarles las dimensiones o el color. Se utilizará para la animación correspondiente al metal.

6.1.6. Imagen objetivo y botones virtuales

La imagen que se quiera usar en el tablero tendrá que ser evaluada con una calificación de 5 estrellas, que es el máximo posible. Para ello, Vuforia¹⁷ establece 3 criterios para conseguir esa valoración. Esos criterios son:

1. **Riqueza de detalles**: los elementos de la imagen deben tener una buena resolución y con muchos detalles diferentes entre sí. Un buen ejemplo sería un frutero con su diversidad de frutas.
2. **Buen contraste**: Vuforia evalúa las imágenes en blanco y negro por lo que es necesario convertir la imagen del tablero a este formato e intentar que haya el mayor contraste de grises posible.
3. **Evitar patrones repetitivos**: Hay que intentar buscar elementos naturales para la creación del tablero dado que estos, por lo general, no siguen un patrón determinado.

¹⁷Más información: https://developer.vuforia.com/library/articles/Best_Practices/Recommendations-for-Improving-Target-Detection-and-Tracking-Stability

En la Figura 11 se pueden ver los puntos de referencia que crea Vuforia, a partir de los anteriores criterios, sobre la imagen de un árbol, para posteriormente, evaluar y calificar la imagen.



Figura 11: Puntos referencia en una imagen objetivo

Respecto a los botones virtuales que se van a utilizar, cabe destacar que, cuantos más puntos de referencia tenga la imagen sobre el lugar donde se posiciona el botón, mejor funcionará este, porque la cámara puede captar mejor ese trozo de la imagen para saber si se ha puesto la mano encima o no.

6.2. Diagrama de estados

En este apartado, se detallan los diferentes estados de la aplicación y su relación entre ellos. El diagrama de estados correspondiente a la aplicación se muestra en la Figura 12.

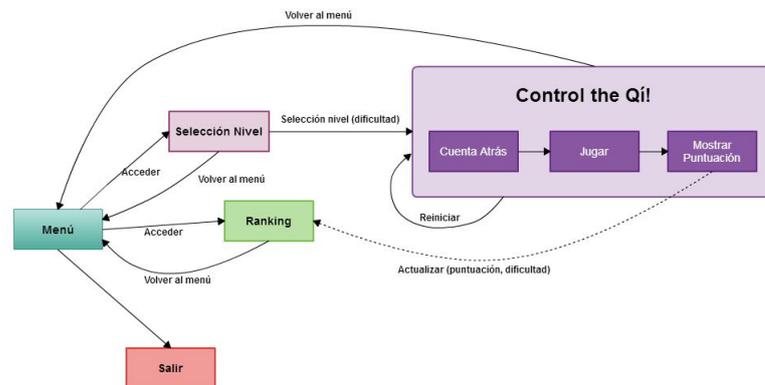


Figura 12: Diagrama de estados

- **Menú:** es el primer estado que aparece una vez que se lanza la app. Desde él, se puede acceder a la selección de nivel del juego, a visualizar el ranking o a salir de la aplicación.
- **Selección nivel:** en este estado, se selecciona la dificultad de la partida y se lanza el juego, o también se puede volver al menú.
- **Ranking:** en este estado, se muestran las 3 mejores puntuaciones conseguidas en cada nivel. Dispondrá de una opción para volver al menú principal.
- **Control the Qí!:** en este bloque, se albergan los estados que corresponden a la ejecución del juego. Se necesita la cámara del dispositivo móvil y el tablero de la app para su funcionamiento y consta de los siguientes 3 estados:
 1. **Cuenta atrás:** secuencia de 10 segundos de preparación para el usuario. Este estado se lanza cuando la cámara del dispositivo localiza el tablero.
 2. **Jugar:** en este estado el usuario jugará con la aplicación utilizando sus propias manos para cerrar los cerrojos virtuales que le aparezcan por pantalla sobre el tablero. Se terminará cuando se alcance un número determinado de fallos sobre uno de los cerrojos.
 3. **Mostrar puntuación:** estado en el que se muestra los puntos obtenidos por el usuario y se actualiza el *ranking*.

En cualquier momento durante la ejecución de este bloque de estados, mediante la pulsación de ciertos botones virtuales, se podrá volver a reiniciar la partida con el mismo nivel, o volver al menú.

- **Salir:** estado para salir de la aplicación.

6.3. Diagrama de clases

Un diagrama de clases es un buen método de ver la estructura de una aplicación orientada a objetos. En él aparecen los atributos y métodos de cada entidad así como su relación.

En la Figura 13 se muestra el diagrama de clases correspondiente a la aplicación. El único cambio relevante respecto al diseño inicial ha sido la creación de las clases **Animaciones** y **Cerrojos** que estaban anteriormente dentro de la clase **ControladorJuego**. Los motivos fueron la búsqueda de modularidad y la adaptación a la temática que finalmente se le ha dado al juego.

- **ControladorJuego**: es el *script* principal del juego, el cual se encarga de activar/desactivar los botones, gestionar tanto el tiempo, la pulsación correcta de los botones y el estado de la partida.
- **ControladorBotonesVirtuales**: controla cuando el usuario pulsa o suelta los botones virtuales del tablero.
- **ControlBotones**: guarda las teclas activadas y desactivadas en cada momento de la partida. Sirve para saber si el usuario ha tocado algún botón frontera cuando ha puesto la mano sobre el cerrojo.
- **Ranking**: gestiona los mejores resultados obtenidos por nivel, ordenándolos y mostrándolos por pantalla.
- **CambiarEscenas**: controla el cambio de escenas de la aplicación.
- **Animaciones**: controla las animaciones que aparecen en la partida.
- **Cerrojos**: controla los cerrojos que aparecen en la partida.



Figura 13: Diagrama de clases

6.4. Diagramas de secuencia

Los diagramas de secuencia representan la iteración entre los diferentes scripts de la escena para realizar alguna acción determinada. Se han creado los diagramas de secuencia que pertenecen al funcionamiento interno de una partida, desde que se inicia la cuenta atrás, hasta que se visualiza la puntuación obtenida. Se han creado 2 diagramas independientes para su correcta visualización, pero funcionan en paralelo. En el primero, aparece la gestión de la partida y en el segundo, la gestión de los botones virtuales durante la partida.

El juego dispone de 5 cerrojos con sus respectivos botones virtuales, pero en los diagramas sólo aparece el funcionamiento de un cerrojo para no resultar redundante, pero se le aplica la misma lógica a todos. Como se puede observar en la Figura 14, la secuencia temporal de una partida queda dividida en 3 secciones.

La primera es la cuenta atrás, en la cual se activan y desactivan los cerrojos a modo de entrada. A continuación empieza la partida, en este apartado se controla el tiempo de cada ronda activando y desactivando los cerrojos, así como si el usuario no ha introducido la mano en un cerrojo a tiempo. También se controla, a cada iteración, el límite de fallos en cada cerrojo, para acabar la partida en cuanto un contador de fallos llegue a 4. Por último, tenemos la sección en la que se muestra el resultado obtenido.

En el otro diagrama de secuencia, representado en la Figura 15, aparece el control de los botones durante el desarrollo de la partida.

Por un lado, la aplicación va a disponer de un conjunto de instancias booleanas, representando a los botones frontera de los cerrojos, en la clase ControlBotones, en la cual se activarán dichas instancias si el usuario pone la mano sobre el botón virtual o se desactivarán si la quita.

Por el otro lado, se controla cuando el usuario pulsa el botón correcto de un cerrojo, para ver si ese botón está activado y no se ha pulsado ninguno de sus botones fronteras al mismo tiempo. Si todo es correcto, el cerrojo se cerrará y la puntuación será aumentada. En caso de que el usuario pulse alguno de sus botones fronteras a la vez, se iluminará en naranja el cerrojo y no se cerrará. Si el cerrojo no está activado y el usuario pulsa el botón, se le añadirá un fallo al contador de ese cerrojo.

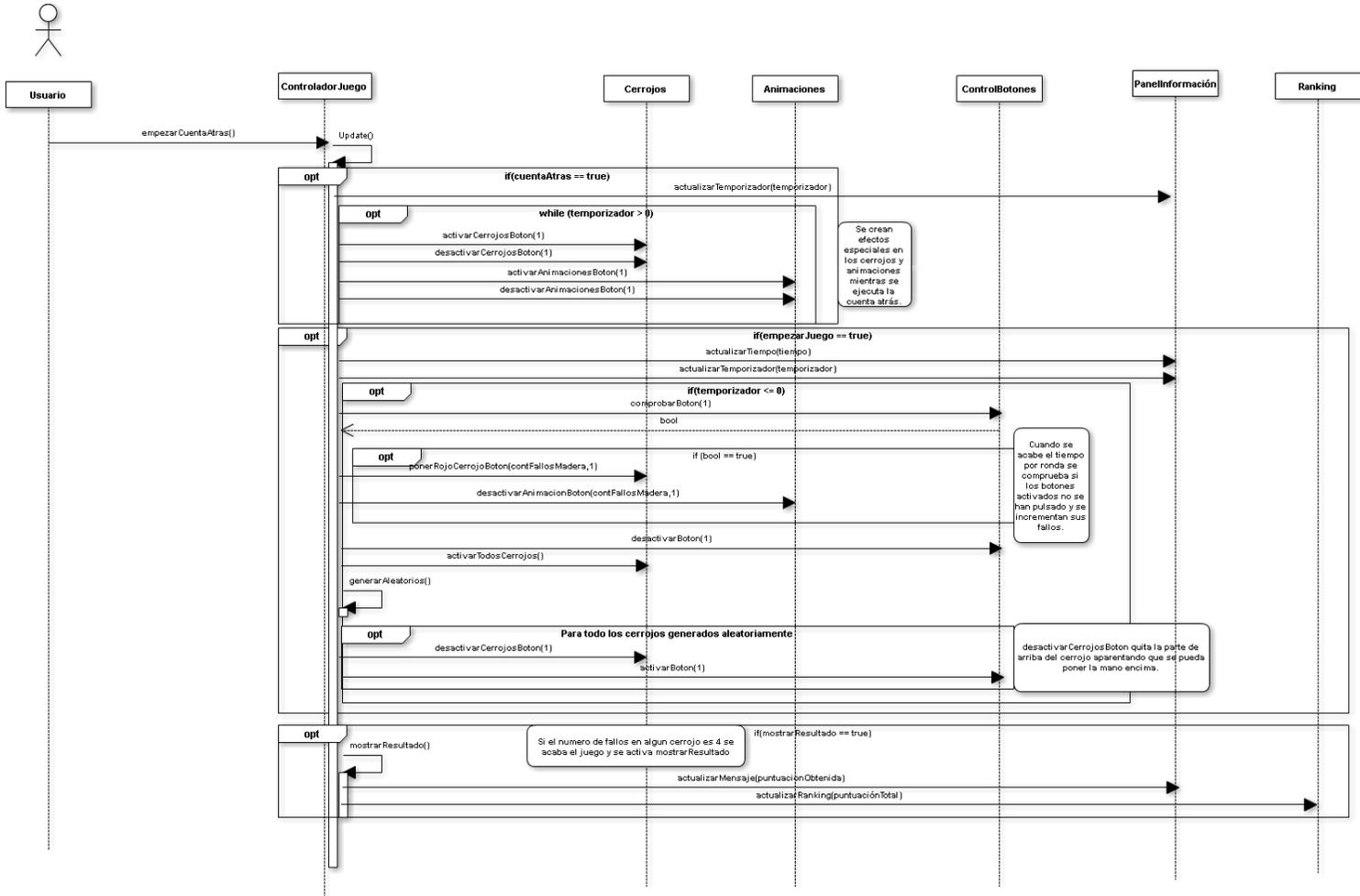


Figura 14: Diagrama de secuencia de la partida (1)

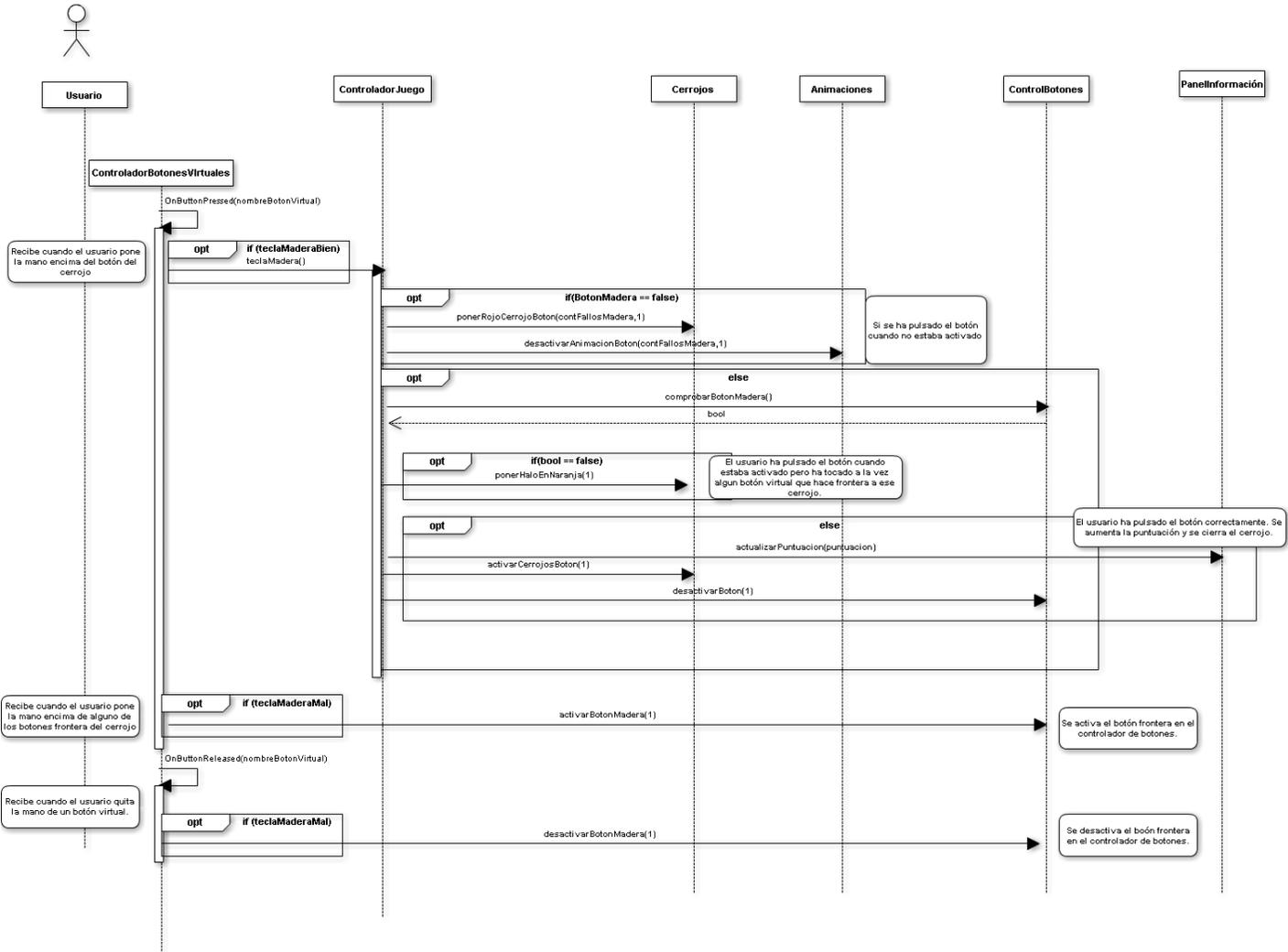


Figura 15: Diagrama de secuencia de la partida (2)

7. Desarrollo

En esta sección se van a explicar de una forma sencilla los pasos realizados para la elaboración de la aplicación. También se expondrá la evolución que han tenido los diseños iniciales a la hora de adaptarlos a la temática que finalmente se le ha dado al juego.

Cabe destacar que se ha realizado un cambio respecto al alcance y la planificación temporal (Sec. 2.7 y 2.8 respectivamente) , en los bloques de análisis y diseño e implementación y desarrollo. La nueva organización de las tareas aparece reflejada en la planificación final (Ver Sec. 9.1).

7.1. Desarrollo inicial de la aplicación

Al no disponer de la temática del juego al comienzo de la implementación, se dejó a un lado el carácter estético del mismo. Se creó una estructura de juego general y funcional, lo que facilitaba adaptar la temática y perder el menor número de horas posible en el futuro.

En la Figura 16 aparece el diseño inicial del menú:



Figura 16: Diseño inicial del menú

Como se puede observar, el tamaño de los botones y de la información que contienen debería ser más grande con el fin de facilitar la lectura y pulsación al usuario. Las pantallas de selección nivel y *ranking* también se crearon con un diseño similar por lo que presentaban el mismo problema que se solventó en el desarrollo final del juego.

El tablero que se usó en este diseño inicial aparece en la Figura 17 junto con el resto de objetos virtuales del juego. Se empezó a usar una imagen de un árbol como referencia, debido al gran contraste que ofrecían sus ramas, y de esta forma, poder garantizar el correcto funcionamiento de todos los botones virtuales.



Figura 17: Diseño inicial del juego

7.2. Desarrollo final de la aplicación

En un buen día de inspiración de finales de Marzo, se decidió que la temática del juego iba a ser sobre el Qí, el Yin Yang y los 5 elementos, disponible en el manual de usuario (Sección A.3), y se adaptó a todo lo implementado y diseñado anteriormente.

Respecto a las escenas del menú, selección de nivel y *ranking*, se rediseñaron por completo basándose en la temática y en los fallos vistos durante el desarrollo inicial. En la Figura 18 aparece la versión definitiva del menú.



Figura 18: Diseño final del menú

A la hora de crear el tablero se tuvieron en cuenta los criterios que establecía Vuforia mencionados en la Sección 6.1.6 para garantizar la calidad y el contraste de la imagen. El tablero final que se ha creado aparece en la Figura 19.



Figura 19: Tablero final de Control the Qi!

Como se puede observar, en las zonas que delimitan los botones sobre el tablero se han añadido diferentes imágenes representando cada uno de los cinco elementos. A estas imágenes, se les ha aumentado el contraste con respecto al fondo del tablero. Con esto, se consigue aumentar los puntos de referencia en esos lugares concretos del tablero que son en los que el usuario va a tener que poner la mano encima para tocar los botones virtuales.

En la Figura 20 aparecen los puntos de referencia obtenidos mediante la evaluación de la imagen desde la página de Vuforia. Cabe añadir que esta imagen ha obtenido la máxima calificación por parte de Vuforia que son 5 estrellas.



Figura 20: Puntos referencia del tablero

Como se puede apreciar, la gran mayoría de los puntos de referencia aparecen en las áreas donde están marcados los cerrojos. En la Figura 21 aparece el posicionamiento que se le ha dado a los botones virtuales en el tablero.

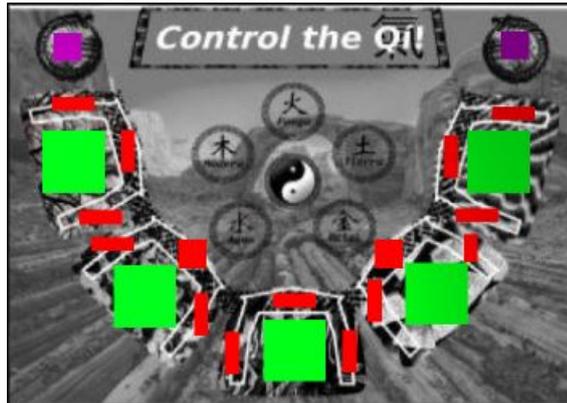


Figura 21: Localización botones virtuales en el tablero

Los botones virtuales sólo pueden adquirir forma cuadrangular o rectangular, por lo que se han intentado cuadrar de la mejor manera posible con la superficie de los cerrojos y con los puntos de referencia de la Figura 20.

Los botones verdes representan el lugar en el que tiene que poner la mano el usuario, los botones rojos son usados como fronteras para controlar si el usuario introduce la mano correctamente y los botones morados representan las opciones de reiniciar y salir.

El diseño final del juego difiere bastante en estética del creado en el desarrollo inicial (Figura 17). El cambio más visual es el aumento de cerrojos así como la mejora de los mismos. También se han creado las animaciones correspondientes a la temática, se ha añadido el temporizador en medio del tablero y se ha puesto un panel informativo. Todos estos cambios se pueden ver en la Figura 22.



Figura 22: Imagen de Control the Qí!

Aparte de estos cambios visuales se han tenido que rediseñar los diagramas de clases y secuencia con respecto al desarrollo inicial, por lo que también se tuvo que cambiar el código de programación C# generado. Todos estos cambios no supusieron una gran pérdida de tiempo dado que en el desarrollo inicial se intentó crear una estructura lo más neutral posible para tener los menores problemas posibles a la hora de adaptar el juego.

Para saber cómo se juega a Control the Qí!, las diferentes animaciones creadas, los tipos de objetos virtuales generados y el resto de información relativa al juego hay que dirigirse al manual de usuario (Anexo A)

8. Pruebas

En esta sección se van a exponer las pruebas que se han realizado para comprobar el correcto funcionamiento de la aplicación. Cabe mencionar que se han utilizado pruebas de caja negra, las cuales reciben una entrada y devuelven una salida sin dar importancia al funcionamiento interno. Este tipo de pruebas resultan sencillas de implementar para el desarrollador y, aunque se varíe el código fuente en un futuro se pueden seguir manteniendo la gran mayoría de ellas.

8.1. Pruebas del menú

En primer lugar, se exponen las pruebas realizadas para asegurar la correcta movilidad del usuario a través de la aplicación.

<i>Prueba 1 del menú</i>
Descripción de la prueba: desde la pantalla del menú principal pulsar la tecla Juego Survival.
Salida esperada: se cambia a la escena de seleccionar nivel.
Resultado obtenido: éxito.

<i>Prueba 2 del menú</i>
Descripción de la prueba: desde la pantalla de seleccionar nivel se pulsa la tecla fácil.
Salida esperada: se cambia a la escena de la partida, se activa la cámara del dispositivo apareciendo lo que enfoca por pantalla y se actualiza la dificultad.
Resultado obtenido: éxito.

<i>Prueba 3 del menú</i>
Descripción de la prueba: desde la pantalla de seleccionar nivel se pulsa la tecla medio.
Salida esperada: se cambia a la escena de la partida, se activa la cámara del dispositivo apareciendo lo que enfoca por pantalla y se actualiza la dificultad.
Resultado obtenido: éxito.

<i>Prueba 4 del menú</i>
Descripción de la prueba: desde la pantalla de seleccionar nivel se pulsa la tecla difícil.
Salida esperada: se cambia a la escena de la partida, se activa la cámara del dispositivo apareciendo lo que enfoca por pantalla y se actualiza la dificultad.
Resultado obtenido: éxito.

Cabe mencionar que para conseguir el éxito en estas 3 últimas pruebas se ha guardado la dificultad pulsada por el usuario de la misma forma que se hiciera con las puntuaciones del *ranking* (Sección 6.1.2), utilizando los *PlayerPrefs* con la estructura (dificultad, nivel) para posteriormente, poder acceder desde la escena del propio juego.

<i>Prueba 5 del menú</i>
Descripción de la prueba: desde la pantalla de seleccionar nivel se pulsa la tecla volver.
Salida esperada: se vuelve al menú principal.
Resultado obtenido: éxito.

<i>Prueba 6 del menú</i>
Descripción de la prueba: desde el menú principal pulsar la tecla <i>ranking</i> .
Salida esperada: se cambia a la escena del <i>ranking</i> .
Resultado obtenido: éxito.

<i>Prueba 7 del menú</i>
Descripción de la prueba: desde la escena del <i>ranking</i> pulsar el botón de volver al menú.
Salida esperada: se vuelve al menú principal.
Resultado obtenido: éxito.

<i>Prueba 8 del menú</i>
Descripción de la prueba: desde la escena del juego poner la mano en el tablero sobre el botón virtual de reiniciar.
Salida esperada: vuelve a comenzar la cuenta atrás del juego.
Resultado obtenido: éxito.

<i>Prueba 9 del menú</i>
Descripción de la prueba: desde la escena del juego poner la mano en el tablero sobre el botón virtual de volver al menú.
Salida esperada: se vuelve a la escena del menú principal.
Resultado obtenido: éxito.

8.2. Pruebas del juego

Todas las pruebas realizadas en esta sección han sido desde la escena del juego con el respectivo uso de la cámara del dispositivo móvil o con la *web cam* del ordenador.

<i>Prueba 1 del juego</i>
Descripción de la prueba: desde la escena del juego, proyectar la cámara hacia la imagen del tablero.
Salida esperada: aparecen los objetos virtuales reflejados en la pantalla y empieza la cuenta atrás.
Resultado obtenido: éxito.

<i>Prueba 2 del juego</i>
Descripción de la prueba: en cualquier momento de la partida se deja de enfocar al tablero con la cámara, se esperan un determinado tiempo y se vuelve a enfocar al mismo.
Salida esperada: el juego se ha parado cuando la cámara ha perdido el objetivo y se ha reanudado cuando lo ha vuelto a encontrar.
Resultado obtenido: éxito.

<i>Prueba 3 del juego</i>
Descripción de la prueba: comprobar el correcto funcionamiento de la cuenta atrás.
Salida esperada: el temporizador ejecuta la cuenta atrás correctamente.
Resultado obtenido: éxito.

<i>Prueba 4 del juego</i>
Descripción de la prueba: comprobar el correcto funcionamiento de los cerrojos y animaciones durante la cuenta atrás.
Salida esperada: se activan o desactivan los cerrojos y las animaciones conforme a la lógica establecida mientras se ejecuta la cuenta atrás.
Resultado obtenido: éxito.

<i>Prueba 5 del juego</i>
Descripción de la prueba: comprobar que todos los cerrojos estén cerrados y las animaciones activadas cuando la cuenta atrás acabe.
Salida esperada: cerrojos cerrados y animaciones activadas cuando el temporizador marque 0.
Resultado obtenido: éxito.

<i>Prueba 6 del juego</i>
Descripción de la prueba: comprobar que el tiempo en segundos de cada ronda aparece correctamente en el temporizador con respecto a la dificultad seleccionada.
Salida esperada: nivel fácil = 8 segundos/ronda. Nivel medio = 6 segundos/ronda ; Nivel difícil = 4 segundos/ronda.
Resultado obtenido: éxito.

<i>Prueba 7 del juego</i>
Descripción de la prueba: comprobar que se activan 2 botones virtuales y sus cerrojos a cada ronda del temporizador. Los botones activados no pueden ser los mismos que los de la ronda anterior
Salida esperada: cerrojos y sus respectivos botones activados de 2 en 2 y diferentes con respecto a los de la ronda anterior.
Resultado obtenido: éxito.

<i>Prueba 8 del juego</i>
Descripción de la prueba: comprobar que los botones y cerrojos se desactivan cuando se acabe la ronda.
Salida esperada: los cerrojos se desactivan, se pone en rojo uno de sus aros superiores y se aumenta su número de fallos, .
Resultado obtenido: éxito.

<i>Prueba 9 del juego</i>
Descripción de la prueba: pulsar un boton virtual correctamente cuando está activado.
Salida esperada: se aumenta la puntuación obtenida, se cierra el cerrojo.
Resultado obtenido: éxito.

<i>Prueba 10 del juego</i>
Descripción de la prueba: pulsar un botón virtual incorrectamente cuando está activado, porque se ha pulsado a la vez algunos de los botones que hace frontera a ese cerrojo.
Salida esperada: la iluminación de ese cerrojo pasa a ser naranja.
Resultado obtenido: éxito.

<i>Prueba 11 del juego</i>
Descripción de la prueba: pulsar un botón virtual cuando no está activado.
Salida esperada: se aumenta el número de fallos en ese cerrojo y se pone en rojo uno de sus aros superiores.
Resultado obtenido: éxito.

<i>Prueba 12 del juego</i>
Descripción de la prueba: cuando acabe cada ronda comprobar si los cerrojos no se ha cerrado.
Salida esperada: si no se han cerrado, se aumenta el número de fallos en ese cerrojo y se pone en rojo uno de sus aros superiores.
Resultado obtenido: éxito.

<i>Prueba 13 del juego</i>
Descripción de la prueba: comprobar que se acaba la partida cuando llega a 4 el número de fallos de algún cerrojo.
Salida esperada: los botones , cerrojos y animaciones se desactivan y se muestra la puntuación en el panel informativo del tablero.
Resultado obtenido: éxito.

<i>Prueba 14 del juego</i>
Descripción de la prueba: comprobar que el cálculo de la puntuación obtenida ha sido el correcto.
Salida esperada: puntuación final = tiempo resistido + puntuación obtenida.
Resultado obtenido: éxito.

<i>Prueba 15 del juego</i>
Descripción de la prueba: comprobar que la información que aparece en el panel de información es la correcta durante todo el desarrollo de la partida.
Salida esperada: los mensajes, tiempo resistido y puntuación deben de irse actualizando correctamente en el panel de información.
Resultado obtenido: éxito.

Durante la ejecución de las pruebas correspondientes al juego, en una primera instancia hubo más fracasos que éxitos. Esto era de esperar, y más, debido a la cantidad de variables que utiliza el juego en paralelo y que se tienen que actualizar correctamente, por lo que se fueron depurando las líneas de código generadas buscando errores para, finalmente, conseguir el éxito en todas las pruebas.

8.3. Pruebas del *ranking*

Estas pruebas han sido diseñadas para comprobar el almacenamiento de las mejores puntuaciones, así como para comprobar la correcta visualización de las mismas en la escena del *ranking*.

<i>Prueba 1 del ranking</i>
Descripción de la prueba: al iniciar la aplicación por primera vez, pulsar la tecla <i>ranking</i> .
Salida esperada: todas las mejores puntuaciones deben aparecer inicializadas a 0.
Resultado obtenido: éxito.

<i>Prueba 2 del ranking</i>
Descripción de la prueba: jugar una partida en un nivel determinado, volver al menú y acceder al <i>ranking</i> .
Salida esperada: si la puntuación ha sido mejor que las que había en ese nivel deberá aparecer en la posición adecuada.
Resultado obtenido: éxito.

<i>Prueba 3 del ranking</i>
Descripción de la prueba: jugar varias partidas en cada nivel de dificultad y dirigirse al <i>ranking</i> .
Salida esperada: todas las mejores puntuaciones obtenidas por cada nivel de dificultad deberán aparecer ordenadas en la escena del <i>ranking</i> .
Resultado obtenido: éxito.

8.4. Pruebas de dispositivos

<i>Prueba 1 de dispositivos</i>
Descripción de la prueba: probar la aplicación en un dispositivo móvil Motorola MotoG 4G con un sistema operativo Android 5.1.
Salida esperada: el funcionamiento de la aplicación es el correcto.
Resultado obtenido: éxito.

<i>Prueba 2 de dispositivos</i>
Descripción de la prueba: probar la aplicación en un dispositivo móvil Google Nexus 5X con un sistema operativo Android 6.0.
Salida esperada: el funcionamiento de la aplicación es el correcto.
Resultado obtenido: éxito.

En un principio hubo un problema con el sistema operativo Android 6.0. y la versión de Vuforia que se estaba utilizando. La imagen proyectada por la cámara, aparecía invertida en la pantalla del dispositivo, o en otras palabras, se veía el mundo al revés. La nueva actualización que sacó Vuforia solventaba este fallo por lo que no hubo más que actualizar el software dentro del proyecto.

9. Conclusiones

En esta última sección, se va a realizar una comparativa entre la planificación que se estimó a comienzos del proyecto y la que finalmente ha sido, posibles líneas futuras para Control the Qí!, licencias y por último, una reflexión final.

9.1. Análisis entre planificación estimada y real

A continuación, se exponen los cambios entre la planificación de tareas estimada en el alcance del proyecto (Sección 2.7) y la que finalmente ha sido.

9.1.1. Cambios respecto al alcance

Los bloques de tareas (Figura 4) se han mantenido iguales, por lo que se van a explicar las diferencias internas de cada bloque.

Organización

En este bloque, no se ha realizado ningún cambio en las tareas. Finalmente el planteamiento de la aplicación requirió menos tiempo y se efectuaron menos reuniones con el director del TFG de las que se preveían, por lo que el cómputo real de tiempo en este bloque ha sido menor del estimado.

Aprendizaje

Al igual que en el bloque anterior, no se ha realizado ningún cambio sobre las tareas y el tiempo real ha sido algo menor del estimado.

Análisis y Diseño

En este bloque, se ha añadido una nueva tarea, rediseñar los diagramas de la aplicación, que se ha realizado entre el desarrollo inicial y final del juego, los cuales se muestran en el bloque de implementación y desarrollo.

<i>Rediseño de diagramas</i>
Paquete de trabajo: Análisis y diseño.
Duración: 14 horas.
Descripción: rediseñar los diagramas de la aplicación adaptándolos a la temática del juego.
Salidas/Entregables: Diagramas adaptados a la temática.
Recursos necesarios: Cacao.com.

Al añadir esta tarea, el tiempo real ha sido mayor que el estimado para este bloque.

Implementación y desarrollo

Se ha creado una nueva estructura para este bloque de tareas que aparece reflejada en la Figura 23. Como se puede observar, se han creado dos tareas principales que albergan todas las tareas definidas en el alcance (Sección 2.7.4) a excepción de la creación de las gafas.

Estas nuevas tareas representan, por un lado, el desarrollo inicial de la aplicación sin conocer la temática, y por el otro, una vez que se supo la temática, el desarrollo final. Cabe decir que el rediseño de diagramas del bloque anterior se realiza entre estas dos tareas. Las horas totales en este bloque han sido algo mayores a las estimadas.

<i>Desarrollo inicial del juego</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 40 horas.
Descripción: desarrollar la aplicación de una forma general, centrándose en la funcionalidad y no en la temática.
Salidas/Entregables: aplicación que se pueda jugar.
Recursos necesarios: Unity, MonoDevelop, SDK de Vuforia, Blender, GIMP 2 y los diagramas de la aplicación.

<i>Desarrollo final del juego</i>
Paquete de trabajo: Implementación y desarrollo.
Duración: 85 horas.
Descripción: adaptar la aplicación generada en el desarrollo inicial a la temática que se le ha dado al juego.
Salidas/Entregables: aplicación Control the Qí!
Recursos necesarios: Unity, MonoDevelop, SDK de Vuforia, Blender, GIMP 2 y el rediseño de los diagramas de la aplicación.

Pruebas

Al final, no se han realizado las pruebas de automatización debido a falta de tiempo, por lo que el tiempo real en este bloque de tareas ha sido menor del estimado.

Documentación

Respecto al documento de objetivos del proyecto se han mantenido las tareas, y el tiempo real ha sido igual que el estimado.

Por otro lado, al bloque de la memoria del proyecto se le ha añadido la tarea de revisión del documento que no se puso, en un primer momento, en el alcance (Sección 2.7.6).

<i>Revisión del documento</i>
Paquete de trabajo: Documentación. Duración: 20 horas.
Descripción: revisar el documento de objetivos del proyecto y la memoria en busca de errores y solventarlos.
Salidas/Entregables: documento revisado y listo para su entrega.
Recursos necesarios: TeXstudio, Cacao.com, LibreOffice, GanttProject, diccionarios online y la aplicación desarrollada.

Esta tarea ha supuesto un incremento en el número de horas invertidas frente a las estimadas para este bloque.

9.1.2. Reevaluación económica

Tal y como se observa en la Figura 24, se han trabajado 6 horas más de las estimadas en el alcance de este proyecto. Dado que se trata de un pequeño exceso de horas trabajadas, se ha decidido mantener la evaluación económica inicial (Sección 2.9) dando un total de **3577,03 €** por el completo desarrollo de la aplicación.

9.1.3. Rediseño de los diagramas de planificación

En las siguientes paginas aparece la estructura de descomposición de trabajo actualizada de este proyecto, junto con una comparativa de las horas estimadas en el alcance y las que han sido realmente, así como el diagrama Gantt final de este proyecto.

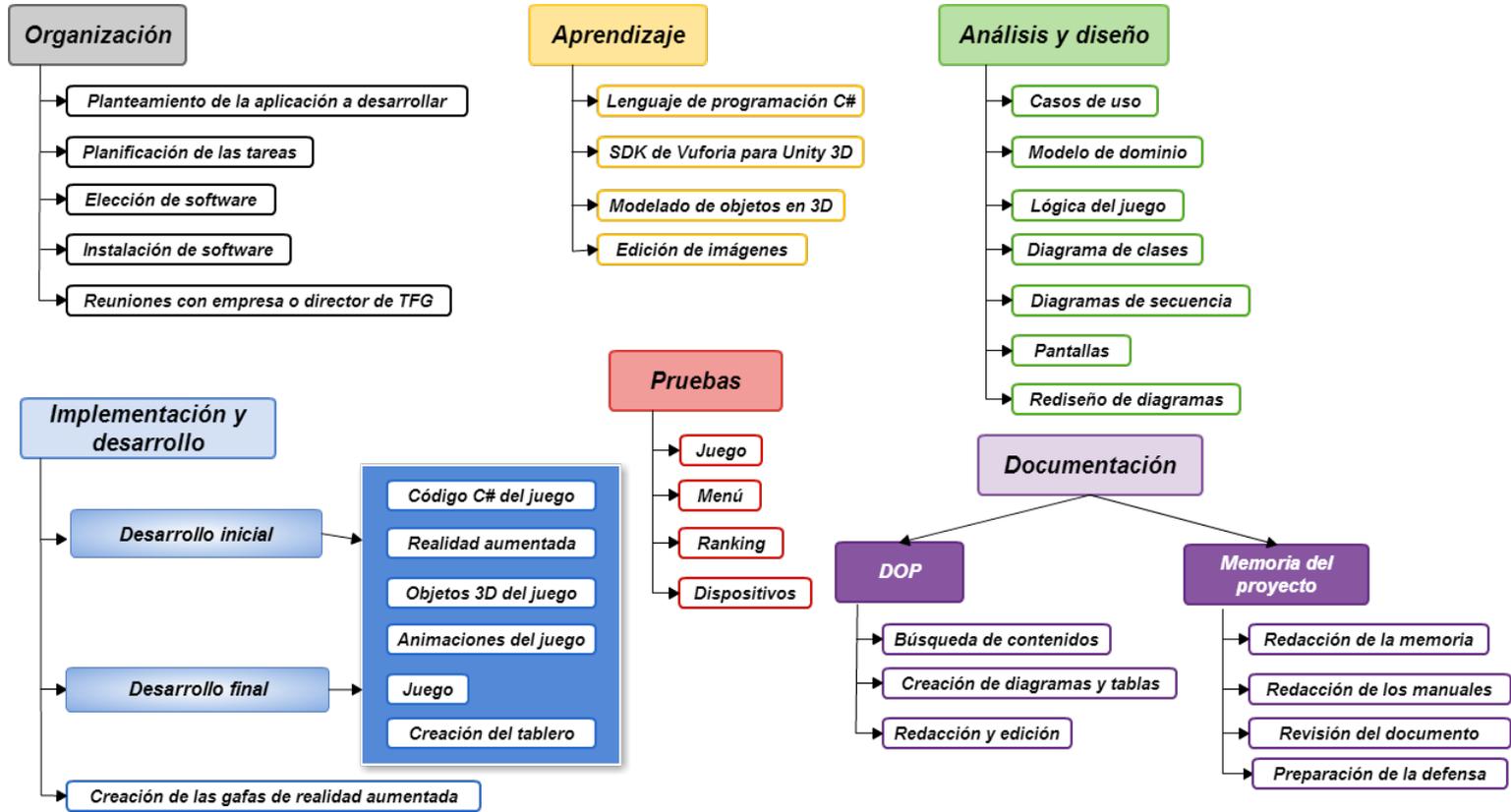


Figura 23: Diagrama final EDT por tareas

Planificación estimada	Duración estimada (horas)	Planificación real	Duración real (horas)	Diferencia (horas)
Control de Q1	437	Control de Q1	443	6
Organización	55	Organización	45	-10
Planteamiento de la aplicación	15	Planteamiento de la aplicación	10	
Planificación de las tareas	5	Planificación de las tareas	5	
Elección de software	5	Elección de software	5	
Instalación de software	5	Instalación de software	5	
Reuniones con empresa o director	25	Reuniones con empresa o director	20	
Aprendizaje	51	Aprendizaje	46	-5
Lenguaje de programación C#	20	Lenguaje de programación C#	17	
SDK de Vuforia para Unity 3D	15	SDK de Vuforia para Unity 3D	13	
Modelado de objetos en 3D	10	Modelado de objetos en 3D	10	
Edición de imágenes	6	Edición de imágenes	6	
Análisis y diseño	46	Análisis y diseño	60	14
Casos de uso	6	Casos de uso	6	
Modelo de dominio	6	Modelo de dominio	6	
Lógica del juego	6	Lógica del juego	6	
Diagrama de clases	6	Diagrama de clases	6	
Diagramas de secuencia	16	Diagramas de secuencia	16	
Pantallas	6	Pantallas	6	
		Rediseño de diagramas	14	
Implementación y desarrollo	130	Implementación y desarrollo	140	10
Código C# del juego	30	Desarrollo inicial del juego	40	
Realidad aumentada	20	Desarrollo final del juego	85	
Objetos 3D del juego	20	Gafas de realidad aumentada	15	
Animaciones del juego	15			
Juego	20			
Creación del tablero	10			
Gafas de realidad aumentada	15			
Pruebas	44	Pruebas	28	-16
Juego	10	Juego	10	
Menú	6	Menú	6	
Ranking	6	Ranking	6	
Dispositivos	6	Dispositivos	6	
Automatización	16			
Documentación	111	Documentación	124	13
<i>DOP</i>	<i>36</i>	<i>DOP</i>	<i>36</i>	<i>0</i>
Búsqueda de contenidos	6	Búsqueda de contenidos	6	
Creación de diagramas y tablas	10	Creación de diagramas y tablas	10	
Redacción y edición	20	Redacción y edición	20	
Memoria del proyecto	75	Memoria del proyecto	88	7
Redacción de la memoria	30	Redacción de la memoria	30	
Redacción de los manuales	15	Redacción de los manuales	8	
Preparación de la defensa	30	Revisión del documento	20	
		Preparación de la defensa	30	

Figura 24: Comparativa horas estimadas y reales

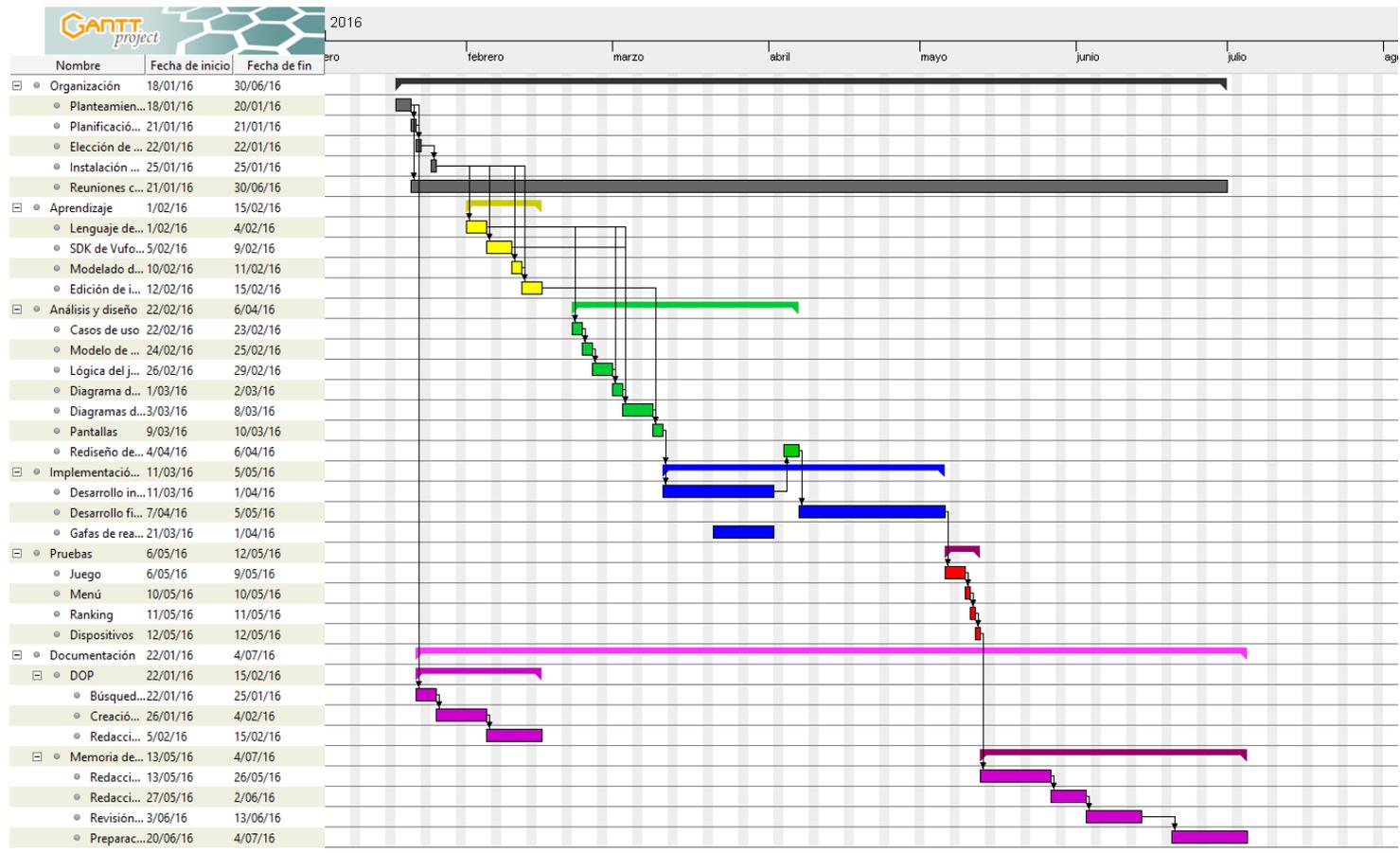


Figura 25: Diagrama Gantt final

9.2. Líneas futuras para Control the Qí!

En este apartado se van a reflejar las ideas surgidas para mejorar Control the Qí! en un futuro.

- **Combos:** el juego creado sólo permite sumar de 1 en 1 los puntos conforme el usuario va pulsando los cerrojos correctamente. Una mejora potencial para la partida sería premiar al usuario cuando toque consecutivamente un número de cerrojos bien. Este premio podría consistir en un aumento de la puntuación o en que se reduzca el número de fallos en algún cerrojo.
- **Crea tu tablero:** esta opción aparece deshabilitada en el menú principal de la aplicación. Consiste en que el usuario pueda crear su propio tablero mediante una foto captada por la cámara y que pueda jugar sobre esa imagen de la misma forma que con el tablero de Control the Qí!.
- **Añadir sonido:** se quiere añadir algún tipo de sonido cuando el usuario pulse correcta o incorrectamente algún cerrojo. Se intentará buscar sonidos con licencia abiertas.
- **Mejora de los objetos virtuales:** se podría mejorar el diseño de todos los objetos virtuales que aparecen en la partida. Para ello, serán requeridas alguna hora extra de estudio de Blender, dado que es una herramienta de modelado muy amplia y no se han podido estudiar todas sus funcionalidades durante el aprendizaje de este proyecto.
- **Gafas de realidad aumentada v2:** tras haber realizado las gafas de realidad aumentada para esta aplicación (ver Anexo B), se quieren realizar otras gafas mejorando el diseño con lo aprendido durante la creación de las primeras.

9.3. Licencias

En este apartado se verán, por un lado, las licencias que tienen aquellos recursos que se han utilizado para desarrollar la aplicación, y por el otro, el tipo de licencia que se le ha aplicado a Control the Qí!

9.3.1. Recursos utilizados

Algunas imágenes utilizadas para la construcción del tablero y del logotipo han sido descargadas de la pagina pixabay.com. La licencia de todas ellas es CC-Zero la cual no necesita reconocimiento del autor y es gratuita para fines comerciales. Las letras chinas han sido generadas con LibreOffice y se han convertido posteriormente a imágenes.

Todos los objetos virtuales han sido generados desde 0, así como todas las animaciones y demás recursos del juego, por lo que no se requiere hacer mención de ninguna licencia más.

9.3.2. Control the Qí!

La licencia sobre los derechos de uso que se ha establecido para Control the Qí! es de Creative Commons [7] y se ha elegido el tipo de licencia "**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa)**: *No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.*"

9.4. Reflexión personal

Para terminar con esta memoria de mi Trabajo de Fin de Grado me gustaría dedicar unas palabras de cómo ha sido mi experiencia universitaria a lo largo de estos años. Todavía me acuerdo como si fuera ayer de mi primer día en la universidad, entrando por la puerta de la antigua facultad de la Casilla, con los nervios clásicos de toda aquella persona que llega desde Bachiller a la universidad y que no sabe con certeza que le deparará el futuro, ni si ha elegido la carrera correcta.

Ahora que estoy redactando estas últimas líneas de la memoria sigo sin tener claro que me deparará el futuro, pero sí que sé con certeza que he elegido la carrera correcta.

Tras 5 años y un cambio de edificio de por medio, es imposible no echar la vista atrás y ver la cantidad ingente de horas dedicadas a trabajos, proyectos, competiciones y exámenes para llegar a este momento. Pero, por otro lado, me muestro satisfecho, dado que todas esas horas han valido para algo y si la vida es ir escalando montaña tras montaña, ese sacrificio de horas me ha valido para hacer cumbre en esta montaña.

Me gustaría agradecer a Jorge y Enara, de la empresa Creativitic, por haber podido contar con su ayuda para realizar este juego de realidad aumentada y también, por qué no decirlo, por haberme aguantado todos estos meses.

También me gustaría darle las gracias a mi director de proyecto Juanan Pereira por ayudarme en todo lo relacionado con la gestión de la aplicación y la redacción de la memoria. Siempre que he necesitado de su ayuda, me ha contestado a la mayor brevedad posible, ya sea por correo electrónico o yendo a sus tutorías, y esa disposición continua es algo que se agradece mucho.

Respecto a la realidad aumentada, veo un auténtico mundo de posibilidades además de ser una tecnología en plena expansión. Me encuentro muy ilusionado con el proyecto desarrollado, por lo que me gustaría incrementar mis conocimientos en un futuro sobre esta materia y poder, o bien mejorar esta aplicación, o crear otra totalmente diferente y novedosa.

Para concluir esta memoria, me gustaría agradecer a todas aquellas personas que han dedicado una parte de su tiempo a la lectura de este documento. Espero que no les hayan quedado dudas y que disfruten del juego de la misma forma que un servidor ha disfrutado desarrollándolo.

Bibliografía

- [1] Curso FOPU - Formacion docente del Profesorado Universitario - Realidad Aumentada - CreativiTIC Innova, 2016. Disponible en <https://www.dropbox.com/s/38c8jbhzbq60n4s2/Curso%20FOPU%20-%20CreativiTIC%202016.pdf?dl=0>.
- [2] Museum of London - Streetmuseum - Disponible en <http://www.museumoflondon.org.uk/Resources/app/you-are-here-app/home.html>.
- [3] Ministerio de Empleo y Seguridad Social - Boletín Oficial de Estado - Miercoles 10 Febrero 2016. Disponible en: <https://www.boe.es/boe/dias/2016/02/10/pdfs/BOE-A-2016-1290.pdf>.
- [4] Google Play Store - Acuerdo de distribución para desarrolladores de Google Play. Disponible en https://play.google.com/intl/ALL_es/about/developer-distribution-agreement.html.
- [5] Iñigo Allende - 2016 - Aprende: números y letras - Universidad del País Vasco - EUITI Bilbao <https://www.dropbox.com/s/1p8ra4hv940oeu8/Memoria.pdf?dl=0>.
- [6] A Spatial Augmented Reality Rehab System for Post-Stroke Hand Rehabilitation - University of California - Departments of Information and Computer Sciences - Department of Neurology <http://www.ics.uci.edu/~mkhademi/files/Publications/NextMed2013.pdf>.
- [7] Creative Commons - Tipos de licencia - <http://es.creativecommons.org/blog/licencias/>.

Anexos Control the Qí!

A. Anexo 1: Manual de usuario

Este es el manual de usuario correspondiente al juego Control the Qí! para dispositivos Android, donde se van a explicar los requisitos mínimos para poder jugar, los diferentes componentes del juego y el funcionamiento del mismo.

A.1. Requisitos mínimos

En la siguiente lista aparece lo necesario para poder jugar a la aplicación Control the Qí!

- **Dispositivo Android:** se debe disponer de un dispositivo con un sistema operativo Android 3.2 (*HoneyComb*) o superior, con cámara incorporada.
- **Aplicación Control the Qí!:** la aplicación es un tipo de archivo (.apk) y deberá ser instalada en el dispositivo Android.
- **Tablero:** El tablero de la aplicación se debe imprimir y colgar de una pared en una zona que no tenga muchos reflejos de luz. Con esto, se ayuda a la cámara del dispositivo a captar mejor el objetivo. En la siguiente imagen aparece el tablero de Control the Qí!.



- **Gafas de realidad aumentada:** Se ha adjuntado una guía acerca de cómo crear unas gafas de realidad aumentada personales utilizando en su mayor medida materiales reciclados. Esta guía aparece en el Anexo B. También existe la opción de comprar estas gafas en una tienda especializada.

A.2. Pantallas de Control the Qí!

En este apartado se mostrarán las diferentes pantallas del juego Control the Qí! y cómo moverse a través de ellas.

Al arrancar la aplicación desde el dispositivo Android nos aparecerá el menú principal del juego:



Si el usuario quiere jugar una partida deberá pulsar el botón Juego Survival que le aparece a la izquierda de la pantalla. Esto le llevará a una pantalla de seleccionar nivel:



Aquí tendrá que seleccionar una dificultad y se le lanzará la cámara del dispositivo para poder jugar una partida en esa dificultad. El funcionamiento del juego y la posición del cuerpo con respecto al tablero se detallan en las últimas secciones de este manual.

El usuario también puede volver al menú principal pulsando sobre el botón Volver situado en la esquina inferior izquierda de la pantalla.

Desde el menú principal si el usuario pulsa sobre el botón *Ranking* le aparecerá la siguiente pantalla.

Posición	Niveles		
	Fácil	Medio	Difícil
1	25	44	19
2	20	39	0
3	17	0	0

Volver

v1.0 By Jonatan Pérez

En esta pantalla se podrán ver las 3 mejores puntuaciones obtenidas por cada nivel de dificultad en este dispositivo. Dispone de un botón para volver al menú principal.

La opción Crea tu tablero del menú principal no esta disponible en esta versión del juego.

Por ultimo,si el usuario desea cerrar la aplicación, lo puede hacer desde el menú principal con el botón Salir.

A.3. Componentes y temática del juego

Los objetos virtuales que se han creado, al igual que el tablero, están relacionados con el Qí, el Yin Yang y los cinco elementos. Estos objetos se ven reflejados sobre el tablero de la siguiente forma.



Lo primero, se va a explicar de una forma simple el concepto del Qí y el Yin Yang. Se trata de una temática bastante amplia y rica en conocimientos por lo que se recomienda buscar información acerca de ello al margen de lo que aparezca en este manual.

El Qí (pronunciado Chi) es un término que ha sido utilizado durante miles de años en la cultura china, su traducción al castellano sería energía vital. Se puede asociar tanto a la energía que tiene todo lo que esta a nuestro alrededor, como a la energía propia de nuestro interior. Esta energía está directamente relacionada con el Yin Yang y puede variar dependiendo de este.

El Yin Yang representan la dualidad en todos los ámbitos de la vida y del universo. El Yin representa la oscuridad, el frío, el agua, las bajadas, la humedad, lo redondeado... El Yang, por el contrario, representa la luz, el calor, el cielo, lo que crece, lo geométrico... Son totalmente opuestos, pero no pueden existir el uno sin el otro. Además, siempre se intentan compensar mutuamente para mantener el equilibrio.

El Yin Yang tienen una relación directa con los 5 elementos de la naturaleza, representados en forma de cerrojos en el juego. Al igual que pasaba con el Qí, el Yin Yang puede descompensarse por cambios o alteraciones en estos 5 elementos de los cuales se hablará más adelante.

Una vez vista esta breve explicación, se procede a detallar cada componente del juego:

- **Panel Información:** aparece en la parte superior del tablero. En él se muestra información relativa a la partida, el tiempo acumulado y la puntuación que lleva obteniendo el usuario en dicha partida.



- **Temporizador:** aparece sobre la imagen del Yin Yang. Se usa para proporcionar al usuario el tiempo que tiene para cerrar los cerrojos que están abiertos y como contador de la cuenta atrás inicial de cada partida.



- **Botones Reiniciar/Salir:** aparecen en la parte de arriba del tablero, a la izquierda y a la derecha respectivamente. Si el usuario pone la mano sobre esa zona del tablero para pulsar estos botones virtuales se reiniciará la partida o se volverá al menú principal.

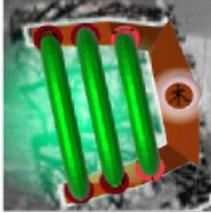
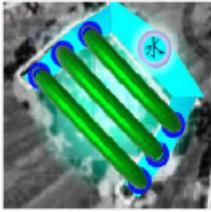
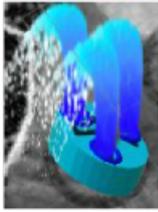
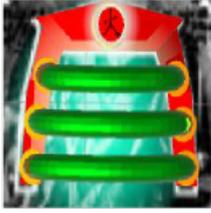
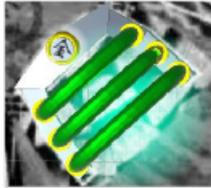
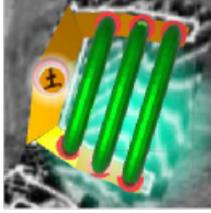


- **Cerrojos y animaciones:** representan los 5 elementos de la naturaleza: la madera, el agua, el fuego, el metal y la tierra. Estos elementos, y su relación con el Yin Yang y el Qí, están asociados a muchos aspectos de la cultura china y se les considera la base de disciplinas como el yoga, diferentes artes marciales o por su aplicación en la medicina tradicional china, métodos los cuales se siguen usando actualmente.

Cada elemento, en este juego, viene representado por un cerrojo y 3 animaciones situadas sobre una misma base. A continuación se detalla cada elemento:

1. **Madera:** asociado a la primavera, a la naturaleza de los paisajes, a la gentileza humana y a la arquitectura de objetos verticales entre otros aspectos. La animación que representa este elemento es un árbol.
2. **Agua:** asociado al invierno, a las emociones de las personas junto con su control y a todo lo relacionado con fenómenos fluviales. Su animación es un géiser.
3. **Fuego:** asociado al verano, al honor y la pasión de las personas, y en general, a símbolos relacionados con el Sol. Se representa mediante una llama.
4. **Metal:** asociado al otoño, a la organización y madurez en las personas y a todos los objetos o acciones que tengan algún tipo de comunicación. Esto se debe a que el metal es un tipo de material conductor. Sus animaciones son: unas barras de oro, plata y bronce.
5. **Tierra:** representa el ciclo entre estaciones, para garantizar el paso del tiempo, se encarga de ayudar y apoyar a los otros 4 elementos con el fin de conseguir el equilibrio. En las personas, representa la sabiduría. Su animación es un tornado de arena.

Todas las imágenes asociadas a estos elementos aparecen en la siguiente figura:

	Cerrojo	Animaciones
Madera		
Agua		
Fuego		
Metal		
Tierra		

A.4. Cómo jugar una partida

Una vez que en la pantalla del dispositivo Android aparezca lo que está enfocando la cámara, tras haber seleccionado el nivel de dificultad, se debe dirigir el objetivo de la cámara hacia el tablero de Control the Qí!

Cuando la cámara localice el tablero, aparecerán todos los objetos virtuales y empezará la cuenta atrás desde el temporizador.



Una vez que el temporizador llegue a 0, comenzará la partida. El juego consiste en aguantar el máximo tiempo posible pulsando los botones virtuales que se van activando en el tablero.



Los botones se activan de dos en dos por cada ronda. La duración de cada ronda depende del nivel de dificultad seleccionado previamente.

- **Nivel fácil:** 8 segundos/ronda.
- **Nivel medio:** 6 segundos/ronda.
- **Nivel difícil:** 4 segundos/ronda.

Una vez que se acaba cada ronda, los botones activados que no se han pulsado aumentan en 1 sus fallos y se ilumina 1 de sus anillos en color rojo representando el fallo. También se desactivará una de sus animaciones.

A la hora de pulsar los botones situados en el tablero pueden darse 3 situaciones distintas:

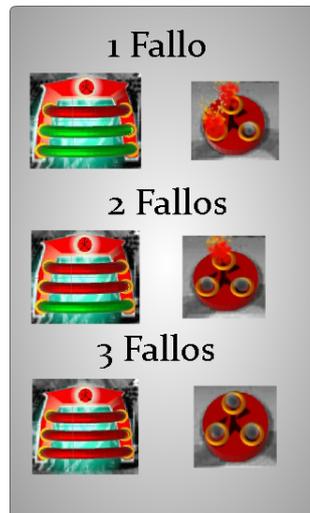
- **Cerrojo activado y pulsado correctamente:** el usuario ha puesto la mano correctamente sobre el lugar del tablero donde se encontraba el cerrojo. Se le aumenta en 1 la puntuación y se cierra el cerrojo.
- **Cerrojo activado y pulsado incorrectamente:** el usuario ha introducido la mano en el cerrojo tocando sus límites mientras estaba activado. El movimiento no se considera válido, se iluminará el cerrojo en naranja y el usuario tendrá que introducir la mano de nuevo.
- **Cerrojo desactivado y pulsado:** el usuario ha introducido la mano en el cerrojo cuando no estaba activado. Se le aumentará el número de fallos a este cerrojo y se le desactivará una animación.

Cuando el número de fallos en cualquier cerrojo sea 4, el juego se acabará, desactivándose todos los cerrojos y animaciones. Se le mostrará al usuario la puntuación final obtenida que es la suma de los puntos obtenidos más el tiempo resistido.



El usuario puede reiniciar la partida o volver al menú pulsando los respectivos botones virtuales.

Los fallos en los cerrojos y en las animaciones quedan representados de la siguiente forma:



A.5. Posición del cuerpo con respecto al tablero

En esta última sección del manual de usuario, se darán las guías de cómo situar el cuerpo con respecto al tablero y una serie de ejercicios para realizar mientras se juega una partida.

Posicionamiento del tablero

Para garantizar un correcto funcionamiento del juego, se tiene que buscar una pared con buena iluminación para colgar el tablero. Cuantas menos sombras haya, mejor captará la cámara del dispositivo los diferentes elementos del tablero.

La referencia para colgar o fijar el tablero en una pared es el símbolo del Yin Yang del mismo. El usuario debe situar este símbolo en torno a la altura de los ojos y de la nariz. La distancia que tiene que dejar el usuario con respecto al tablero debe ser un poco mayor que la longitud de su brazo.

El usuario puede jugar tanto de pies como sentado en una silla. Ambas posiciones aparecen en la siguiente Figura.



Posición de las manos y los brazos

La mejor posición de las manos para jugar a Control the Qí! es con las palmas abiertas. El usuario puede elegir poner hacia delante o hacia atrás las manos dependiendo de como se encuentre más cómodo.



La distancia con respecto al tablero se ha especificado con los brazos completamente estirados hacia delante. El usuario, si se siente incómodo en esta posición, puede jugar con los brazos arqueados por lo que tendrá que reducir algo su distancia con respecto al tablero.



A continuación, se muestran los músculos implicados dependiendo de la posición del brazo y de la mano:

- **Brazos estirados:** deltoides, tríceps y vasto interno.
- **Brazos arqueados:** bíceps y tríceps.
- **Palmas hacia delante:** flexores de la mano.
- **Palmas hacia atrás:** extensores de la mano.

Si el usuario utiliza el tablero de tamaño A2, no tendrá problemas para poder introducir la mano en él. Por el otro lado, si se utiliza el tamaño A4, es probable que tenga que jugar utilizando únicamente los dedos índice y corazón de cada mano.

Ejercicios de piernas a realizar mientras se juega

Control the Qí! es una aplicación pensada para ejercitar brazos, manos y hombros. Pero hay una serie de ejercicios específicos de piernas que se pueden realizar a la par que se juega y así, conseguir un entrenamiento de cuerpo completo

- **Pata coja:** el usuario, de pies, se debe apoyar sólo en una pierna mientras juega una partida. Los músculos implicados son el cuádriceps, gemelo, glúteo y músculo estabilizador del tobillo de la pierna que se apoya.



- **Sentadillas:** a cada final de ronda, el usuario deberá realizar una sentadilla y volver a su posición original. Se recomienda usar este ejercicio con nivel fácil para disponer del máximo tiempo posible por ronda. Se ejercitan isquiotibiales, cuádriceps y glúteos.



- **Mantener sentadilla:** el usuario deberá situar la altura del tablero como si fuera a jugar sentado, pero con la diferencia que no tendrá una silla en la que sentarse. Deberá aguantar la posición de sentadilla mientras juega la partida. Con este ejercicio se ejercitan en particular los isquiotibiales y glúteos.



B. Anexo 2: Gafas de realidad aumentada: BilboFranky

Para construir estas gafas de realidad aumentada se han intentado utilizar, en la medida de lo posible, materiales reciclados para su elaboración.

El nombre de BilboFranky surgió como una fusión de conceptos. Por un lado, la parte de Bilbo es porque se utilizan noticias e imágenes del Athletic Club de Bilbao y Bilbao Basket para su decoración. Estas noticias proceden de periódicos gratuitos repartidos a diario en el metro. Y por el otro, el nombre Franky es atribuido a que estas gafas son el primer prototipo creado y, durante la creación de las mismas, se ha ido experimentado con ellas buscando y añadiendo diversas mejoras.

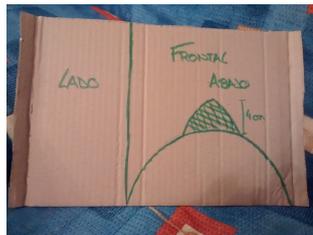
B.1. Materiales

La lista de materiales para la elaboración de estas gafas se muestra a continuación:

- Cartón (1 caja tamaño mediano/grande).
- Periódicos o papel higiénico.
- Espuma de embalar.
- Pinceles.
- Cola blanca para madera.
- Agua.
- Rotuladores y pinturas.
- Gomas de pelo.
- Grapadora.
- Tiras elásticas con velcro.

B.2. Instrucciones

1. En primer lugar, se debe medir el dispositivo móvil o tablet que se quiera usar para calcular las dimensiones de las gafas. En este caso, se ha diseñado pensando en dispositivos móviles de hasta 5 pulgadas. La estructura principal de las gafas es una altura de 10,5 cm, una anchura de 18 cm y una profundidad de 20 cm. Esta estructura se ha creado en dos partes independientes, tal y como se puede ver en la siguiente figura.



Además, se han añadido dos solapas laterales a cada lado para así, facilitar montar ambas estructuras. También se han marcado las zonas donde van a ir la frente y la nariz del usuario dejando un espacio de 10 cm entre los ojos del usuario y la pantalla del dispositivo.

2. Recortar la zona de la frente y de la nariz, grapar la espuma de embalar alrededor de estas zonas y recortar un poco las solapas laterales para facilitar la unión. El resultado sería:



3. Se encolan las solapas laterales y se unen ambas piezas dando como resultado:



4. Ahora se va a crear la pieza que va a albergar el dispositivo móvil. Se han mantenido las medidas de la pieza anterior (10,5 y 18 cm) para crear esta pieza. Además, se le ha añadido un pequeño muro de contención encolando pequeños trozos de cartón uno encima de otro en la parte de abajo. Esto ayuda a que el móvil se mantenga estable cuando el usuario tenga las gafas puestas.



La zona blanca central albergará el móvil y se ha dejado 1 cm alrededor de ella para que se pueda unir con la otra pieza. A esta pieza se le ha creado un agujero en medio para la cámara y se han añadido unas gomás del pelo en los laterales pegándolas con el muro de contención creado.

5. Utilizando una mezcla de 40% cola blanca y 60% agua, con la ayuda de un pincel, se le irán aplicando capas con trozos de papel higiénico o periódicos a las dos piezas con el objetivo de endurecerlas. En este caso concreto, se han utilizado ambas cosas, para que se puedan apreciar ambas técnicas.

6. Una vez que se ha secado, añadir los velcros de sujeción y pintar al gusto, utilizando rotuladores, pintura acrílica o el material que se desee.
7. A la hora de realizar la sujeción para la cabeza se ha añadido una estructura de papel sobre las cintas elásticas. Con esto se consigue rigidez para que las gafas se adapten correctamente al cráneo del usuario. El resultado final se puede ver a continuación:

