



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
(Mención Computación)

Creación de una plataforma para la gamificación de una asignatura

Autor:
D. Juan Rodríguez Sanz



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
(Mención Computación)

Creación de una plataforma para la gamificación de una asignatura

Autor:

D. Juan Rodríguez Sanz

Tutores:

**Dña. Alma María Pisabarro Marrón
Dr. Carlos Enrique Vivaracho Pascual**

Resumen

Este proyecto se ha realizado como Trabajo de Fin de Grado de Ingeniería Informática con mención en Computación de la Universidad de Valladolid.

En este documento se describe el desarrollo de una plataforma web para albergar minijuegos a medida, con el objetivo de gamificar una asignatura, y que estará disponible en múltiples navegadores. La plataforma gestionara el control de acceso de los usuarios con distintos roles de entrada, guardara información de todas las actividades realizadas por los usuarios, permitirá subir, cargar, alojar y ejecutar minijuegos ajenos a esta aplicación, guardará el estado de estos para cada usuario permitiendo continuar una partida empezada, entre otras funcionalidades.

Este proyecto se ha realizado siguiendo el el desarrollo de software en cascada, que consta de las fases de planificación, desarrollo y mantenimiento. En la fase de planificación se ha realizado el planteamiento inicial, propuestas y en definitiva la toma de requisitos y análisis. En la fase de desarrollo se especifica el diseño de la aplicación, con tareas como la especificación de casos de uso o el desarrollo de la arquitectura, la implementación del código de la aplicación, teniendo en cuenta el análisis previo y las pruebas. Por ultimo en la fase de mantenimiento se han desarrollado los manuales.

Tabla de contenidos

Resumen	5
Introducción	15
1.1 Ventajas e inconvenientes de la gamificación de una asignatura.....	16
1.1.1 Ventajas.....	16
1.1.2 Inconvenientes.....	16
1.2 Nomenclatura.....	16
1.3 Objetivo general.....	17
Conceptos previos	19
2.1 Unity.....	19
2.1.1 ¿Por que Unity?.....	19
2.1.2 Ventajas.....	20
2.1.3 Inconvenientes.....	20
2.2 WebGL.....	20
Análisis	23
3.1 Descripción de los actores y roles de la aplicación.....	23
3.1.1 Usuario.....	23
3.1.2 Desarrollador.....	23
2.1.3 Administrador.....	23
3.2 Especificación de requisitos.....	24
3.2.1 Requisitos funcionales.....	24
3.2.2 Requisitos no funcionales.....	25
3.2.3 Requisitos de información.....	25
3.2.4 Casos de uso.....	26
3.3 Casos de uso.....	27
3.3.1 Diagrama de casos de uso.....	27
3.3.2 Especificación de los casos de uso.....	28
3.4 Modelo de dominio.....	60
Plan de proyecto	61
4.1 Plan de proceso.....	61
4.1.1 Funcionamiento.....	61
4.1.2 Ventajas.....	62
4.1.3 Inconvenientes.....	63
4.2 Plan de gestión de riesgos.....	63
4.2.1 Análisis de riesgos.....	63
4.3 Gestión de proceso.....	66
4.3.1 Plan de trabajo.....	66
4.3.2 Seguimiento del plan de trabajo.....	76
4.3.3 Desviaciones.....	80
4.4 Presupuesto.....	80
Diseño	83
5.1 Patrón MVC.....	83
5.2 Arquitectura del sistema.....	84
5.2.1 Arquitectura lógica.....	84
5.2.2 Despliegue.....	84
5.2.3 Arquitectura cliente.....	85
5.2.4 Arquitectura servidor.....	85

5.3 Modelo de datos.....	86
5.3.1 Modelo entidad-relación.....	86
5.3.2 Modelo relacional.....	86
5.4 Realización en diseño de casos de uso.....	86
5.4.1 CU-01: Identificarse.....	87
5.4.2 CU-02 Cerrar sesión.....	88
5.4.3 CU-03 Mostrar puntuación.....	88
5.4.4 CU-05 Mostrar mundos.....	88
5.4.5 CU-07 Jugar minijuego.....	90
5.4.6 CU-08 Añadir minijuego.....	91
5.4.7. CU-12 Eliminar minijuego.....	92
Implementación	93
6.1 Entorno de desarrollo.....	93
6.2 Implementación de la Base de Datos.....	93
6.3 Copias de seguridad.....	93
6.4 Organización del código.....	93
6.5 Integración de elementos.....	93
6.5.1 Configuración del servidor.....	93
6.5.2 Comunicación Web – Unity.....	94
6.5.3 Otro problema.....	99
Plan de pruebas y evaluación	101
7.1 Casos de prueba.....	101
7.1.1 Usuarios.....	101
7.1.2 Mundos y minijuegos.....	103
7.1.3 Jugar.....	106
7.2 Resultado de las pruebas.....	106
7.2.1 Google Chrome.....	107
7.2.2 Firefox.....	108
7.2.3 Microsoft Edge.....	109
Conclusiones y trabajo futuro	111
8.1 Conclusiones.....	111
8.2 Trabajo futuro.....	111
Manual de usuario	116

Índice de figuras

Figura 1: Diagrama de casos de uso.....	27
Figura 2: Modelo de dominio.....	60
Figura 3: Desarrollo en cascada.....	61
Figura 4: Desarrollo en cascada agrupado.....	62
Figura 5: Arquitectura lógica.....	84
Figura 6: Despliegue.....	84
Figura 7: Arquitectura cliente.....	85
Figura 8: Arquitectura servidor.....	85
Figura 9: Modelo entidad-relacion.....	86
Figura 10: Modelo relacional.....	86
Figura 11: Caso de uso identificarse.....	87
Figura 12: Caso de uso cerrar sesión.....	88
Figura 13: Mostrar puntuación.....	88
Figura 14: Mostrar mundos.....	89
Figura 15: Jugar minijuego.....	90
Figura 16: Añadir minijuego.....	91
Figura 17: Eliminar minijuego.....	92
Figura 18: Comunicación Web – Unity, planteamiento inicial.....	95
Figura 19: Comunicación Web – Unity, primera solución.....	96
Figura 20: Comunicación Web – Unity, solución final.....	98

Índice de tablas

Tabla 1: Requisitos funcionales.....	24
Tabla 2: Requisitos no funcionales.....	25
Tabla 3: Requisitos de información.....	25
Tabla 4: Casos de uso.....	26
Tabla 5: Descripción del caso de uso: Identificarse.....	28
Tabla 6: Descripción del caso de uso: Cerrar sesión.....	29
Tabla 7: Descripción del caso de uso: Mostrar puntuaciones.....	30
Tabla 8: Descripción del caso de uso: Mostrar puntuaciones de grupos.....	30
Tabla 9: Descripción del caso de uso: Mostrar mundos.....	31
Tabla 10: Descripción del caso de uso: Mostrar minijuegos.....	32
Tabla 11: Descripción del caso de uso: Jugar a un minijuego.....	33
Tabla 12: Descripción del caso de uso: Añadir minijuego.....	34
Tabla 13: Descripción del caso de uso: Añadir proyecto al minijuego.....	35
Tabla 14: Descripción del caso de uso: Revisar minijuegos.....	36
Tabla 15: Descripción del caso de uso: Modificar minijuego.....	37
Tabla 16: Descripción del caso de uso: Modificar proyecto del minijuego.....	38
Tabla 17: Descripción del caso de uso: Eliminar minijuego.....	39
Tabla 18: Descripción del caso de uso: Revisar mundos y minijuegos.....	40
Tabla 19: Descripción del caso de uso: Hacer visible minijuego.....	41
Tabla 20: Descripción del caso de uso: Ocultar minijuego.....	42
Tabla 21: Descripción del caso de uso: Abrir minijuego.....	43
Tabla 22: Descripción del caso de uso: Cerrar minijuego.....	44
Tabla 23: Descripción del caso de uso: Cambiar posición minijuego.....	45
Tabla 24: Descripción del caso de uso: Añadir mundo.....	46
Tabla 25: Descripción del caso de uso: Modificar mundo.....	47
Tabla 26: Descripción del caso de uso: Eliminar mundo.....	48
Tabla 27: Descripción del caso de uso: Hacer visible mundo.....	49
Tabla 28: Descripción del caso de uso: Ocultar mundo.....	50
Tabla 29: Descripción del caso de uso: Abrir mundo.....	51
Tabla 30: Descripción del caso de uso: Cerrar mundo.....	52
Tabla 31: Descripción del caso de uso: Cambiar posición mundo.....	53
Tabla 32: Descripción del caso de uso: Añadir usuario.....	54
Tabla 33: Descripción del caso de uso: Añadir lista usuarios.....	55
Tabla 34: Descripción del caso de uso: Revisar usuarios.....	56
Tabla 35: Descripción del caso de uso: Modificar usuario.....	57
Tabla 36: Descripción del caso de uso: Eliminar usuario.....	58
Tabla 37: Descripción del caso de uso: Eliminar todos los usuarios.....	59
Tabla 38: Disponibilidad del desarrollador.....	63
Tabla 39: Enfermedad.....	64
Tabla 40: Pérdida de datos o/y documentación.....	64
Tabla 41: Fallo en la máquina.....	64
Tabla 42: Diseño mal elaborado.....	64
Tabla 43: Conocimiento insuficiente sobre las tecnologías.....	65
Tabla 44: Cambios en los requisitos.....	65
Tabla 45: Fallo en la conexión a Internet.....	65
Tabla 46: Calendarización fase de inicio.....	66
Tabla 47: Inicio de la fase de inicio.....	67
Tabla 48: Planteamiento inicial del problema.....	67
Tabla 49: Toma de requisitos.....	67

Tabla 50: Análisis de viabilidad.....	67
Tabla 51: Definición de las tareas a desarrollar.....	67
Tabla 52: Calendarización.....	67
Tabla 53: Gestión de riesgos.....	68
Tabla 54: Memoria.....	68
Tabla 55: Presupuesto.....	68
Tabla 56: Documento de costes.....	68
Tabla 57: Preparar el entorno y herramientas.....	68
Tabla 58: Finalización de la fase de inicio.....	68
Tabla 59: Calendarización fase de análisis.....	69
Tabla 60: Inicio de la fase de análisis.....	69
Tabla 61: Modelo de casos de uso.....	69
Tabla 62: Especificación de casos de uso.....	69
Tabla 63: Modelo de dominio.....	70
Tabla 64: Investigación de WebGL.....	70
Tabla 65: Investigación de Unity.....	70
Tabla 66: Investigación de PHP.....	70
Tabla 67: Diseño de las vistas UI.....	70
Tabla 68: Finalización de la fase de <i>análisis</i>	70
Tabla 69: Calendarización fase de diseño.....	71
Tabla 70: Inicio de la fase de diseño.....	71
Tabla 71: Investigación del patrón de diseño de la aplicación.....	71
Tabla 72: Arquitectura de la aplicación.....	71
Tabla 73: Diseño de la base de datos.....	71
Tabla 74: Finalización de la fase de diseño.....	72
Tabla 75: Calendarización fase de implementación.....	72
Tabla 76: Inicio de la fase de implementación.....	72
Tabla 77: Configuración del servidor.....	72
Tabla 78: Configuración de Unity.....	73
Tabla 79: Implementación de las vistas de Unity.....	73
Tabla 80: Implementación de las funcionalidades de las vistas de Unity.....	73
Tabla 81: Implementación de las vistas de la Web.....	73
Tabla 82: Implementación de las funcionalidades de las vistas de Web.....	73
Tabla 83: Implementación de la conexión Unity – Web.....	73
Tabla 84: Finalización de la fase de implementación.....	73
Tabla 85: Calendarización fase de pruebas.....	74
Tabla 86: Inicio de la fase de pruebas.....	74
Tabla 87: Pruebas individuales de funcionalidad de Unity.....	74
Tabla 88: Pruebas individuales de funcionalidad de la Web.....	74
Tabla 89: Pruebas de la aplicación completa.....	75
Tabla 90: Pruebas de carga y descarga de datos.....	75
Tabla 91: Pruebas de velocidad.....	75
Tabla 92: Elaborar el manual de usuario.....	75
Tabla 93: Finalización de la fase de pruebas.....	75
Tabla 94: Resumen del plan del proyecto.....	75
Tabla 95: Calendarización real de la fase de inicio.....	76
Tabla 96: Calendarización real de la fase de análisis.....	77
Tabla 97: Calendarización real de la fase de diseño.....	77
Tabla 98: Calendarización real de la fase de implementación.....	78
Tabla 99: Calendarización real de la fase de pruebas.....	79
Tabla 100: Resumen del plan del proyecto.....	79
Tabla 101: Plan de costes del proyecto.....	81
Tabla 102: Problema con los servicios.....	94
Tabla 103: Problema con imágenes y archivos pesados.....	94
Tabla 104: Problema con los minijuegos en la comunicación Web – Unity.....	96

Tabla 105: Problema con el controlador y carga de un minijuego.....	97
Tabla 106: Problema con la comunicación Unity – Web.....	98
Tabla 107: Problema con la carga de imágenes de forma dinámica.....	99
Tabla 108: Añadir un nuevo usuario.....	101
Tabla 109: Añadir lista de usuarios.....	102
Tabla 110: Modificar un usuario.....	102
Tabla 111: Eliminar un usuario.....	103
Tabla 112: Añadir un nuevo mundo.....	103
Tabla 113: Modificar un mundo.....	103
Tabla 114: Añadir un nuevo minijuego a un mundo.....	104
Tabla 115: Añadir un proyecto a un minijuego.....	104
Tabla 116: Modificar un proyecto de un minijuego.....	104
Tabla 117: Modificar un minijuego.....	105
Tabla 118: Eliminar un minijuego.....	105
Tabla 119: Eliminar un mundo.....	105
Tabla 120: Iniciar un minijuego.....	106
Tabla 121: Terminar un minijuego.....	106
Tabla 122: Salir de un minijuego.....	106
Tabla 123: Resultado de las pruebas en Google Chrome.....	107
Tabla 124: Resultado de las pruebas en Firefox.....	108
Tabla 125: Resultado de las pruebas en Microsoft Edge.....	109

Capítulo 1

Introducción

Cuando somos jóvenes, al jugar ensayamos la vida adulta, aprendiendo que hay reglas, colaboración, competición, roles, premios o peligro en un entorno seguro donde se puede fallar. También se dice que el juego tiene el poder de liberar tensiones. En la antigua Grecia los juegos olímpicos era una forma de hacer la guerra pero sin matarse [1].

El sistema de enseñanza tradicional es lineal. Eso implica que los libros de texto se empiezan a leer por un sitio, se acaban por otro y se debe pasar por todo lo que hay entre medias: toda la información es igual de importante. Este tipo de enseñanza estaba pensado para una época en la que la información era escasa y generalmente el libro de texto era el único material didáctico disponible [2]. Actualmente existe una gran cantidad de información fácilmente disponible tanto para los docentes como para los alumnos, esto permite flexibilizar los métodos de aprendizaje. Se podrían diseñar "rutas" de aprendizaje individualizadas sin tener que recurrir al sistema lineal. Los juegos son abiertos, tienen un principio y un final pero entre medias tú eliges con qué personajes hablas, no tienes por qué realizar todos los retos,... tú decides que camino sigues. Este mecanismo utilizado por los juegos se asemeja más a como nos movemos en una época con una enorme cantidad de información, y puede acomodarse a las necesidades de los métodos de enseñanza proporcionando grandes beneficios.

Los juegos poseen características que hacen que sean divertidos, lo que los convierte en una poderosa herramienta para aumentar la motivación de los estudiantes cuando se utilizan en un entorno educativo, incluida la enseñanza universitaria. Con esto en mente nos propusimos la gamificación de la asignatura de *Fundamentos de Programación*, planteando distintos "minijuegos" asociados con los distintos temas de la asignatura. Con el concepto "minijuego" nos estamos refiriendo a pequeños juegos, simples y fáciles de implementar, en contraposición a lo que suele ser habitual que es la gamificación basada en un único juego. Como otros juegos deben contener un objetivo, un marco de tiempo definido, un marcador y un premio. Se busca la creación de pequeñas piezas (minijuegos), fáciles de sustituir y modificar si las circunstancias cambian, logrando un entorno de gamificación más flexible y adaptable. Lo importante dentro de la gamificación basada en minijuegos es que el alumno aprenda a su ritmo, el minijuego aumenta la motivación en el aprendizaje, ayuda al alumno a adquirir conocimientos de una manera atractiva y contribuye al desarrollo de competencias [3]. El objetivo final es, principalmente, permitir que los usuarios puedan aprender a programar de forma más divertida y motivadora.

El problema de la técnica de gamificación que se propone, es que se necesita algo que "ensamble" los distintos minijuegos. Esto es lo que se propone en el presente proyecto. Para dar libertad de desarrollo, cada minijuego debe ser completamente independiente del resto. La plataforma será la que proporcione la sensación de unidad al entorno de juego. De igual manera debe gestionar todo lo referente a puntuaciones, usuarios y su evolución en el juego. Para un uso más flexible y amplio (se piensa que su utilización pueda ir más allá del de la asignatura para la que se ha planteado), se pretende que todo lo referente a minijuegos sea editable: se puedan añadir,

modificar, eliminar, ocultar y abrir o cerrar minijuegos según se necesite. Por último debe presentar un interfaz de comunicación con los minijuegos estándar.

Para el desarrollo de minijuegos se ha elegido la herramienta **Unity**. Es una herramienta integrada de desarrollo de videojuegos muy utilizada actualmente. Aparte de su amplio uso y la gran documentación disponible, se puede usar de manera gratuita para fines no lucrativos. Aunque para el desarrollo de este proyecto se han usado, como se verá, más plataformas de desarrollo, la elección de **Unity** ha sido un punto clave, ya que condiciona el entorno de desarrollo de los minijuegos.

1.1 Ventajas e inconvenientes de la gamificación de una asignatura

1.1.1 Ventajas

En el siguiente listado aparecen las principales ventajas de la ramificación de los juegos en el ámbito educativo:

- Aumenta la motivación de los alumnos.
- El protagonista es el alumno.
- Permite crear diferentes ritmos.
- Fallar no es malo.
- *Feedback* en tiempo real.
- Los docentes también reciben *feedback*.
- Los juegos favorecen la sociabilización.
- Los juegos favorecen la sana competencia.
- Jugar desarrolla la creatividad.
- Jugar es divertido.

1.1.2 Inconvenientes

En el siguiente listado aparecen los principales inconvenientes de la ramificación de los juegos en el ámbito educativo:

- La visión general del tema quede reducida a puntos demasiados concretos.
- Puede focalizar al jugador en ganar el juego y no en aprender.
- Tentación de hacer trampas.
- Puede intensificar la desmotivación ante posibles dificultades.
- Pueden ser adictivos.
- Consumen demasiado tiempo.
- Inconvenientes específicos de cada juego [4].

1.2 Nomenclatura

A lo largo de este documento se usarán distintas palabras que sin un contexto previo a su significado puede dificultar la comprensión de las mismas. Estas son:

- Minijuego: Pequeños juegos, simples y fáciles de implementar, en contraposición a lo que suele ser habitual que es la gamificación basada en un único juego. Como otros juegos deben contener un objetivo, un marco de tiempo definido, un marcador y un premio.

- Proyecto del minijuego: Es el código que se asocia a un minijuego.
- Mundo: Un mundo se podría explicar como si fuera un tema de una asignatura, en el que hay una cantidad determinada de minijuegos relacionados con ese tema.
- Estrellas: Un sistema de puntuación que refleja si un minijuego a sido completado.

1.3 Objetivo general

El objetivo general será la creación una plataforma para la gamificación de una asignatura basada en minijuegos.

Este objetivo general se concreta en los siguientes objetivos específicos:

- Desarrollar una aplicación web que sea compatible con los principales navegadores: *Google Chrome, Microsoft Edge y Mozilla Firefox*.
- Permitir agregar, eliminar, abrir y cerrar minijuegos.
- Permitir enlazar los minijuegos con la plataforma de forma automática.
- Fomentar la competitividad entre usuarios y grupos disponiendo de un *ranking* global.
- Permitir a los desarrolladores de los minijuegos guardar el estado del minijuego de cada usuario en cada partida.
- Permitir la investigación acerca del uso del juego. Para ello deberá guardar un registro con información acerca de la interacción del usuario. El objetivo es aprender, no enganchar y poder investigar acerca de cómo el uso de la plataforma y los minijuegos afecta a este aprendizaje es fundamental.

Capítulo 2

Conceptos previos

2.1 Unity

Unity es una herramienta de desarrollo de videojuegos creada por la empresa *Unity Technologies* y basada en el lenguaje de programación **C#**. **Unity** es lo que se conoce como motor de desarrollo o motor de juegos. El término de motor de videojuegos hace referencia a un software el cual tiene una serie de rutinas de programación que permiten el diseño, la creación y el funcionamiento de un entorno interactivo, por ejemplo un videojuego. También se ha utilizado para crear experiencias de realidad virtual interactivas e incluso miniseries producidas por *Disney* donde se ha utilizado el editor para procesar y pre-visualizar en tiempo real todos los capítulos de la miniserie.

Unity es una herramienta que no engloba únicamente motores para el renderizado de imágenes, de físicas de 2D/3D, de audio, de animaciones y otros motores, sino que engloba además herramientas de *networking* para multijugador, herramientas de navegación *NavMesh* para inteligencia artificial o soporte de realidad virtual.[5]

Por otra parte una de las características más importantes y más cómodas de **Unity** es que soporta la exportación a una cantidad enorme de plataformas, entre ellas web utilizando **WebGL** que es el que nos va a interesar.

Los elementos más usados en **Unity** son los escenas, objetos y componentes. Cada escena contiene múltiples objetos además de una cámara principal que se encarga de mostrar lo que ve el jugador. Cada objeto, en el juego es un *GameObject*, que puede contener desde personajes y objetos coleccionables hasta luces, cámaras y efectos especiales. A cada objeto se le puede añadir componentes como los *colliders* que determinan el ancho y el alto del “terreno” en otras palabras donde empieza el objeto, un ejemplo de *collider* sería el área que usarían un enemigos para detectar que estás cerca de el.

2.1.1 ¿Por que Unity?

Unity ofrece una gran serie de servicios que serán de gran ayudar en este proyecto ya que permite:

- Monitorizar el juego.
- Realizar analíticas para comprobar cómo se comportan los jugadores.
- Realizar construcciones remotas en la nube.
- Colaborar con otras personas en el mismo proyecto, ofreciendo un sistema similar a un sistema de control de versiones.
- Obtener la certificación **Unity** en los dos tipos de carreras, artística o técnica, de forma que podamos validar nuestro conocimiento de cara a un posible empleador.

Y ofrece otras muchas más herramientas, de colaboración, retención de jugadores, compras, multijugador, diagnósticos, etc.

También dispone de una excelente documentación, de las mejores documentaciones de software que existen. En el manual de **Unity** [6], se pueden consultar todos y cada uno de sus distintos apartados, desde cómo actualizar a una versión concreta, hasta guías de expertos para realizar tareas bastante avanzadas y que no todo el mundo necesita. En dicho manual también aparece el historial de las versiones anteriores de su documentación. Esto permite no estar limitados a la última versión del manual de la última versión de **Unity** que exista, sino que se puede consultar todo el histórico hacia atrás.

Otra de las grandes ventajas o virtudes de **Unity** es la inmensa comunidad de usuarios que tiene, no solamente dentro de los propios foros de **Unity** sino por todo Internet.

Es muy extraño que no surja una pregunta, una duda o algún problema en el desarrollo de nuestro juego con **Unity**, que no haya sido ya respondido o se esté discutiendo en algunos de los múltiples foros de **Unity**, dónde se explican las técnicas empleadas, los problemas que surgen y las soluciones que se encuentran para los mismos.

Finalmente destacar otra de las ventajas de **Unity**, y es que permite centrarse en lo es realmente importante para sus usuarios, la creación de videojuegos.

2.1.2 Ventajas

- Es fácil de usar.
- Permite el manejo de dos lenguajes para sus *scripts* (**C#** o **JavaScript**).
- Casi todo se puede referenciar con el manejo del ratón (arrastrar y soltar).
- Posee la tienda mas completa de complementos para los juegos, la *Asset Store*.
- Es multiplataforma.

2.1.3 Inconvenientes

- El espacio ocupado por los proyectos elevado.
- Requiere un alto rendimiento (se necesita una máquina preparada para la ejecución de emuladores con soltura).
- Puede dar problemas con el mantenimiento de versiones anteriores.
- La sencillez de su manejo permite aumentar la cantidad de juegos pero no la calidad de los mismos. Es el efecto denominado *Invasión de Juegos*

2.2 WebGL

WebGL (*Web Graphics Library*) es una especificación estándar que define una *API* implementada en **JavaScript** para la renderización de gráficos en 3D dentro de cualquier navegador web. **WebGL** está integrada completamente en todos los estándares web del navegador, permitiendo el procesamiento de imágenes y efectos como parte del lienzo o "*canvas*" de la página web. Los elementos de **WebGL** se pueden combinar con otros elementos **HTML** y estar compuestos con otras partes de la página o del fondo de la página. Los programas **WebGL** consisten en un código de control escrito en **JavaScript** que permite usar la implementación nativa de **OpenGL**, un lenguaje similar a **C** o **C++**, que se ejecuta en una unidad de procesamiento gráfico (*GPU*). **WebGL** está diseñado y gestionado por el consorcio de tecnología sin ánimo de lucro *Khronos Group* [7].

Unity WebGL es la opción de construcción de contenido **Unity** como programas **JavaScript** que utiliza tecnologías **HTML5** y el *API* de renderización **WebGL** para correr el contenido de **Unity** en el explorador web.

Capítulo 3

Análisis

En este capítulo, se pasa a describir de forma detallada la aplicación a desarrollar. Se definen los actores y roles, se especifican los requisitos y los casos de uso.

3.1 Descripción de los actores y roles de la aplicación

En este primer punto, se describen los roles que van a formar parte de la aplicación. En esta aplicación va a haber tres actores que dependiendo de su rol (y por lo tanto, de como inicien su sesión) podrán realizar una serie de funcionalidades u otras.

3.1.1 Usuario

Actor con rol “Usuario” lo representara cualquier persona registrada en el sistema. Podrá jugar a un minijuego, ver su puntuación, ver la puntuación de los diez mejores aunque el no este incluido y ver la puntuación de su grupo.

3.1.2 Desarrollador

Actor con rol “Desarrollador” lo representará cualquier persona designada como tal por el administrador. Podrá subir y administrar minijuegos hechos por esta misma persona. Además podrá hacer las mismas funcionalidades que el actor usuario a excepción de mostrar su puntuación ya que los desarrolladores no participan en la clasificación.

2.1.3 Administrador

Actor con rol “Administrador”, solo lo representan las personas designadas por otro administrador. Podrá crear y administrar mundos y minijuegos creados por otras personas así como añadir, modificar o eliminar usuarios del sistema. También podrá cambiar la configuración de los mundos y minijuegos, visible o no visible, abierto o cerrado y modificar la asignación de imágenes asociadas a cada minijuego. Además podrá hacer las mismas funcionalidades que el actor usuario a excepción de mostrar su puntuación ya que los administradores no participan en la clasificación.

3.2 Especificación de requisitos

3.2.1 Requisitos funcionales

En la tabla 1, se detallan los requisitos funcionales de la aplicación.

ID	Descripción
RF01	El sistema permitirá al usuario identificarse en la aplicación.
RF02	El sistema permitirá al usuario cerrar sesión.
RF03	El sistema permitirá al usuario inicial un minijuego.
RF04	El sistema permitirá al usuario consultar su puntuación personal.
RF05	El sistema permitirá al usuario consultar su puntuación de grupo.
RF06	El sistema permitirá al desarrollador consultar sus minijuegos agregados.
RF07	El sistema permitirá al desarrollador añadir un minijuego a la aplicación.
RF08	El sistema permitirá al desarrollador modificar sus minijuegos.
RF09	El sistema permitirá al desarrollador eliminar sus minijuegos.
RF10	El sistema permitirá al desarrollador probar sus minijuegos en la aplicación.
RF11	El sistema permitirá al administrador añadir mundos a la aplicación.
RF12	El sistema permitirá al administrador modificar mundos.
RF13	El sistema permitirá al administrador eliminar mundos.
RF14	El sistema permitirá al administrador consultar mundos.
RF15	El sistema permitirá al administrador consultar todos los minijuegos agregados.
RF16	El sistema permitirá al administrador añadir usuarios nuevos.
RF17	El sistema permitirá al administrador modificar usuarios.
RF18	El sistema permitirá al administrador eliminar usuarios.
RF19	El sistema permitirá al administrador añadir un lista de usuarios.
RF20	El sistema permitirá al administrador eliminar a todos los usuarios.
RF21	El sistema permitirá al administrador consultar todos los usuarios.

Tabla 1: Requisitos funcionales

3.2.2 Requisitos no funcionales

En la tabla 2, se muestra una serie de requisitos de no funcionales.

ID	Descripción	Importancia
RNF01	El sistema estará disponible a través de un navegador web.	Crítica
RNF02	El sistema deberá asegurar que todas las imágenes de los minijuegos se adapten a la aplicación	Deseable
RNF03	El sistema deberá garantizar que el 99% de los usuarios serán capaces de usar el 99% de las funcionalidades.	Deseable
RNF04	El sistema deberá estar disponible las 24 horas del día, 365 días al año.	Deseable
RNF05	El sistema deberá tener un tiempo de respuesta máximo de 5 segundos	Deseable
RNF06	El sistema deberá disponer de una interfaz que se adapte a dispositivos móviles, tablets y ordenadores	Deseable
RNF07	El sistema deberá utilizar el formato UTF-8	Crítica
RNF08	El sistema deberá ser capaz de funcionar en varios navegadores, como Chrome, Microsoft Edge y Firefox	Deseable

Tabla 2: Requisitos no funcionales

3.2.3 Requisitos de información

En la tabla 3, se muestra una serie de requisitos de información.

ID	Descripción
RI01	El sistema almacena información de los minijuegos, en particular: url de imagen, nombre y puntuación máxima, entre otros.
RI02	El sistema almacena información de los mundos, en particular: url del icono, url de imagen de fondo, nombre y posición, entre otros.
RI03	El sistema almacena información de los usuarios, en particular: nombre, contraseña, grupo y rol.
RI04	El sistema almacena información de los usuarios por minijuego, en particular: puntuación obtenida en el minijuego, tiempo, si se ha completado o no, fecha de inicio y fecha de fin del minijuego.
RI05	El sistema almacena información de la mejor partida jugada en cada nivel de cada usuario.
RI06	El sistema almacena información de todas las interacciones del usuario en la aplicación.

Tabla 3: Requisitos de información

3.2.4 Casos de uso

En esta sección, en la tabla 4, se detalla la lista de casos de uso identificados en el proceso de análisis, para la aplicación.

ID	Descripción	ID	Descripción
CU01	Identificarse	CU02	Cerrar sesión
CU03	Mostrar puntuaciones	CU04	Mostrar puntuaciones de grupo
CU05	Mostrar mundos	CU06	Mostrar minijuegos
CU07	Jugar minijuego	CU08	Añadir minijuego
CU09	Añadir proyecto al minijuego	CU10	Revisar minijuegos
CU11	Modificar minijuego	CU12	Modificar proyecto del minijuego
CU13	Eliminar minijuego	CU14	Revisar mundos y minijuegos
CU15	Hacer visible minijuego	CU16	Ocultar minijuego
CU17	Abrir minijuego	CU18	Cerrar minijuego
CU19	Cambiar posición minijuego	CU20	Añadir mundo
CU21	Modificar mundo	CU22	Eliminar mundo
CU23	Hacer visible mundo	CU24	Ocultar mundo
CU25	Abrir mundo	CU26	Cerrar mundo
CU27	Cambiar posición mundo	CU28	Añadir usuario
CU29	Añadir lista de usuarios	CU30	Revisar usuarios
CU31	Modificar usuario	CU32	Eliminar usuario
CU33	Eliminar todos los usuarios		

Tabla 4: Casos de uso

3.3 Casos de uso

3.3.1 Diagrama de casos de uso

En la figura 1 se presenta el diagrama de casos de uso desarrollado para la aplicación.

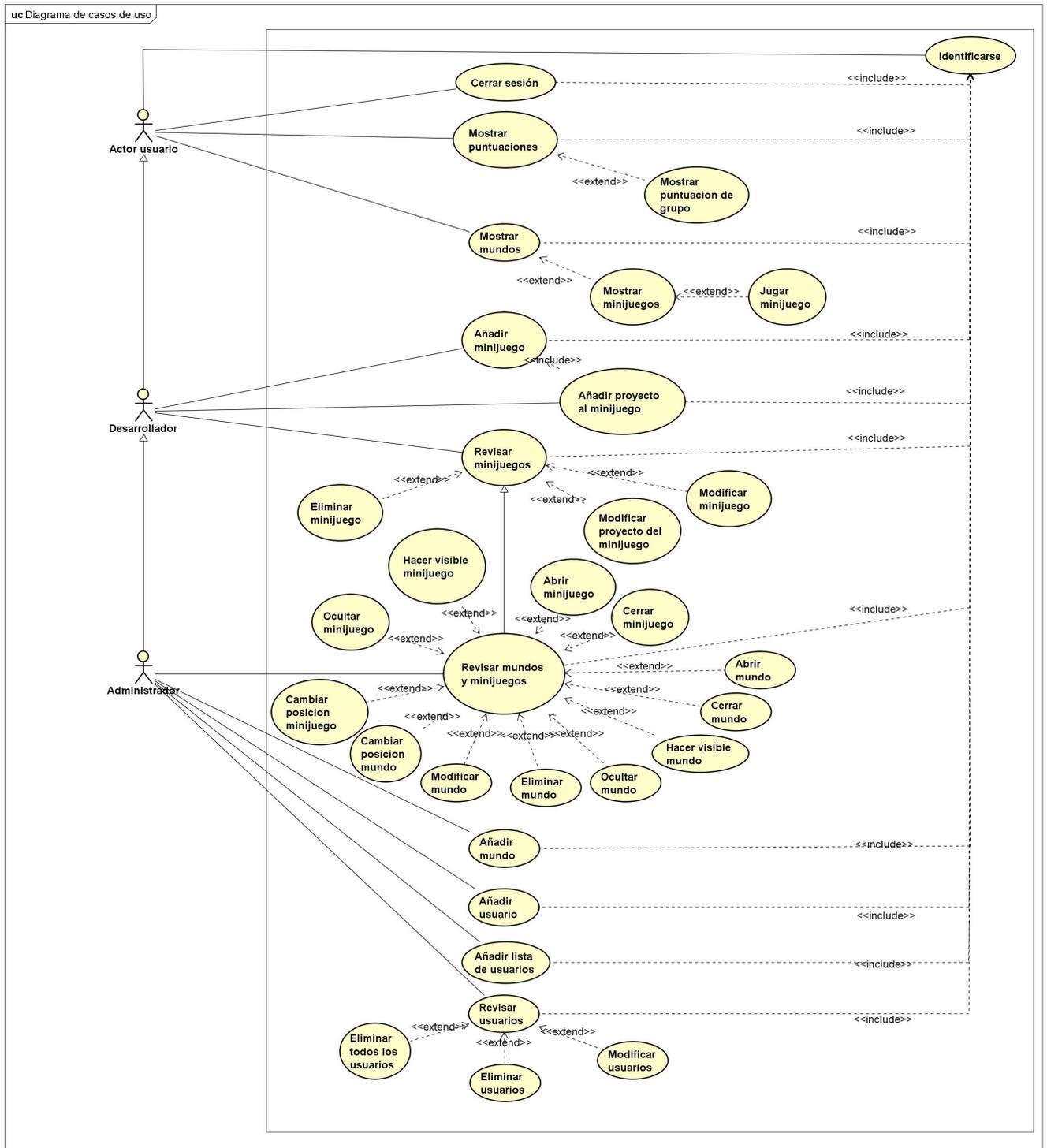


Figura 1: Diagrama de casos de uso

3.3.2 Especificación de los casos de uso

En las tablas (Tabla 5 - Tabla 37), se encuentran las especificaciones textuales de todos los casos de uso identificados.

CU-01	Identificarse	
Versión	1.0	
Dependencias		
Actores	Usuario, Desarrollador o Administrador	
Precondición	El usuario está registrado en el sistema.	
Descripción	El usuario inicia sesión en el sistema a través de sus credenciales	
Secuencia normal	Paso	Acción
	1.	El sistema solicita el nombre de usuario y la contraseña.
	2.	El usuario introduce el nombre de usuario y la contraseña.
	3.	El sistema comprueba que los datos son correctos.
	4.	El sistema muestra la página principal.
	5.	Fin del caso de uso.
Postcondición	Se ha iniciado sesión en el sistema	
Excepciones	Paso	Acción
	3a.	El sistema comprueba que los datos no son correctos, el caso de uso retorna al paso 1.
Comentarios		

Tabla 5: Descripción del caso de uso: Identificarse

CU-02	Cerrar sesión	
Versión	1.0	
Dependencias		
Actores	Usuario, Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse".	
Descripción	El usuario cierra sesión de la aplicación.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Salir".
	2.	El sistema solicita confirmación.
	3.	El usuario confirma.
	4.	El sistema cierra la sesión del usuario.
	5.	Fin del caso de uso.
Postcondición	Se ha cerrado la sesión del usuario en el sistema.	
Excepciones	Paso	Acción
	3a.	El usuario cancela y el caso de uso queda sin efecto.
Comentarios		

Tabla 6: Descripción del caso de uso: Cerrar sesión

CU-03	Mostrar puntuaciones	
Versión	1.0	
Dependencias		
Actores	Usuario, Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse".	
Descripción	El usuario consulta la tabla de puntuaciones.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Ranking".
	2.	El sistema muestra la lista de puntuaciones ordenada.
	3.	Fin del caso de uso.
Postcondición	Se muestra la lista de puntuaciones ordenada.	
Excepciones	Paso	Acción
Comentarios		

Tabla 7: Descripción del caso de uso: Mostrar puntuaciones

CU-04	Mostrar puntuación de grupo	
Versión	1.0	
Dependencias		
Actores	Usuario, Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Mostrar puntuaciones".	
Descripción	El usuario consulta la tabla de puntuaciones de grupos.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona el filtro "Grupo"
	2	El sistema muestre la lista de puntuaciones de grupos ordenada
	3.	Fin del caso de uso.
Postcondición	Se muestra la lista de puntuaciones de grupos ordenada.	
Excepciones	Paso	Acción
Comentarios		

Tabla 8: Descripción del caso de uso: Mostrar puntuaciones de grupos

CU-05	Mostrar mundos	
Versión	1.0	
Dependencias		
Actores	Usuario, Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse".	
Descripción	El usuario consulta los mundos disponibles.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Start".
	2.	El sistema muestra los mundos ordenados por posición.
	3.	Fin del caso de uso.
Postcondición	Se muestra los mundos ordenados por posición.	
Excepciones	Paso	Acción
	2a.	No hay mundos en el sistema por lo que no muestra ninguno y el caso de uso continua.
Comentarios		

Tabla 9: Descripción del caso de uso: Mostrar mundos

CU-06	Mostrar minijuegos	
Versión	1.0	
Dependencias		
Actores	Usuario, Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Mostrar mundos". En el sistema haya al menos un mundo.	
Descripción	El usuario consulta los minijuegos de un mundo seleccionado.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona un mundo.
	2.	El sistema muestra los minijuegos del mundo seleccionado.
	3.	Fin del caso de uso.
Postcondición	Se muestran los minijuegos de un mundo seleccionado.	
Excepciones	Paso	Acción
	2a.	El mundo seleccionado no tiene minijuegos por lo que no muestra ninguno y el caso de uso continua.
Comentarios		

Tabla 10: Descripción del caso de uso: Mostrar minijuegos

CU-07	Jugar minijuego	
Versión	1.0	
Dependencias		
Actores	Usuario, Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Mostrar minijuegos". En el sistema haya al menos un minijuego en el mundo seleccionado.	
Descripción	El usuario juega a un minijuego.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona un minijuego
	2.	El sistema comprueba si ha jugado previamente y carga sus anteriores resultados.
	3.	El sistema crea una nueva partida con una fecha de inicio.
	4.	El sistema ejecuta el minijuego correspondiente.
	5.	El usuario termina el minijuego.
	6.	El sistema comprueba los resultados, si son mejores los actualiza.
	7.	El sistema actualiza los datos de la nueva partida, establece una fecha final, puntuación, tiempo y completado o no.
	8.	El sistema muestra los minijuegos del mundo seleccionado.
9.	Fin del caso de uso.	
Postcondición	El usuario juega a un minijuego.	
Excepciones	Paso	Acción
	2a.	El sistema no encuentra datos anteriores de este usuario, el sistema continúa.
	5a.	El usuario no termina el minijuego, el sistema no establece una fecha final y el caso de uso termina.
Comentarios	No es lo mismo los datos de la nueva partida que los datos de guardado. Los datos de nueva partida se crean siempre que se juega a un minijuego y pueden repetirse. Los datos de guardado son para que los creadores de minijuegos puedan guardar un estado del minijuego para ese usuario, lo que comúnmente se conoce como una partida guardada.	

Tabla 11: Descripción del caso de uso: Jugar a un minijuego

CU-08	Añadir minijuego	
Versión	1.0	
Dependencias		
Actores	Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como desarrollador o administrador. Hay al menos un mundo en el sistema.	
Descripción	El usuario añade un minijuego al sistema.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Panel de desarrollador". ("Panel de administrador" si es el caso de un usuario administrador).
	2.	El sistema muestra un panel de opciones: "Mundos y minijuegos" ("Revisar minijuegos" si el usuario es desarrollador), "Revisar usuarios", "Añadir mundo", "Añadir minijuego" y "Añadir usuario".
	3.	El usuario selecciona la opción "Añadir minijuego".
	4.	El sistema muestra el formularios de añadir minijuego.
	5.	El usuario ingresa los datos de mundo, nombre, imagen y puntuación máxima en el formulario añadir minijuego.
	6.	El sistema comprueba que los datos se han introducido correctamente
	7.	El sistema añade el minijuego.
	8.	El sistema informa al usuario y muestra el código del minijuego creado.
	9.	Fin del caso de uso.
Postcondición	El usuario añade un minijuego al sistema.	
Excepciones	Paso	Acción
	6a.	El sistema comprueba que los datos no son correctos y el caso de uso vuelve al paso 4.
Comentarios		

Tabla 12: Descripción del caso de uso: Añadir minijuego

CU-09	Añadir proyecto al minijuego	
Versión	1.0	
Dependencias		
Actores	Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como desarrollador o administrador. Se ha realizado el caso de uso: "Añadir minijuego" y en el sistema ya ha proporcionado el código del minijuego.	
Descripción	El usuario añade un proyecto a un minijuego.	
Secuencia normal	Paso	Acción
	1.	El usuario añade el proyecto con nombre del código facilitado.
	2.	El sistema comprueba que el proyecto añadido es el correcto.
	3.	El sistema añade el proyecto al minijuego.
	4.	El sistema informa al usuario que el proyecto ha sido añadido correctamente.
	5.	Fin del caso de uso.
Postcondición	El minijuego ya está disponible para jugarse.	
Excepciones	Paso	Acción
	2a.	El sistema detecta que el proyecto no es correcto y el caso de uso vuelve al paso 1.
Comentarios	Este caso de uso puede hacerse en cualquier momento: este caso de uso no tiene por qué ejecutarse inmediatamente después que el caso de uso añadir minijuego. Pero si necesitas haber ejecutado el caso de uso añadir minijuego primero.	

Tabla 13: Descripción del caso de uso: Añadir proyecto al minijuego

CU-10	Revisar minijuegos	
Versión	1.0	
Dependencias		
Actores	Desarrollador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como desarrollador.	
Descripción	El usuario consulta la lista de minijuegos creados por el usuario	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Panel de desarrollador".
	2.	El sistema muestra un panel de opciones: "Revisar minijuegos", "Revisar usuarios", "Añadir mundo", "Añadir minijuego" y "Añadir usuario".
	3.	El usuario selecciona "Revisar minijuegos"
	4.	El sistema muestra la lista de los minijuegos que pertenecen a este usuario y a que mundos pertenecen.
	5.	Fin del caso de uso.
Postcondición	El usuario consulta los minijuegos agregados por el usuario.	
Excepciones	Paso	Acción
Comentarios		

Tabla 14: Descripción del caso de uso: Revisar minijuegos

CU-11	Modificar minijuego	
Versión	1.0	
Dependencias		
Actores	Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: “Revisar minijuegos” o “Revisar mundos y minijuegos”.	
Descripción	El usuario modifica un minijuego.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción “Modificar” del minijuego.
	2.	El sistema carga los datos del minijuego y muestra los formularios de añadir minijuego y añadir proyecto.
	4.	El usuario modifica los datos de mundo, nombre, imagen y puntuación máxima en el formulario.
	5.	El sistema comprueba que los datos se han introducido correctamente
	6.	El sistema actualiza el minijuego.
	7.	El sistema informa al usuario de que el minijuego ha sido modificado correctamente.
	8.	Fin del caso de uso.
Postcondición	El usuario modifica un minijuego.	
Excepciones	Paso	Acción
	5a.	El sistema comprueba que los datos no son correctos y el caso de uso vuelve al paso 4.
Comentarios		

Tabla 15: Descripción del caso de uso: Modificar minijuego

CU-12	Modificar proyecto del minijuego	
Versión	1.0	
Dependencias		
Actores	Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: “Revisar minijuegos” o “Revisar mundos y minijuegos”.	
Descripción	El usuario modifica un minijuego.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción “Modificar” del minijuego.
	2.	El sistema carga los datos del minijuego y muestra los formularios de añadir minijuego y añadir proyecto.
	3.	El usuario modifica el proyecto con nombre del código facilitado.
	4.	El sistema comprueba que el proyecto añadido es el correcto.
	5.	El sistema actualiza el proyecto del minijuego.
	6.	El sistema informa al usuario que el proyecto ha sido añadido correctamente.
	7.	Fin del caso de uso.
Postcondición	El usuario modifica el proyecto del minijuego.	
Excepciones	Paso	Acción
	4a.	El sistema detecta que el proyecto no es correcto y el caso de uso vuelve al paso 3.
Comentarios	El código facilitado es siempre será el mismo para ese minijuego.	

Tabla 16: Descripción del caso de uso: Modificar proyecto del minijuego

CU-13	Eliminar minijuego	
Versión	1.0	
Dependencias		
Actores	Desarrollador o Administrador	
Precondición	Se ha realizado el caso de uso: “Revisar minijuegos” o “Revisar mundos y minijuegos”.	
Descripción	El usuario elimina un minijuego.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción “Eliminar” del minijuego.
	2.	El sistema pide confirmación.
	3.	El usuario acepta.
	4.	El sistema comprueba que no haya puntuaciones de los jugadores.
	5.	El sistema elimina el minijuego y todo lo relacionado con el.
	6.	El sistema informa al usuario que el minijuego ha sido eliminado correctamente.
7.	Fin del caso de uso.	
Postcondición	El usuario elimina el minijuego.	
Excepciones	Paso	Acción
Comentarios		

Tabla 17: Descripción del caso de uso: Eliminar minijuego

CU-14	Revisar mundos y minijuegos	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como administrador.	
Descripción	El usuario consulta la lista de todos minijuegos creados	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Panel de administrador".
	2.	El sistema muestra un panel de opciones: "Mundos y minijuegos", "Revisar usuarios", "Añadir mundo", "Añadir minijuego" y "Añadir usuario".
	3.	El usuario selecciona la opción "Mundos y minijuegos"
	3.	El sistema muestra la lista de los minijuegos que pertenecen a este todos los usuario y a que mundos pertenecen.
	3.	Fin del caso de uso.
Postcondición	El usuario consulta todos los minijuegos agregados.	
Excepciones	Paso	Acción
Comentarios		

Tabla 18: Descripción del caso de uso: Revisar mundos y minijuegos

CU-15	Hacer visible minijuego	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El usuario hará visible un minijuego para el resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Hacer visible" del minijuego.
	2.	El sistema hace visible el minijuego.
	3.	Fin del caso de uso.
Postcondición	El usuario ya puede ver el minijuego en el mundo que le corresponde.	
Excepciones	Paso	Acción
Comentarios		

Tabla 19: Descripción del caso de uso: Hacer visible minijuego

CU-16	Ocultar minijuego	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El usuario ocultara un minijuego del resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Ocultar" del minijuego.
	2.	El sistema oculta el minijuego.
	3.	Fin del caso de uso.
Postcondición	El usuario ya no puede ver el minijuego en el mundo que le corresponde.	
Excepciones	Paso	Acción
Comentarios	Solo los administradores podrán verlo.	

Tabla 20: Descripción del caso de uso: Ocultar minijuego

CU-17	Abrir minijuego	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El sistema permitirá el acceso a un minijuego al resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Abrir" del minijuego.
	2.	El sistema permite el acceso al minijuego.
	3.	Fin del caso de uso.
Postcondición	El usuario ya puede acceder al minijuego en el mundo que le corresponde.	
Excepciones	Paso	Acción
Comentarios		

Tabla 21: Descripción del caso de uso: Abrir minijuego

CU-18	Cerrar minijuego	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El sistema no permitirá el acceso a un minijuego al resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Cerrar" del minijuego.
	2.	El sistema no permite el acceso al minijuego.
	3.	Fin del caso de uso.
Postcondición	El usuario ya no puede acceder al minijuego en el mundo que le corresponde.	
Excepciones	Paso	Acción
Comentarios	Solo los administradores podrán acceder.	

Tabla 22: Descripción del caso de uso: Cerrar minijuego

CU-19	Cambiar posición minijuego	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El usuario cambiará la posición del minijuego	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Subir posición" del minijuego.
	2.	El sistema intercambia la posición con el siguiente minijuego en la lista.
	3.	Fin del caso de uso.
Secuencia alternativa	Paso	Acción
	1.	El usuario selecciona la opción "Bajar posición" del minijuego .
	2.	El sistema intercambia la posición con el anterior minijuego en la lista.
	3.	Fin del caso de uso.
Postcondición	El minijuego ha cambiado su posición	
Excepciones	Paso	Acción
Comentarios	No es posible subir mas arriba si ya estás en la primera posición y no es posible bajar más abajo si ya estás en la última posición	

Tabla 23: Descripción del caso de uso: Cambiar posición minijuego

CU-20	Añadir mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como administrador.	
Descripción	El usuario añade un mundo al sistema.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Panel de administrador".
	2.	El sistema muestra un panel de opciones: "Mundos y minijuegos", "Revisar usuarios", "Añadir mundo", "Añadir minijuego" y "Añadir usuario".
	3.	El usuario selecciona la opción "Añadir mundo".
	4.	El sistema muestra el formulario de añadir mundo.
	5.	El usuario ingresa los datos de nombre, imagen y imagen de fondo.
	6.	El sistema comprueba que los datos se han introducido correctamente.
	7.	El sistema añade el mundo.
	8.	El sistema informa al usuario de que el mundo se ha añadido correctamente.
9.	Fin del caso de uso.	
Postcondición	El usuario añade un mundo al sistema.	
Excepciones	Paso	Acción
	6a.	El sistema comprueba que los datos no son correctos y el caso de uso vuelve al paso 4.
Comentarios		

Tabla 24: Descripción del caso de uso: Añadir mundo

CU-21	Modificar mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos"	
Descripción	El usuario modifica un mundo.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Modificar" del mundo.
	2.	El sistema carga los datos del mundo y muestra el formulario.
	4.	El usuario modifica los datos de nombre, imagen e imagen fondo.
	5.	El sistema comprueba que los datos se han introducido correctamente
	6.	El sistema actualiza el mundo.
	7.	El sistema informa al usuario de que el mundo se ha modificado correctamente.
	8.	Fin del caso de uso.
Postcondición	El minijuego ha sido modificado.	
Excepciones	Paso	Acción
	5a.	El sistema comprueba que los datos no son correctos y el caso de uso vuelve al paso 4.
Comentarios		

Tabla 25: Descripción del caso de uso: Modificar mundo

CU-22	Eliminar mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos"	
Descripción	El usuario elimina un mundo.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Eliminar" del mundo.
	2.	El sistema pide confirmación.
	3.	El usuario acepta.
	4.	El sistema comprueba que no haya minijuegos asociados a ese mundo.
	5.	El sistema elimina el mundo y todo lo relacionado con el.
	6.	El sistema informa al usuario que el mundo ha sido eliminado correctamente.
	7.	Fin del caso de uso.
Postcondición	El mundo ha sido eliminado.	
Excepciones	Paso	Acción
Comentarios		

Tabla 26: Descripción del caso de uso: Eliminar mundo

CU-23	Hacer visible mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El usuario hará visible un mundo para el resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Hacer visible" del mundo..
	2.	El sistema hace visible el mundo..
	3.	Fin del caso de uso.
Postcondición	El usuario ya puede ver el mundo.	
Excepciones	Paso	Acción
Comentarios		

Tabla 27: Descripción del caso de uso: Hacer visible mundo

CU-24	Ocultar mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El usuario ocultara un mundo del resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Ocultar" del mundo.
	2.	El sistema oculta el mundo.
	3.	Fin del caso de uso.
Postcondición	El usuario ya no puede ver el mundo.	
Excepciones	Paso	Acción
Comentarios	Solo los administradores podrán verlo.	

Tabla 28: Descripción del caso de uso: Ocultar mundo

CU-25	Abrir mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El sistema permitirá el acceso a un mundo al resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Abrir" del mundo.
	2.	El sistema permite el acceso al mundo.
	3.	Fin del caso de uso.
Postcondición	El usuario ya puede acceder al mundo.	
Excepciones	Paso	Acción
Comentarios		

Tabla 29: Descripción del caso de uso: Abrir mundo.

CU-26	Cerrar mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El sistema no permitirá el acceso a un mundo al resto de usuarios.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Cerrar" del mundo.
	2.	El sistema no permite el acceso al mundo.
	3.	Fin del caso de uso.
Postcondición	El usuario ya no puede acceder al mundo.	
Excepciones	Paso	Acción
Comentarios	Solo los administradores podrán acceder.	

Tabla 30: Descripción del caso de uso: Cerrar mundo

CU-27	Cambiar posición mundo	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar mundos y minijuegos".	
Descripción	El usuario cambiará la posición del mundo	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Subir posición" del mundo.
	2.	El sistema intercambia la posición con el siguiente mundo en la lista.
	3.	Fin del caso de uso.
Secuencia alternativa	Paso	Acción
	1.	El usuario selecciona la opción "Bajar posición" del mundo.
	2.	El sistema intercambia la posición con el anterior mundo en la lista.
	3.	Fin del caso de uso.
Postcondición	El mundo ha cambiado su posición	
Excepciones	Paso	Acción
Comentarios	No es posible subir mas arriba si ya estás en la primera posición y no es posible bajar más abajo si ya estás en la última posición	

Tabla 31: Descripción del caso de uso: Cambiar posición mundo

CU-28	Añadir usuario	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como administrador.	
Descripción	El usuario añade un usuario al sistema.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Panel de administrador".
	2.	El sistema muestra un panel de opciones: "Mundos y minijuegos", "Revisar usuarios", "Añadir mundo", "Añadir minijuego" y "Añadir usuario".
	3.	El usuario selecciona la opción "Añadir usuario".
	4.	El sistema muestra los formularios de añadir usuario y añadir lista de usuarios.
	5.	El usuario ingresa los datos de nombre, contraseña, rol y grupo en el formulario de añadir usuario.
	6.	El sistema comprueba que los datos se han introducido correctamente.
	7.	El sistema comprueba si existe el usuario.
	8.	El sistema añade el usuario al sistema.
	9.	El sistema informa al usuario de que el usuario nuevo ha sido añadido correctamente.
10.	Fin del caso de uso.	
Postcondición	El usuario nuevo ha sido añadido al sistema	
Excepciones	Paso	Acción
	6a.	El sistema comprueba que los datos no son correctos y el caso de uso vuelve al paso 4.
7a.	El sistema encuentra un usuario existente, el sistema informa al usuario y el caso de uso vuelve al paso 5.	
Comentarios		

Tabla 32: Descripción del caso de uso: Añadir usuario

CU-29	Añadir lista usuarios	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como administrador.	
Descripción	El usuario añade una lista de usuarios al sistema.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Panel de administrador".
	2.	El sistema muestra un panel de opciones: "Mundos y minijuegos", "Revisar usuarios", "Añadir mundo", "Añadir minijuego" y "Añadir usuario".
	3.	El usuario selecciona la opción "Añadir usuario".
	4.	El sistema muestra los formularios de añadir usuario y añadir lista de usuarios.
	5.	El usuario ingresa un fichero con la lista de los usuarios.
	6.	El sistema comprueba que los datos se han introducido correctamente.
	7.	El sistema comprueba si existen los usuarios.
	8.	El sistema añade a los usuario al sistema.
	9.	El sistema informa al usuario de que la lista de usuarios ha sido añadida correctamente.
10.	Fin del caso de uso.	
Postcondición	La lista de usuarios nuevos han sido añadidos al sistema	
Excepciones	Paso	Acción
	6a.	El sistema comprueba que el fichero no es correcto y el caso de uso vuelve al paso 4.
7a.	El sistema encuentra un usuario existente, modifica los usuarios y el caso de uso continua.	
Comentarios	El caso de uso "Añadir lista usuarios" está pensado para introducir grandes cantidades de usuarios o resetearlos.	

Tabla 33: Descripción del caso de uso: Añadir lista usuarios

CU-30	Revisar usuarios	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Identificarse" y el usuario está identificado en el sistema como administrador.	
Descripción	El usuario consulta la lista de todos usuarios creados.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Panel de administrador".
	2.	El sistema muestra un panel de opciones: "Mundos y minijuegos", "Revisar usuarios", "Añadir mundo", "Añadir minijuego" y "Añadir usuario".
	3.	El usuario selecciona la opción "Revisar usuarios".
	4.	El sistema muestra la lista de los usuarios que hay actualmente en el sistema, ordenados por rol y nombre.
	5.	Fin del caso de uso.
Postcondición	El usuario consulta la lista de todos usuarios creados.	
Excepciones	Paso	Acción
Comentarios		

Tabla 34: Descripción del caso de uso: Revisar usuarios

CU-31	Modificar usuario	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar usuarios"	
Descripción	El usuario modifica otro usuario.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Modificar" del usuario.
	2.	El sistema carga los datos del usuario y muestra el formulario.
	3.	El usuario modifica los datos de nombre, contraseña, rol y grupo.
	4.	El sistema comprueba que los datos se han introducido correctamente
	5.	El sistema comprueba si existe el usuario
	6.	El sistema actualiza el usuario.
	7.	El sistema informa al usuario de que el usuario se ha modificado correctamente.
8.	Fin del caso de uso.	
Postcondición	El usuario ha sido modificado.	
Excepciones	Paso	Acción
	4a.	El sistema comprueba que los datos no son correctos y el caso de uso vuelve al paso 3.
	5a.	El sistema no encuentra el usuario que se quiere modificar, el sistema informa al usuario el caso de uso vuelve al paso 3.
Comentarios		

Tabla 35: Descripción del caso de uso: Modificar usuario

CU-32	Eliminar usuario	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar usuarios"	
Descripción	El usuario elimina un mundo.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Eliminar" del usuario.
	2.	El sistema pide confirmación.
	3.	El usuario acepta.
	4.	El sistema elimina el usuario y todo lo relacionado con el.
	5.	El sistema informa al usuario que el usuario ha sido eliminado correctamente.
	6.	Fin del caso de uso.
Postcondición	El mundo ha sido eliminado.	
Excepciones	Paso	Acción
	3a.	El usuario rechaza la confirmación el caso de uso queda sin efecto.
Comentarios		

Tabla 36: Descripción del caso de uso: Eliminar usuario

CU-33	Eliminar todos los usuarios	
Versión	1.0	
Dependencias		
Actores	Administrador	
Precondición	Se ha realizado el caso de uso: "Revisar usuarios"	
Descripción	El usuario elimina un mundo.	
Secuencia normal	Paso	Acción
	1.	El usuario selecciona la opción "Eliminar todos los usuarios"
	2.	El sistema pide confirmación.
	3.	El usuario acepta.
	4.	El sistema pide una segunda confirmación
	5.	El usuario acepta.
	6.	El sistema elimina todos los usuarios con rol usuario y todo lo que tenga que ver con ellos.
	7.	El sistema informa al usuario que los usuarios han sido eliminados correctamente.
8.	Fin del caso de uso.	
Postcondición	El mundo ha sido eliminado.	
Excepciones	Paso	Acción
	3a.	El usuario rechaza la confirmación el caso de uso queda sin efecto.
	5a.	El usuario rechaza la confirmación el caso de uso queda sin efecto.
Comentarios		

Tabla 37: Descripción del caso de uso: Eliminar todos los usuarios

3.4 Modelo de dominio

En la figura 2, se muestra el modelo de dominio. En este, se encuentran las clases identificadas a nivel de análisis y las relaciones entre ellas.

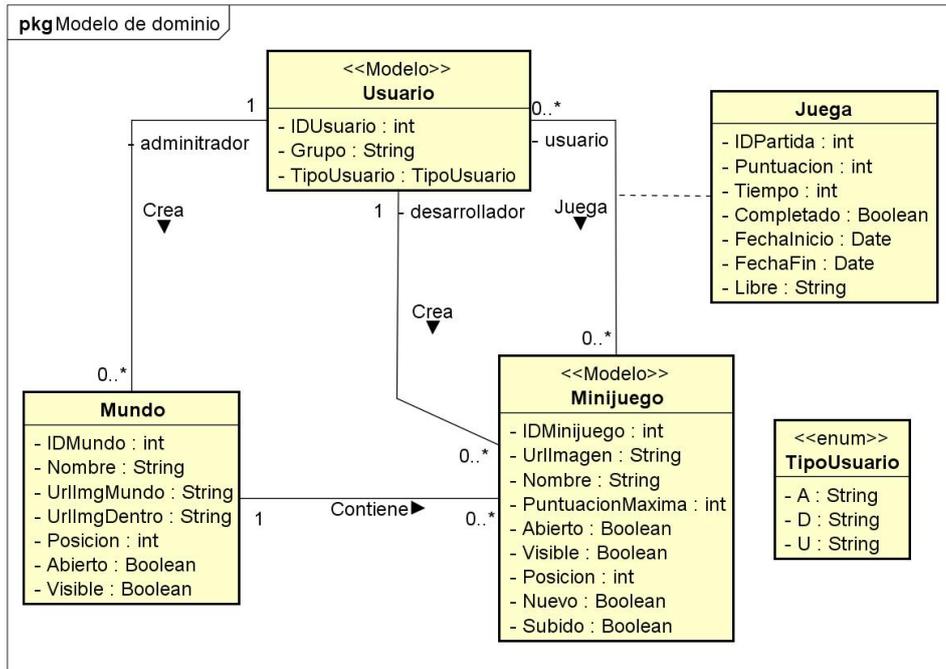


Figura 2: Modelo de dominio

Capítulo 4

Plan de proyecto

4.1 Plan de proceso

Para el desarrollo de la aplicación se ha decidido utilizar un desarrollo en cascada. En la figura 3 se puede observar el proceso:



Figura 3: Desarrollo en cascada

El desarrollo en cascada es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como punto de partida para la siguiente.

4.1.1 Funcionamiento

En la práctica, se aplican diversas versiones del modelo. Los más habituales son los modelos que dividen los procesos de desarrollo en cinco fases.

1. Análisis: Planificación, análisis y especificación de los requisitos.
2. Diseño: Diseño y especificación del sistema.
3. Implementación: Programación y pruebas unitarias.
4. Verificación: Integración de sistemas, pruebas de sistema y de integración.
5. Mantenimiento: Entrega, mantenimiento y mejora.

Estas fases se pueden agrupar en 3 bloques tal como se muestra en la figura 4:

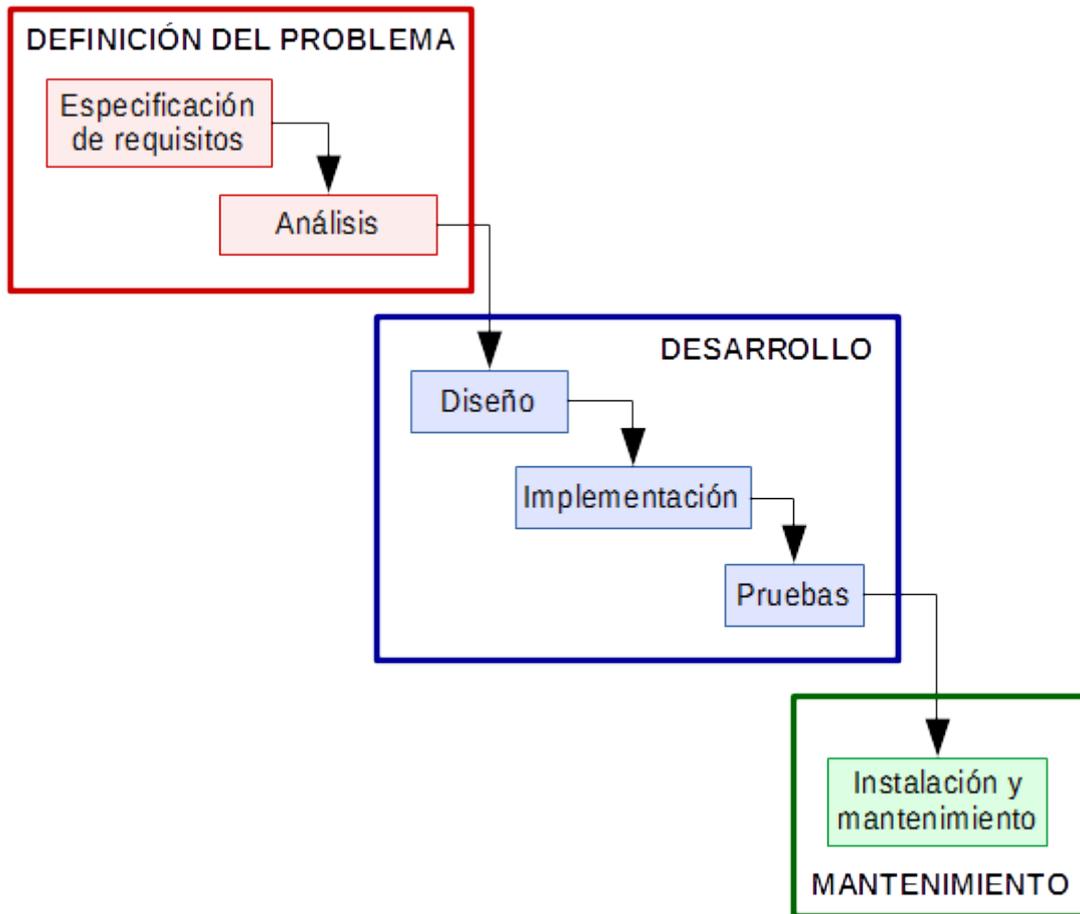


Figura 4: Desarrollo en cascada agrupado

En las ampliaciones de la metodología en cascada se añaden funciones iterativas al modelo básico como, por ejemplo, los saltos hacia atrás, que permiten comparar los resultados de cada una de las fases con las hipótesis obtenidas en la fase anterior, de modo que se puedan verificar.

En este modelo, las diferentes fases de un proceso de desarrollo se suceden una detrás de otra como en una cascada. Cada una de las fases concluye con un resultado provisional (hito) como, por ejemplo, un catálogo de requisitos en forma de pliego de condiciones, la especificación de una arquitectura de software o una aplicación a nivel alfa o beta.

4.1.2 Ventajas

- Se obtiene una estructura sencilla gracias a unas fases de proyecto claramente diferenciadas.
- Se genera una buena documentación del proceso de desarrollo a través de unos hitos bien definidos.
- Los costes y la carga de trabajo se pueden estimar al comenzar el proyecto.
- Aquellos proyectos que se estructuran en base al modelo en cascada se pueden representar cronológicamente de forma sencilla.

4.1.3 Inconvenientes

- Por norma general, los proyectos más complejos o de varios niveles no permiten su división en fases de proyecto claramente diferenciadas.
- Permite poco margen para realizar ajustes a lo largo del proyecto debido a un cambio en las exigencias.
- El usuario final no se integra en el proceso de producción hasta que no termina la programación.
- En ocasiones, los fallos solo se detectan una vez finalizado el proceso de desarrollo [8].

4.2 Plan de gestión de riesgos

La gestión de riesgos es el proceso de valorar y controlar los riesgos que afectan a un producto, proceso o proyecto de software. Se trata de identificar problemas que pueden ocurrir durante el desarrollo del proyecto para tomar medidas en caso de producirse.

Para poder realizar un buen análisis de riesgos y evitar posibles inconvenientes en el desarrollo del proyecto, se debe realizar una cuantificación de los riesgos. Para ello, en el proceso de análisis de los riesgos se debe obtener el grado de exposición al riesgo. Esto se puede obtener dependiendo del nivel de impacto (Alto, Medio o Bajo) y de la probabilidad de suceso (Alto, medio o bajo).

4.2.1 Análisis de riesgos

En esta sección se encuentra el análisis de los riesgos identificados para el proyecto a desarrollar.

Identificador	R01 - Disponibilidad del desarrollador
Descripción	No se puede cumplir con el trabajo establecido debido a la indisponibilidad por trabajo, ya que el desarrollador se encuentra empleado por otra empresa.
Impacto	Alto
Probabilidad	Alto
Plan de mitigación	Dedicar los días no laborales sin excepción al desarrollo del proyecto, es decir, cumplir con la calendarización.
Plan de contingencia	Priorizar las tareas.

Tabla 38: Disponibilidad del desarrollador

Identificador	R02 - Enfermedad
Descripción	No se puede cumplir con el trabajo establecido debido a enfermedad o motivos personales del desarrollador.
Impacto	Alto
Probabilidad	Medio
Plan de mitigación	-
Plan de contingencia	-

Tabla 39: Enfermedad

Identificador	R03 – Perdida de datos o/y documentación
Descripción	Se produce la pérdida de algún documento o de datos ya desarrollados, es decir, pérdida de tiempo empleado en el desarrollo del proyecto
Impacto	Alto
Probabilidad	Bajo
Plan de mitigación	Realizar copias de seguridad en la nube, y en local cada vez que se realice algún documento o se haya avanzado consecuentemente en el trabajo.
Plan de contingencia	Evaluar el impacto y volver a planificar en consecuencia.

Tabla 40: Perdida de datos o/y documentación

Identificador	R04 – Fallo en la maquina
Descripción	Avería del entorno de trabajo producida por causas naturales .
Impacto	Medio
Probabilidad	Bajo
Plan de mitigación	Utilizar el ordenador de manera adecuada y cuidadosa, controlar posibles anomalías detectadas.
Plan de contingencia	Arreglar u obtener otra máquina, y montar el entorno de nuevo.

Tabla 41: Fallo en la maquina

Identificador	R05 – Diseño mal elaborado
Descripción	El diseño no se ajusta a lo que realmente se requiere
Impacto	Alto
Probabilidad	Medio
Plan de mitigación	Ante cualquier duda comunicárselo al cliente y realizarlo con respecto al documento de análisis previamente elaborado.
Plan de contingencia	Replanificar las tareas y corregir el error.

Tabla 42: Diseño mal elaborado

Identificador	R06 – Conocimiento insuficiente sobre las tecnologías
Descripción	El tiempo que se ha establecido para aprender las tecnologías y herramientas a usar, no es suficiente para desarrollar todo lo que se requiere en la aplicación.
Impacto	Alto
Probabilidad	Medio
Plan de mitigación	Realizar una labor de investigación para ver que nivel de conocimiento se va a necesitar de cada tecnología, y establecer el tiempo de aprendizaje en base a las partes que se van a usar, para ser más precisos en tiempo.
Plan de contingencia	Replanificar las tareas y priorizarlas, dedicando más tiempo a la búsqueda de ejemplos y usos de las tecnologías desconocidas, para poder avanzar

Tabla 43: Conocimiento insuficiente sobre las tecnologías

Identificador	R07 – Cambios en los requisitos
Descripción	Durante el desarrollo del proyecto se han encontrado nuevos requisitos que no se tuvieron en cuenta en un principio.
Impacto	Alto
Probabilidad	Bajo
Plan de mitigación	Centrarse en el análisis de requisitos y comentar las dudas en las primeras fases del proyecto. Tener claros los requisitos al inicio al ser un proyecto libre
Plan de contingencia	Evaluar el impacto, replanificar las tareas, priorizarlas y realizar las acciones necesarias para resolverlo.

Tabla 44: Cambios en los requisitos

Identificador	R08 – Fallo en la conexión a Internet
Descripción	Durante el desarrollo del proyecto hay un corte de suministro de Internet no planificado y puntual.
Impacto	Medio
Probabilidad	Bajo
Plan de mitigación	-
Plan de contingencia	Centrarse en las tareas que no requieren de conexión a Internet como la documentación y actualizar las copias de seguridad cuando esta vuelva.

Tabla 45: Fallo en la conexión a Internet

4.3 Gestión de proceso

En la estimación de las tareas a desarrollar, se tendrá como referencia otros proyectos realizados antes de características similares. Para poder desarrollar el proyecto con calidad y de forma correcta se necesitan adquirir las siguientes habilidades:

- Conocimientos de tecnologías de la parte *Front*: **HTML5, CSS3, JavaScript y jQuery**.
- Conocimientos de tecnologías de la parte *Back*: **PHP**.
- Conocimiento del *framework* **Unity** y **C#**.
- Configuración del proyecto y arranque de la aplicación en el servidor **TomCat** con **WampServer**.

No se necesitará nada adicional, puesto que es un proyecto que forma parte del proceso formativo de los estudios, y por tanto su desarrollo será local (casa particular).

4.3.1 Plan de trabajo

En esta sección, se describen las tareas a desarrollar en cada una de las fases del proyecto, así como su calendarización.

Fase de inicio

En la tabla 46 se muestra la calendarización de las tareas de la fase de inicio, tareas que son descritas en las tablas 47 a 58.

Nombre de tarea	Duración	Comienzo	Fin
Fase inicial	72 horas	Mier. 12/02/2020	Sab. 22/02/2020
Inicio	0 horas	Mier. 12/02/2020	Mier. 12/02/2020
Planteamiento inicial del problema.	1 horas	Mier. 12/02/2020	Mier. 12/02/2020
Toma de requisitos.	2 horas	Mier. 12/02/2020	Mier. 12/02/2020
Análisis de viabilidad del proyecto.	10 horas	Jue. 13/02/2020	Vie. 14/02/2020
Definición de las tareas a desarrollar.	15 horas	Sab. 15/02/2020	Mar. 18/02/2020
Calendarización.	10 horas	Mier. 19/02/2020	Jue. 20/02/2020
Riesgos.	1 horas	Vie. 21/02/2020	Vie. 21/02/2020
Memoria	30 horas	Mier. 12/02/2020	Lun. 14/09/2020
Presupuesto	1 hora	Sab. 22/02/2020	Sab. 22/02/2020
Documento de costes.	1 hora	Sab. 22/02/2020	Sab. 22/02/2020
Preparar el entorno y herramientas.	1 hora	Sab. 22/02/2020	Sab. 22/02/2020
Finalización de la fase de inicio	0 horas	Sab. 22/02/2020	Sab. 22/02/2020

Tabla 46: Calendarización fase de inicio

T-01	Inicio de la fase de Inicio
Descripción	-

Tabla 47: Inicio de la fase de inicio

T-02	Planteamiento inicial del problema.
Descripción	Consiste en la recogida de información de forma general sobre que hará la aplicación y como lo hará, así como una idea general del aspecto que se espera que tendrá una vez finalizada.

Tabla 48: Planteamiento inicial del problema.

T-03	Toma de requisitos
Descripción	Consiste en la elaboración de una primera versión de los requisitos funcionales, no funcionales y de información, para entender el funcionamiento general del sistema.

Tabla 49: Toma de requisitos

T-04	Análisis de viabilidad
Descripción	Búsqueda de información sobre los requisitos funcionales establecidos para determinar si las herramientas seleccionadas son las adecuadas para la elaboración de este proyecto.

Tabla 50: Análisis de viabilidad

T-05	Definición de las tareas a desarrollar
Descripción	Consiste en hacer una lista de las tareas a desarrollar durante todas las fases del proyecto. Incluyendo duración y días en los que se realizarán.

Tabla 51: Definición de las tareas a desarrollar

T-06	Calendarización
Descripción	Esta tarea consiste en la elaboración de los tiempos de las tareas a desarrollar y a las fases del proyecto, estableciendo días de comienzo y de fin, así como duración de las tareas.

Tabla 52: Calendarización

T-07	Gestión de riesgos
Descripción	Elaboración del documento de riesgos. En este se incluye la cuantificación de los riesgos, descripción de los riesgos, planes de contingencia y probabilidad de ocurrencia.

Tabla 53: Gestión de riesgos

T-08	Memoria
Descripción	Elaboración de la memoria del proyecto.

Tabla 54: Memoria

T-09	Presupuesto
Descripción	Elaboración de un presupuesto del proyecto.

Tabla 55: Presupuesto

T-10	Documento de costes
Descripción	Elaboración de un documento de costes reales del proyecto, en el cual se realizará un análisis de los costes asociados al proyecto, y finalmente se obtendrá un valor estimado del coste de todo el desarrollo.

Tabla 56: Documento de costes

T-11	Preparar el entorno y herramientas
Descripción	Consiste en la instalación de Unity y las demás herramientas necesarias para desarrollar la aplicación.

Tabla 57: Preparar el entorno y herramientas

T-12	Finalización de la fase de inicio
Descripción	-

Tabla 58: Finalización de la fase de inicio

Fase de análisis

En la tabla 59 se muestra la calendarización de las tareas de la fase de análisis, tareas que son descritas en las tablas 60 a 68.

Nombre de tarea	Duración	Comienzo	Fin
Fase de análisis	100 horas	Lun. 24/02/2020	Mar. 17/03/2020
Inicio	0 horas	Lun. 24/02/2020	Lun. 24/02/2020
Modelo de casos de uso.	15 horas	Lun. 24/02/2020	Mier. 26/02/2020
Especificación de casos de uso.	10 horas	Jue. 27/02/2020	Vie. 28/02/2020
Modelo de dominio.	5 horas	Sab. 29/02/2020	Sab. 29/02/2020
Investigación de WebGL.	10 horas	Lun. 02/03/2020	Mar. 03/03/2020
Investigación de Unity.	25 horas	Mier. 04/03/2020	Lun. 09/03/2020
Investigación de PHP.	10 horas	Mar. 10/03/2020	Mier. 11/03/2020
Diseño de las vistas UI.	25 horas	Jue. 12/03/2020	Mar. 17/03/2020
Finalización de la fase de análisis	0 horas	Mar. 17/03/2020	Mar. 17/03/2020

Tabla 59: Calendarización fase de análisis

T-13	Inicio de la fase de análisis
Descripción	-

Tabla 60: Inicio de la fase de análisis

T-14	Modelo de casos de uso.
Descripción	Elaboración del diagrama de casos de uso general de la aplicación con los actores involucrados.

Tabla 61: Modelo de casos de uso.

T-15	Especificación de casos de uso.
Descripción	Elaboración de una especificación textual de los casos de usos identificados en el sistema funcionales, no funcionales y de información, para entender el funcionamiento general del sistema.

Tabla 62: Especificación de casos de uso.

T-16	Modelo de dominio.
Descripción	Consiste en obtener el diagrama de todas las entidades del dominio de la aplicación con sus atributos y relaciones.

Tabla 63: Modelo de dominio.

T-17	Investigación de WebGL.
Descripción	Realizar una investigación del funcionamiento básico de WebGL.

Tabla 64: Investigación de WebGL.

T-18	Investigación de Unity.
Descripción	Realizar una investigación exhaustiva del funcionamiento de Unity, realización de pruebas para poder aplicarlo más fácilmente a la aplicación.

Tabla 65: Investigación de Unity.

T-19	Investigación de PHP.
Descripción	Realizar una investigación del funcionamiento de PHP.

Tabla 66: Investigación de PHP.

T-20	Diseño de las vistas UI.
Descripción	Realización de los bocetos de las vistas de la aplicación para su posterior desarrollo.

Tabla 67: Diseño de las vistas UI.

T-21	Finalización de la fase de análisis
Descripción	-

Tabla 68: Finalización de la fase de análisis

Fase de diseño

En la tabla 69 se muestra la calendarización de las tareas de la fase de diseño, tareas que son descritas en las tablas 70 a 74.

Nombre de tarea	Duración	Comienzo	Fin
Fase de diseño	17 horas	Mier. 18/03/2020	Vie. 20/03/2020
Inicio	0 horas	Mier. 18/03/2020	Mier. 18/03/2020
Investigación del patrón de diseño de la aplicación	2 horas	Mier. 18/03/2020	Mier. 18/03/2020
Arquitectura de la aplicación.	10 horas	Mier. 18/03/2020	Jue. 19/03/2020
Diseño de la base de datos.	5 horas	Vie. 20/03/2020	Vie. 20/03/2020
Finalización de la fase de diseño	0 horas	Vie. 20/03/2020	Vie. 20/03/2020

Tabla 69: Calendarización fase de diseño

T-22	Inicio de la fase de diseño
Descripción	-

Tabla 70: Inicio de la fase de diseño

T-23	Investigación del patrón de diseño de la aplicación
Descripción	Realizar una investigación sobre cual seria los patrones de diseño que mas se adapta a la aplicación planteada.

Tabla 71: Investigación del patrón de diseño de la aplicación

T-24	Arquitectura de la aplicación.
Descripción	Consiste en la realización de los diagramas correspondientes a la arquitectura de la aplicación. Diagrama de la arquitectura general, del lado del cliente y del lado del servidor. Planificar la estructura.

Tabla 72: Arquitectura de la aplicación.

T-25	Diseño de la base de datos.
Descripción	Consiste en la realización del diagrama entidad relación y el diagrama relacional de la base de datos de la aplicación.

Tabla 73: Diseño de la base de datos.

T-26	Finalización de la fase de diseño
Descripción	-

Tabla 74: Finalización de la fase de diseño

Fase de implementación

En la tabla 75 se muestra la calendarización de las tareas de la fase de implementación, tareas que son descritas en las tablas 76 a 84.

Nombre de tarea	Duración	Comienzo	Fin
Fase de implementación	213 horas	Sab. 21/03/2020	Vier. 08/05/2020
Inicio	0 horas	Sab. 21/03/2020	Sab. 21/03/2020
Configuración del servidor.	2 horas	Sab. 21/03/2020	Sab. 21/03/2020
Configuración de Unity.	1 hora	Sab. 21/03/2020	Sab. 21/03/2020
Implementación de las vistas de Unity.	50 horas	Sab. 21/03/2020	Mier. 01/04/2020
Implementación de las funcionalidades de las vistas de Unity.	50 horas	Jue. 02/04/2020	Lun. 13/04/2020
Implementación de las vistas de la Web.	50 horas	Mar. 14/04/2020	Vie. 24/04/2020
Implementación de las funcionalidades de las vistas de Web.	50 horas	Sab. 25/04/2020	Mier. 06/05/2020
Implementación de la conexión Unity – Web.	10 horas	Jue. 07/05/2020	Vie. 08/05/2020
Finalización de la fase de implementación.	0 horas	Vie. 08/05/2020	Vie. 08/05/2020

Tabla 75: Calendarización fase de implementación

T-27	Inicio de la fase de implementación
Descripción	-

Tabla 76: Inicio de la fase de implementación

T-28	Configuración del servidor.
Descripción	Consiste en la configuración de la parte servidor con todas las dependencias necesarias, creación de la base de datos y configuración de la misma.

Tabla 77: Configuración del servidor.

T-29	Configuración de Unity.
Descripción	Consiste en la configuración del entorno Unity ajustando la interfaz al gusto y agregado de assets.

Tabla 78: Configuración de Unity.

T-30	Implementación de las vistas de Unity.
Descripción	Consiste en la realización del diagrama entidad relación y el diagrama relacional de la base de datos de la aplicación.

Tabla 79: Implementación de las vistas de Unity.

T-31	Implementación de las funcionalidades de las vistas de Unity.
Descripción	Elaboración de toda la funcionalidad asociada a las escenas, navegación entre escenas, controladores, mostrado de errores y conexión con el servidor.

Tabla 80: Implementación de las funcionalidades de las vistas de Unity.

T-32	Implementación de las vistas de la Web.
Descripción	Consiste en la elaboración de todos los HTML y CSS de las páginas que se diseñaron anteriormente, sin funcionalidad asociada.

Tabla 81: Implementación de las vistas de la Web.

T-33	Implementación de las funcionalidades de las vistas de Web.
Descripción	Elaboración de toda la funcionalidad asociada a las vistas, navegación entre páginas, validación de campos, controladores, mostrado de errores y conexión con el servidor.

Tabla 82: Implementación de las funcionalidades de las vistas de Web.

T-34	Implementación de la conexión Unity – Web.
Descripción	Consiste en la realización de la unión entre los componentes Web y con Unity con mutuo intercambio de información.

Tabla 83: Implementación de la conexión Unity – Web.

T-35	Finalización de la fase de implementación
Descripción	-

Tabla 84: Finalización de la fase de implementación

Fase de pruebas

En la tabla 85 se muestra la calendarización de las tareas de la fase de pruebas, tareas que son descritas en las tablas 85 a 93.

Nombre de tarea	Duración	Comienzo	Fin
Fase de pruebas	120 horas	Sab. 09/05/2020	Vier. 05/06/2020
Inicio	0 horas	Sab. 09/05/2020	Sab. 09/05/2020
Pruebas individuales de funcionalidad de Unity.	25 horas	Sab. 09/05/2020	Jue. 14/05/2020
Pruebas individuales de funcionalidad de la Web.	25 horas	Vie. 15/05/2020	Mier. 20/05/2020
Pruebas de la aplicación completa.	50 horas	Jue. 21/05/2020	Lun. 01/06/2020
Pruebas de carga y descarga de datos.	5 horas	Mar. 02/06/2020	Mar. 02/06/2020
Pruebas de velocidad.	5 horas	Mier. 03/06/2020	Mier. 03/06/2020
Elaborar el manual de usuario.	10 horas	Jue. 04/06/2020	Vier. 05/06/2020
Finalización de la fase de pruebas	0 horas	Vier. 05/06/2020	Vier. 05/06/2020

Tabla 85: Calendarización fase de pruebas

ID-36	Inicio de la fase de pruebas
Descripción	-

Tabla 86: Inicio de la fase de pruebas

ID-37	Pruebas individuales de funcionalidad de Unity.
Descripción	Consiste en la realización de todas las pruebas de funcionamiento de la parte de Unity y la resolución de errores en caso necesario. Asegurarse de que la esta parte funciona correctamente.

Tabla 87: Pruebas individuales de funcionalidad de Unity.

ID-38	Pruebas individuales de funcionalidad de la Web.
Descripción	Consiste en la realización de todas las pruebas de funcionamiento de la parte de la Web y la resolución de errores en caso necesario. Asegurarse de que la esta parte funciona correctamente.

Tabla 88: Pruebas individuales de funcionalidad de la Web.

ID-39	Pruebas de la aplicación completa.
Descripción	Consiste en la realización de todas las pruebas de funcionamiento a nivel general de la aplicación. Asegurarse de que la aplicación esta completa y funciona correctamente.

Tabla 89: Pruebas de la aplicación completa.

ID-40	Pruebas de carga y descarga de datos.
Descripción	Consiste en la realización de las pruebas con acceso a datos asegurarse de que la esta parte funciona correctamente.

Tabla 90: Pruebas de carga y descarga de datos.

ID-41	Pruebas de velocidad.
Descripción	Consiste en la realización de las pruebas de velocidad en los puntos mas relevantes de la aplicación, cargar mundos, cargar minijuegos...

Tabla 91: Pruebas de velocidad.

ID-42	Elaborar el manual de usuario.
Descripción	Consiste en la redacción del manual de uso de la aplicación, para su uso correcto.

Tabla 92: Elaborar el manual de usuario.

ID-43	Finalización de la fase de pruebas
Descripción	-

Tabla 93: Finalización de la fase de pruebas.

Resumen

Fase	Horas
Fase de inicio	72 horas
Fase de análisis	100 horas
Fase de diseño	17 horas
Fase de implementación	213 horas
Fase de pruebas	120 horas
Total	522 horas

Tabla 94: Resumen del plan del proyecto

4.3.2 Seguimiento del plan de trabajo

En este apartado se muestra la calendarización prevista junto a la calendarización real de las tareas realizadas.

Fase de inicio

Nombre de tarea	Duración prevista	Duración real	Comienzo previsto	Comienzo real	Fin previsto	Fin real
Fase inicial	72 horas	72 horas	Mier. 12/02/2020	Mier. 12/02/2020	Sab. 22/02/2020	Sab. 22/02/2020
Inicio	0 horas	0 horas	Mier. 12/02/2020	Mier. 12/02/2020	Mier. 12/02/2020	Mier. 12/02/2020
Planteamiento inicial del problema.	1 horas	1 horas	Mier. 12/02/2020	Mier. 12/02/2020	Mier. 12/02/2020	Mier. 12/02/2020
Toma de requisitos.	2 horas	2 horas	Mier. 12/02/2020	Mier. 12/02/2020	Mier. 12/02/2020	Mier. 12/02/2020
Análisis de viabilidad del proyecto.	10 horas	10 horas	Jue. 13/02/2020	Jue. 13/02/2020	Vie. 14/02/2020	Vie. 14/02/2020
Definición de las tareas a desarrollar.	15 horas	15 horas	Sab. 15/02/2020	Sab. 15/02/2020	Mar. 18/02/2020	Mar. 18/02/2020
Calendarización	10 horas	10 horas	Mier. 19/02/2020	Mier. 19/02/2020	Jue. 20/02/2020	Jue. 20/02/2020
Gestión de riesgos	1 horas	1 horas	Vie. 21/02/2020	Vie. 21/02/2020	Vie. 21/02/2020	Vie. 21/02/2020
Documento de seguimiento.	30 horas	30 horas	Mier. 12/02/2020	Mier. 12/02/2020	Lun. 14/09/2020	Lun. 14/09/2020
Presupuesto	1 hora	1 hora	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020
Documento de costes.	1 hora	1 hora	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020
Preparar el entorno y herramientas	1 hora	1 hora	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020
Finalización de la fase de inicio	0 horas	0 días	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020	Sab. 22/02/2020

Tabla 95: Calendarización real de la fase de inicio

Fase de análisis

Nombre de tarea	Duración prevista	Duración real	Comienzo previsto	Comienzo real	Fin previsto	Fin real
Fase de análisis	100 horas	95 horas	Lun. 24/03/2020	Lun. 24/03/2020	Mar. 17/03/2020	Sab. 14/03/2020
Inicio	0 horas	0 horas	Lun. 24/02/2020	Lun. 24/02/2020	Lun. 24/02/2020	Lun. 24/02/2020
Modelo de casos de uso.	15 horas	15 horas	Lun. 24/02/2020	Lun. 24/02/2020	Mier. 26/02/2020	Mier. 26/02/2020
Especificación de casos de uso.	10 horas	15 horas	Jue. 27/02/2020	Jue. 27/02/2020	Vie. 28/02/2020	Sab. 29/02/2020
Modelo de dominio.	5 horas	5 horas	Sab. 29/02/2020	Dom. 01/03/2020	Sab. 29/02/2020	Dom. 01/03/2020
Investigación de WebGL.	10 horas	10 horas	Lun. 02/03/2020	Lun. 02/03/2020	Mar. 03/03/2020	Mar. 03/03/2020
Investigación de Unity.	25 horas	25 horas	Mier. 04/03/2020	Mier. 04/03/2020	Lun. 09/03/2020	Lun. 09/03/2020
Investigación de PHP.	10 horas	0 horas	Mar. 10/03/2020	Mar. 10/03/2020	Mier. 11/03/2020	Mar. 10/03/2020
Diseño de las vistas UI.	25 horas	25 horas	Jue. 12/03/2020	Mar. 10/03/2020	Mar. 17/03/2020	Sab. 14/03/2020
Finalización de la fase de análisis	0 días	0 horas	Mar. 17/03/2020	Sab. 14/03/2020	Mar. 17/03/2020	Sab. 14/03/2020

Tabla 96: Calendarización real de la fase de análisis

Fase de diseño

Nombre de tarea	Duración prevista	Duración real	Comienzo previsto	Comienzo real	Fin previsto	Fin real
Fase de diseño	17 horas	17 horas	Mier. 18/03/2020	Lun. 16/03/2020	Vie. 20/03/2020	Mier. 18/03/2020
Inicio	0 horas	0 horas	Mier. 18/03/2020	Lun. 16/03/2020	Mier. 18/03/2020	Lun. 16/03/2020
Investigación del patrón de diseño de la aplicación	2 horas	2 horas	Mier. 18/03/2020	Lun. 16/03/2020	Mier. 18/03/2020	Lun. 16/03/2020
Arquitectura de la aplicación.	10 horas	10 horas	Mier. 18/03/2020	Lun. 16/03/2020	Jue. 19/03/2020	Mar. 17/03/2020
Diseño de la base de datos.	5 horas	5 horas	Vie. 20/03/2020	Mier. 18/03/2020	Vie. 20/03/2020	Mier. 18/03/2020
Finalización de la fase de diseño	0 horas	0 horas	Vie. 20/03/2020	Mier. 18/03/2020	Vie. 20/03/2020	Mier. 18/03/2020

Tabla 97: Calendarización real de la fase de diseño

Fase de implementación

Nombre de tarea	Duración prevista	Duración real	Comienzo previsto	Comienzo real	Fin previsto	Fin real
Fase de implementación	213 horas	388 horas	Sab. 21/03/2020	Jue. 19/03/2020	Vier. 08/05/2020	Mar. 16/06/2020
Inicio	0 horas	0 días	Sab. 21/03/2020	Jue. 19/03/2020	Sab. 21/03/2020	Jue. 19/03/2020
Configuración del servidor.	2 horas	2 horas	Sab. 21/03/2020	Jue. 19/03/2020	Sab. 21/03/2020	Jue. 19/03/2020
Configuración de Unity.	1 hora	1 hora	Sab. 21/03/2020	Jue. 19/03/2020	Sab. 21/03/2020	Jue. 19/03/2020
Implementación de las vistas de Unity.	50 horas	100 horas	Sab. 21/03/2020	Jue. 19/03/2020	Mier. 01/04/2020	Vie. 10/04/2020
Implementación de las funcionalidades de las vistas de Unity.	50 horas	75 horas	Jue. 02/04/2020	Sab. 11/04/2020	Lun. 13/04/2020	Mar. 28/04/2020
Implementación de las vistas de la Web.	50 horas	80 horas	Mar. 14/04/2020	Mier. 29/04/2020	Vie. 24/04/2020	Sab. 16/05/2020
Implementación de las funcionalidades de las vistas de Web.	50 horas	80 horas	Sab. 25/04/2020	Lun. 18/05/2020	Mier. 06/05/2020	Jue. 04/06/2020
Implementación de la conexión Unity – Web.	10 horas	50 horas	Jue. 07/05/2020	Vie. 05/06/2020	Vie, 08/05/2020	Mar. 16/06/2020
Finalización de la fase de implementación	0 días	0 días	Vie. 08/05/2020	Mar. 16/06/2020	Vie. 08/05/2020	Mar. 16/06/2020

Tabla 98: Calendarización real de la fase de implementación

Fase de pruebas

Nombre de tarea	Duración prevista	Duración real	Comienzo previsto	Comienzo real	Fin previsto	Fin real
Fase de pruebas	120 horas	80 horas	Sab. 09/05/2020	Mier. 17/06/2020	Vier. 05/06/2020	Lun. 06/07/2020
Inicio	0 horas	0 horas	Sab. 09/05/2020	Mier. 17/06/2020	Sab. 09/05/2020	Mier. 17/06/2020
Pruebas individuales de funcionalidad de Unity.	25 horas	5 horas	Sab. 09/05/2020	Jue. 18/06/2020	Jue. 14/05/2020	Jue. 18/06/2020
Pruebas individuales de funcionalidad de la Web.	25 horas	5 horas	Vie. 15/05/2020	Vie. 19/06/2020	Mier. 20/05/2020	Vie. 19/06/2020
Pruebas de la aplicación completa.	50 horas	50 horas	Jue. 21/05/2020	Sab. 20/06/2020	Lun. 01/06/2020	Mier. 01/07/2020
Pruebas de carga y descarga de datos.	5 horas	5 horas	Mar. 02/06/2020	Jue. 02/07/2020	Mar. 02/06/2020	Jue. 02/07/2020
Pruebas de velocidad.	5 horas	5 horas	Mier. 03/06/2020	Vie. 03/07/2020	Mier. 03/06/2020	Vie. 03/07/2020
Elaborar el manual de usuario.	10 horas	10 horas	Jue. 04/06/2020	Sab. 04/07/2020	Vier. 05/06/2020	Lun. 06/07/2020
Finalización de la fase de pruebas	0 horas	0 horas	Vier. 05/06/2020	Lun. 06/07/2020	Vier. 05/06/2020	Lun. 06/07/2020

Tabla 99: Calendarización real de la fase de pruebas

Resumen

Fase	Horas previstas	Horas reales
Fase de inicio	72 horas	72 horas
Fase de análisis	100 horas	95 horas
Fase de diseño	17 horas	17 horas
Fase de implementación	213 horas	388 horas
Fase de pruebas	120 horas	80 horas
Total	522 horas	652 horas

Tabla 100: Resumen del plan del proyecto

4.3.3 Desviaciones

Durante el desarrollo de las distintas fases del proyecto ha habido algunas desviaciones en la duración prevista.

Estas desviaciones se han producido en gran parte por el riesgo R06 (Conocimiento insuficiente sobre las tecnologías), en especial **Unity** con **C#** pues, si bien es un lenguaje sencillo de aprender, se complica cuando intentas salir de este entorno y se agrava dependiendo de lo detallado que se quiera que sea lo desarrollado. En el entorno web ha ocurrido algo similar pero en menor medida debido a que los lenguajes utilizados eran conocidos exceptuando el lenguaje **PHP**.

Otro riesgo que se ha dado ha sido el R07 (cambios en los requisitos), que si bien la variación de estos no han sido muy grandes, han supuesto cambios en bastantes ficheros.

4.4 Presupuesto

El proyecto contará con una persona como desarrollador, y además se necesitarán otra serie de recursos para poder realizarlo, entre ellas, un equipo y una serie de herramientas. El presupuesto se tomara de forma global, es decir, el presupuesto se realizará en función de la cantidad total de horas que lleve desarrollarlo independientemente de los actores que intervengan en su desarrollo. En esta sección se describen todos los recursos necesarios para desarrollar el proyecto.

- Recursos de personal: En este caso el desarrollo lo realizará una persona.
- Equipos y herramientas: Se necesitarán las siguientes herramientas para el desarrollo:
 - Ordenador (portátil o de sobremesa).
 - Sistema Operativo **Windows 10**.
 - **Astah Professional**.
 - **Google Drive** o similar.
 - **IDE Unity**.
 - **Notepad++** o similar.
 - **MySQL** o **MariaDB**.
 - **WampServer 3.2**
- Espacio: Principalmente, la aplicación se desarrollara en casa.
- Tiempo: En este caso, acorde con la planificación, necesitarían 522 horas y acorde con el desarrollo real, se han utilizado 652 horas.

En base a todos estos recursos necesarios, se puede obtener un coste final estimado para el desarrollo del proyecto.

El tiempo estimado necesario proviene de dividir en horas la planificación que se ha desarrollado con anterioridad en la calendarización.

Dado que el proyecto va a ser desarrollado por una persona con un año de experiencia, el coste se calculará a partir del sueldo de un desarrollador junior (18.000€ brutos anuales). Este es el dato que se ha utilizado para los cálculos de la tabla 101.

Equipos y herramientas	Ordenador SO Windows 10. Astah Professional. Google Dirive o similar. IDE Unity. Notpad++ o similar. MySQL o MariaDB. WampServer 3.2	El equipo tendrá coste 0€ por ya haberlo amortizado en proyectos anteriores. Lo mismo ocurre con el sistema operativo del equipo y Astah. El resto de herramientas tiene un coste 0€ , puesto que se usará la versión libre.
Espacio	Local (casa particular)	0€
Tiempo	Desarrollador	<u>Coste previsto</u> $(18.000€ / 253 \text{ días laborables} / 5 \text{ horas/día}) * 522 \text{ horas} = \mathbf{7427,67€}$ <u>Coste real</u> $(18.000€ / 253 \text{ días laborables} / 5 \text{ horas/día}) * 652 \text{ horas} = \mathbf{9277,47€}$
Total	Coste previsto: 7427,67€ Coste real: 9277,47€	

Tabla 101: Plan de costes del proyecto

En la tabla 101 se encuentra el resumen de costes de los recursos necesarios para el desarrollo del proyecto. En total, los costes previstos eran de 7427,67€ pero, basándose en el desarrollo real, los costes para desarrollar este proyecto seria de 9277,47€. Aunque el coste real es de 0€ ya que se trata de un proyecto de fin de carrera.

Capítulo 5

Diseño

Después de realizar el Análisis, el siguiente paso en el proceso, es el Diseño de la aplicación. A continuación, se detallan las decisiones de diseño tomadas, la arquitectura del sistema, subsistemas de la aplicación, relaciones entre ellos, y la explicación de diferentes tecnologías que se usarán en el desarrollo.

5.1 Patrón MVC

Para el diseño de la aplicación se ha seguido el patrón arquitectónico MVC (modelo vista controlador), que se basa en separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos con responsabilidades bien diferenciadas.

- **Modelo:** Contiene una representación de los datos que maneja el sistema, su lógica de negocio y sus mecanismos de persistencia. El modelo es responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. El modelo define las reglas de negocio (la funcionalidad del sistema).
- **Vista:** O interfaz de usuario, compone la información que se envía al cliente y los mecanismos de interacción con este. Las vistas son responsables de recibir datos del modelo y mostrarlos al usuario. O de mandar información a su controlador asociado para solicitar datos al modelo.
- **Controlador:** Actúa como intermediario entre el Modelo y la vista, gestionando el flujo de información entre ellos, las transformaciones para adaptar los datos a las necesidades de cada uno y las comprobaciones o validaciones. El controlador es responsable de recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.). Contiene reglas de gestión de eventos, que pueden suponer peticiones al modelo o a las vistas [9].

La gran ventaja que posee esta técnica de programación es que permite modificar cada uno de ellos sin necesidad de modificar los demás, lo que permite desarrollar aplicaciones modulares y escalables que se puedan actualizar fácilmente y añadir o eliminar nuevos módulos o funcionalidades de forma paquetizada, ya que cada “paquete” utiliza el mismo sistema con sus vistas, modelos y controladores [10].

5.2 Arquitectura del sistema

5.2.1 Arquitectura lógica

En la figura 5 se encuentra la arquitectura lógica definida para la aplicación, en este caso se ha optado por una arquitectura Cliente-Servidor. Se dispone de una capa de presentación o interfaz, que representa el lado del Cliente, y por otro lado el servidor que contiene toda la lógica de negocio.

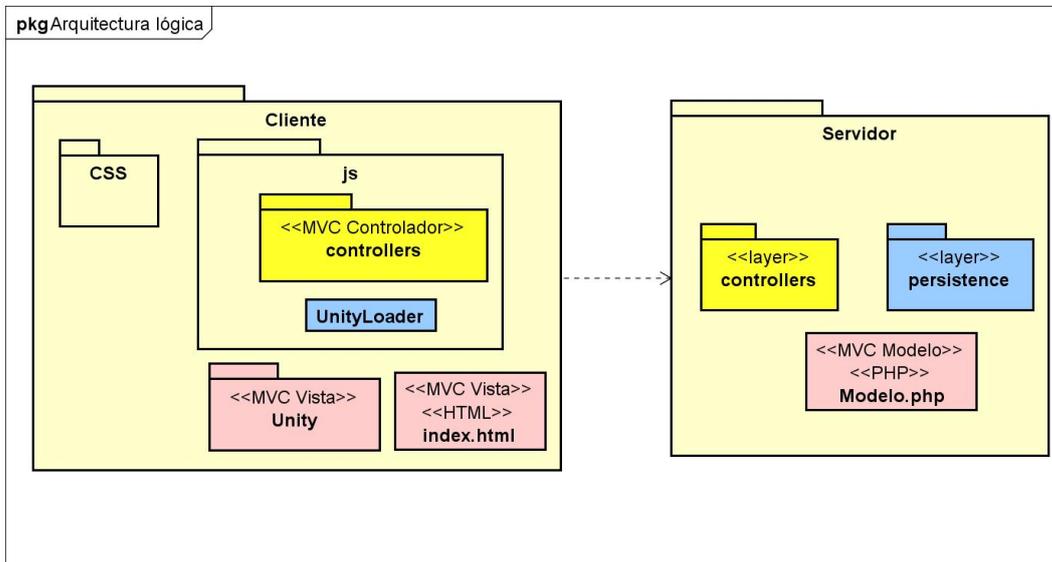


Figura 5: Arquitectura lógica

5.2.2 Despliegue

En la figura 6 se representa el diagrama de despliegue de la aplicación. Se cuenta con una arquitectura cliente-servidor. En el caso particular de este proyecto, se tiene la aplicación desplegada en un servidor que usa **Apache Tomcat**, que se conecta con **MySQL** para acceder a la base de datos. Por otro lado el cliente se conecta al servidor a través de un navegador web por **HTTP**.

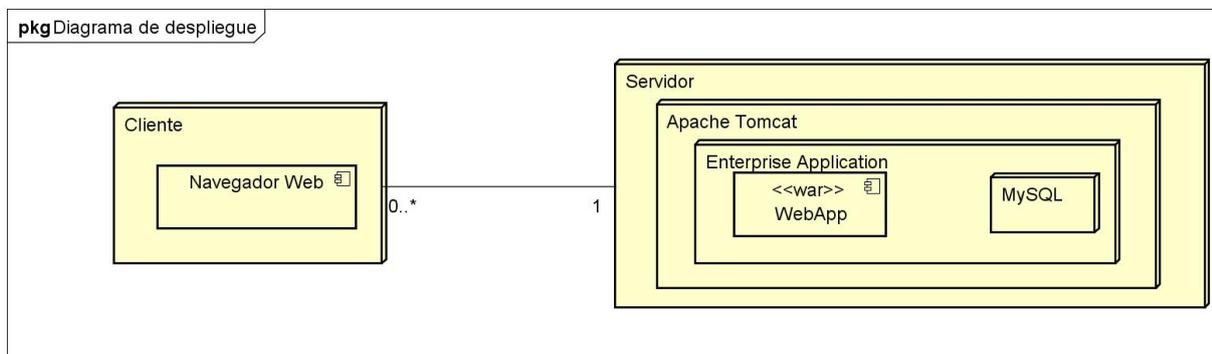


Figura 6: Despliegue

5.2.3 Arquitectura cliente

En la figura 7 se muestra la arquitectura en la parte Cliente. En este caso se esta usando el patrón MVC:

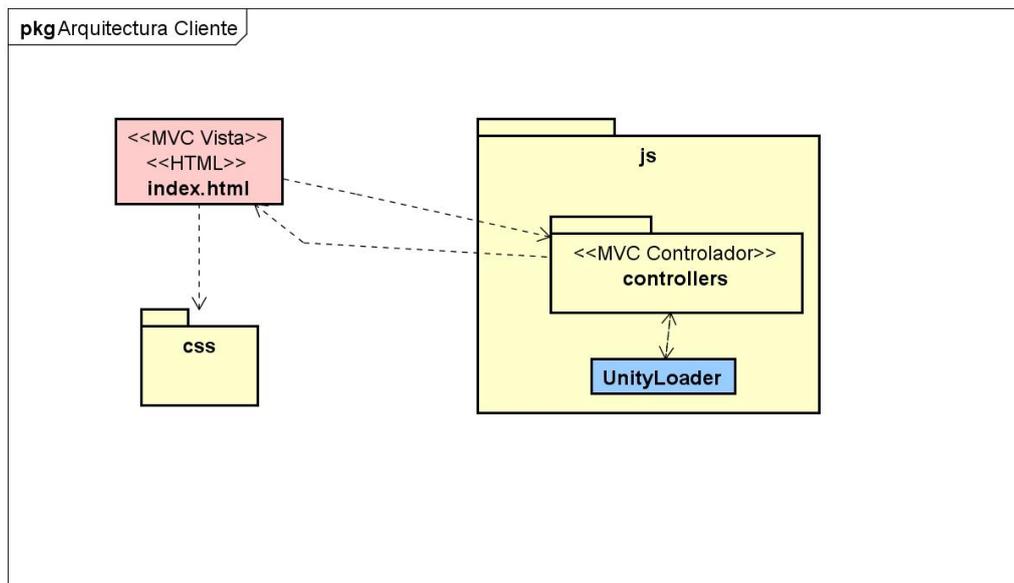


Figura 7: Arquitectura cliente

5.2.4 Arquitectura servidor

En la figura 8, se muestra la arquitectura en la parte Servidor. En ella aparece el conjunto de *controllers* que incluye los controladores que reciben las peticiones del cliente, con la información obtenida del modelo estos se conectan con *persistence* encargado de la conexión con la base de datos.

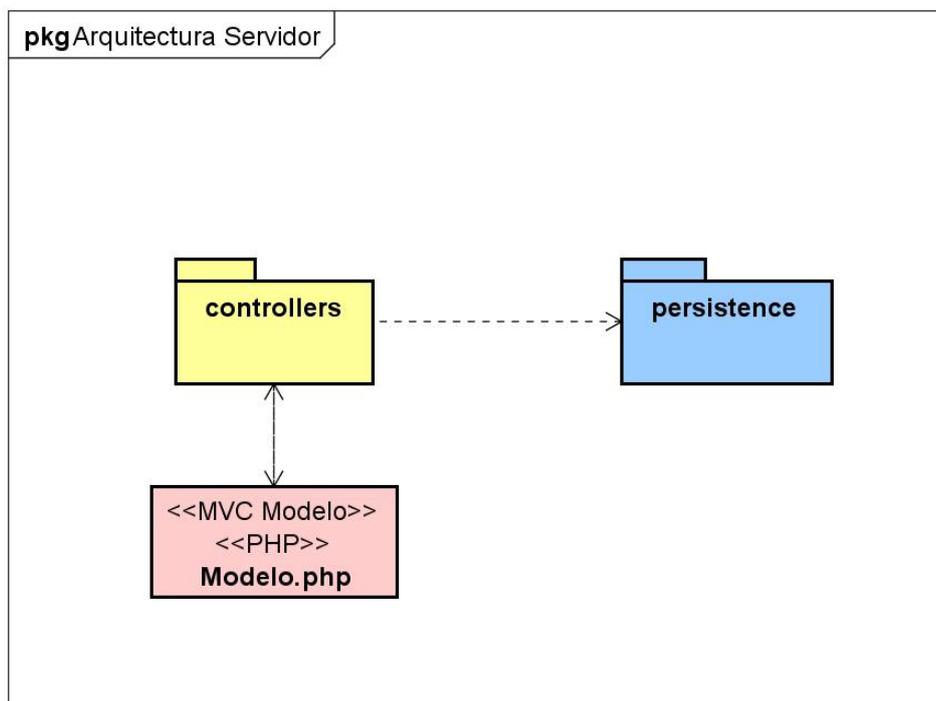


Figura 8: Arquitectura servidor

5.3 Modelo de datos

5.3.1 Modelo entidad-relación

En la figura 9, se incluye el modelo entidad-relación de la base de datos. En este diagrama se representan las tablas, las relaciones entre ellas y los campos necesarios para la aplicación.

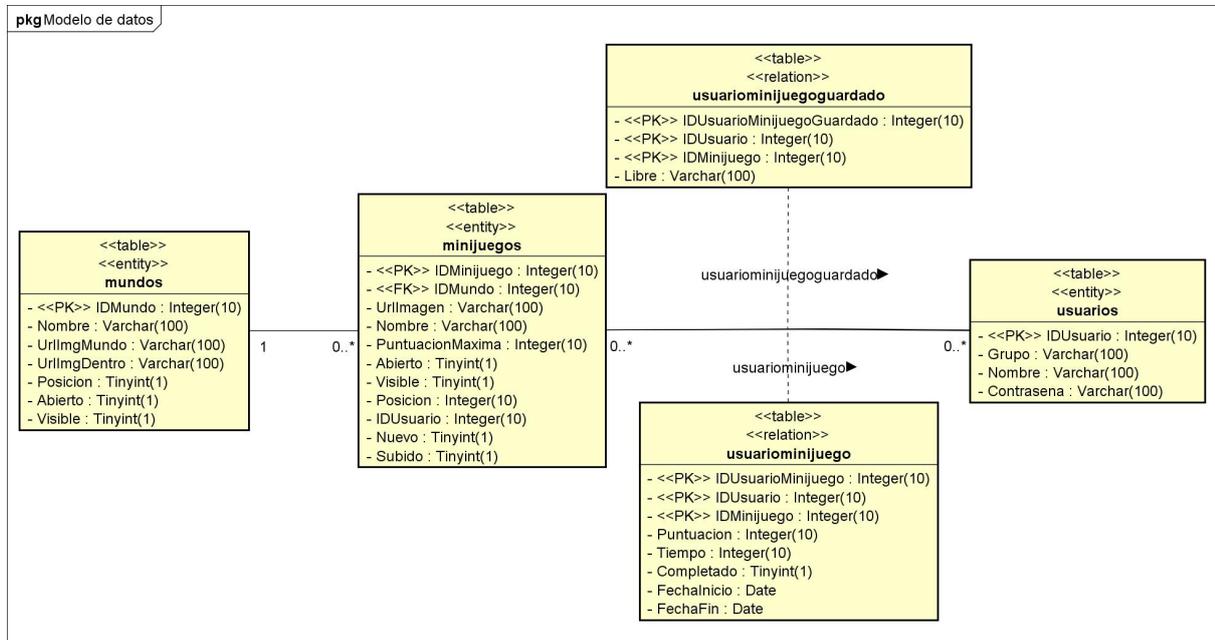


Figura 9: Modelo entidad-relacion

5.3.2 Modelo relacional

En la figura 10, se incluye el modelo relacional de la base de datos.



Figura 10: Modelo relacional

5.4 Realización en diseño de casos de uso

En este punto, se detalla la comunicación entre las diferentes capas del lado del cliente y del servidor para realizar un caso de uso concreto mediante diagramas de secuencia. A continuación se muestran los diagramas de realización de la mayoría de los casos de uso. No se han incluido todos porque algunos son muy similares, aunque se indicarán en cada caso los casos de uso que comparten diagrama.

5.4.1 CU-01: Identificarse

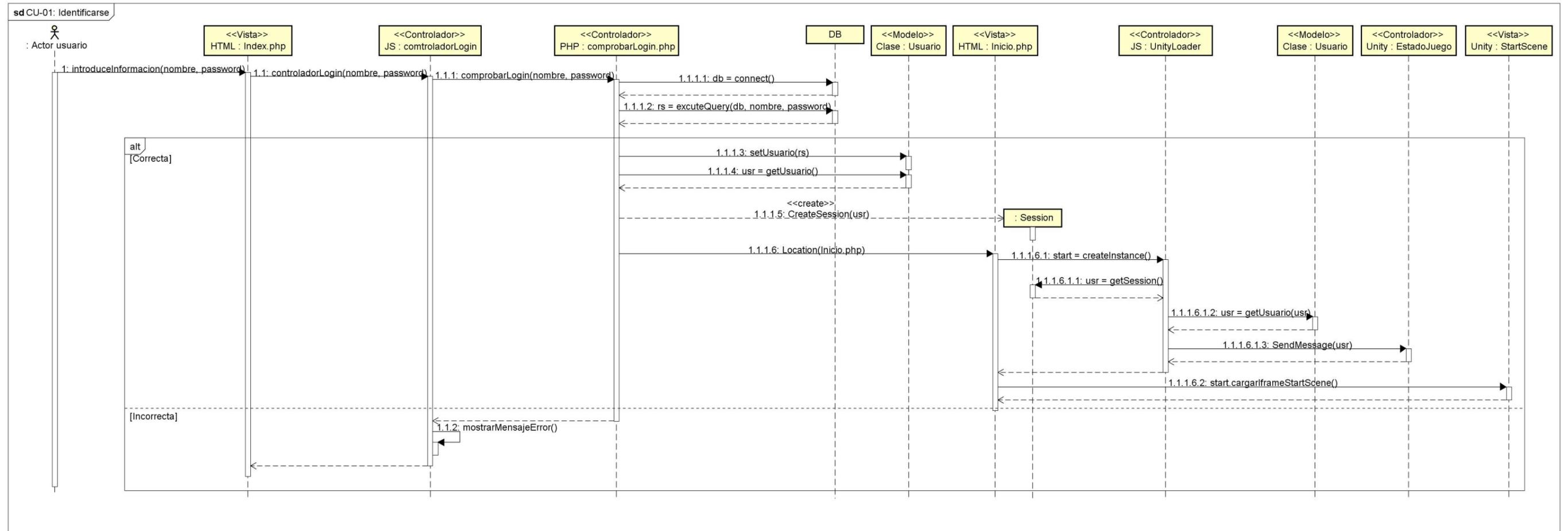


Figura 11: Caso de uso identificarse

5.4.2 CU-02 Cerrar sesión

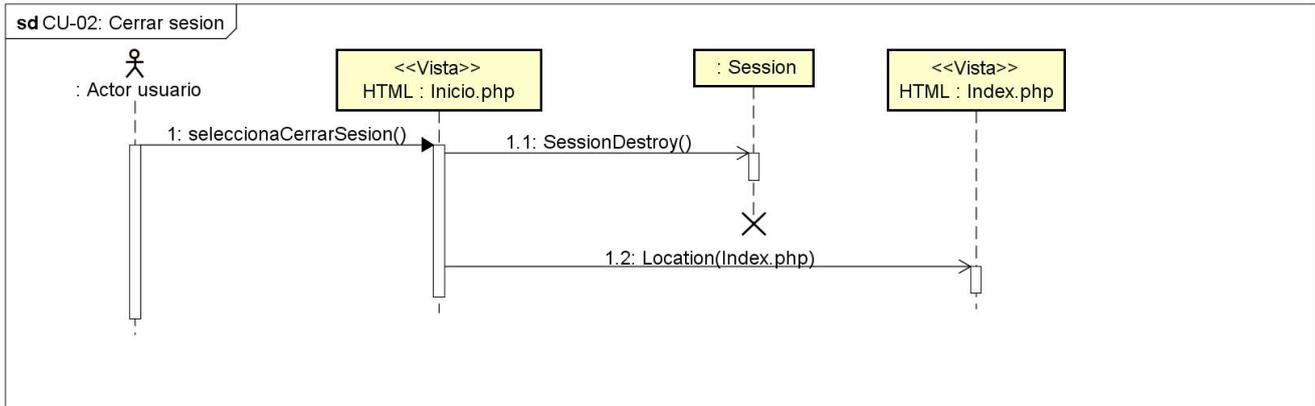


Figura 12: Caso de uso cerrar sesión

5.4.3 CU-03 Mostrar puntuación

En este diagrama se reflejan los casos de uso de mostrar puntuaciones grupo (CU-04), revisar minijuego (CU-10), revisar mundos y minijuegos (CU-14) y revisar usuarios(CU-30).

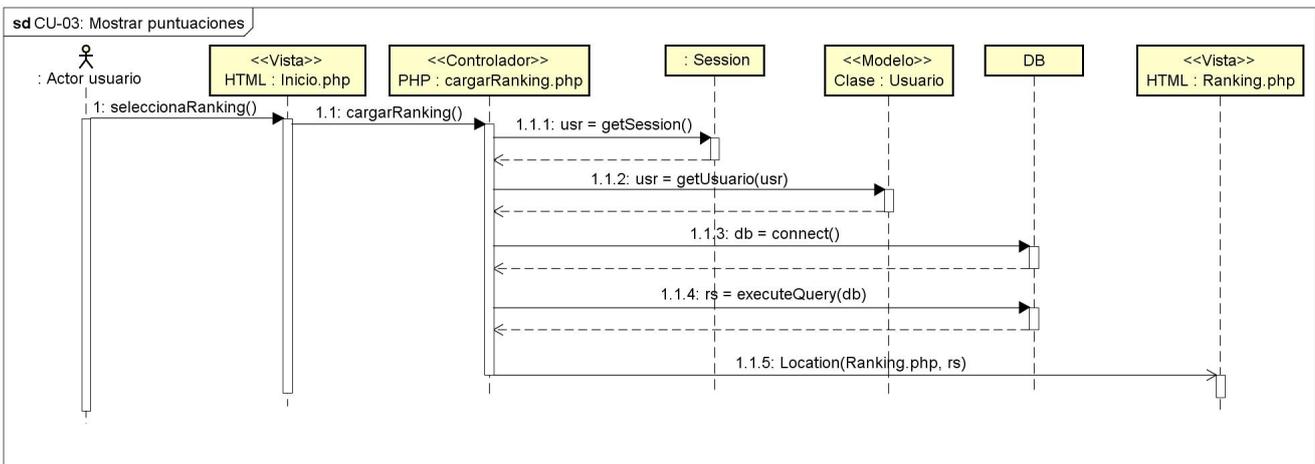


Figura 13: Mostrar puntuación

5.4.4 CU-05 Mostrar mundos

En este diagrama se refleja el caso de uso de mostrar minijuegos (CU-06).

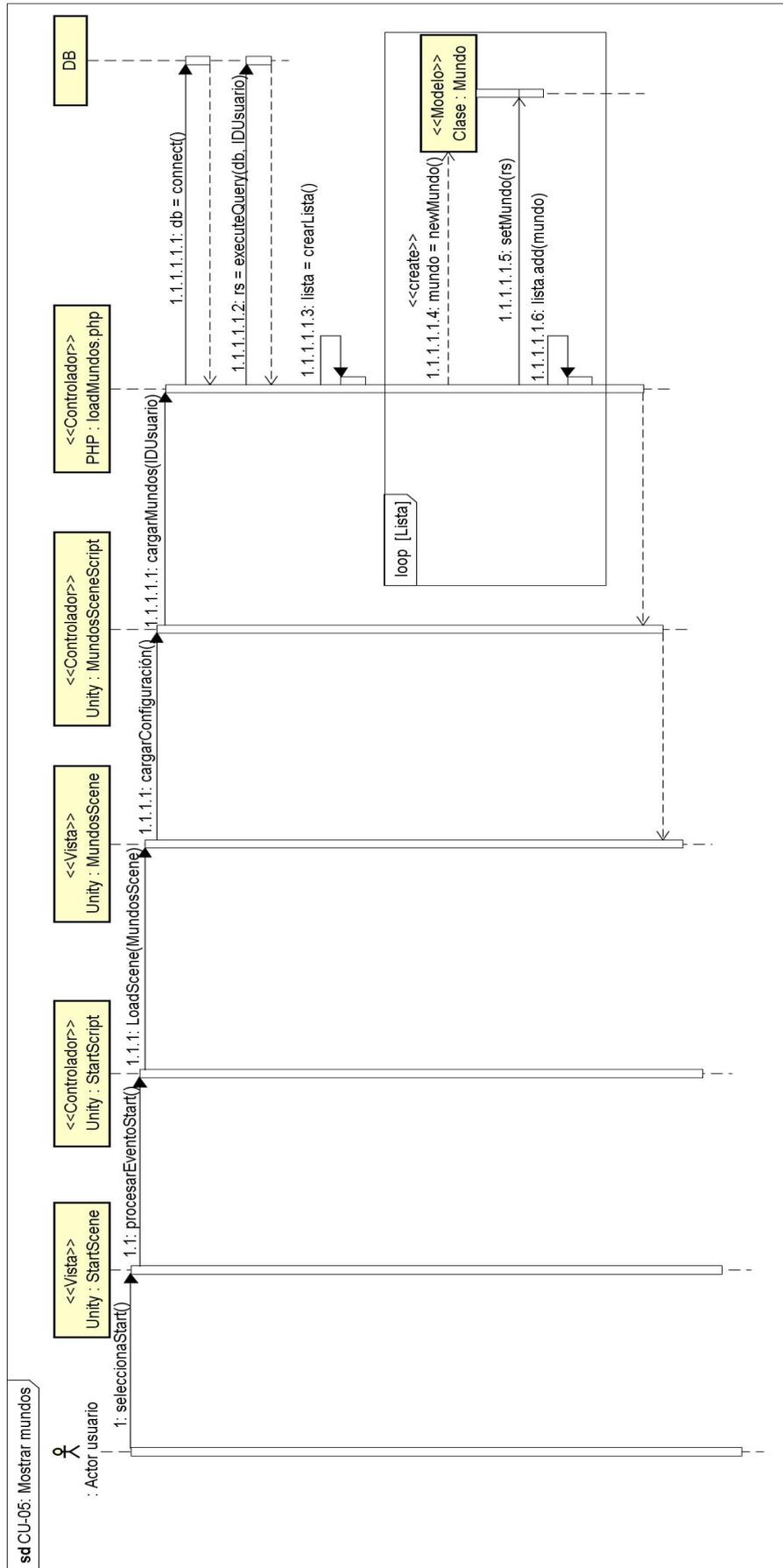


Figura 14: Mostrar mundos

5.4.5 CU-07 Jugar minijuego

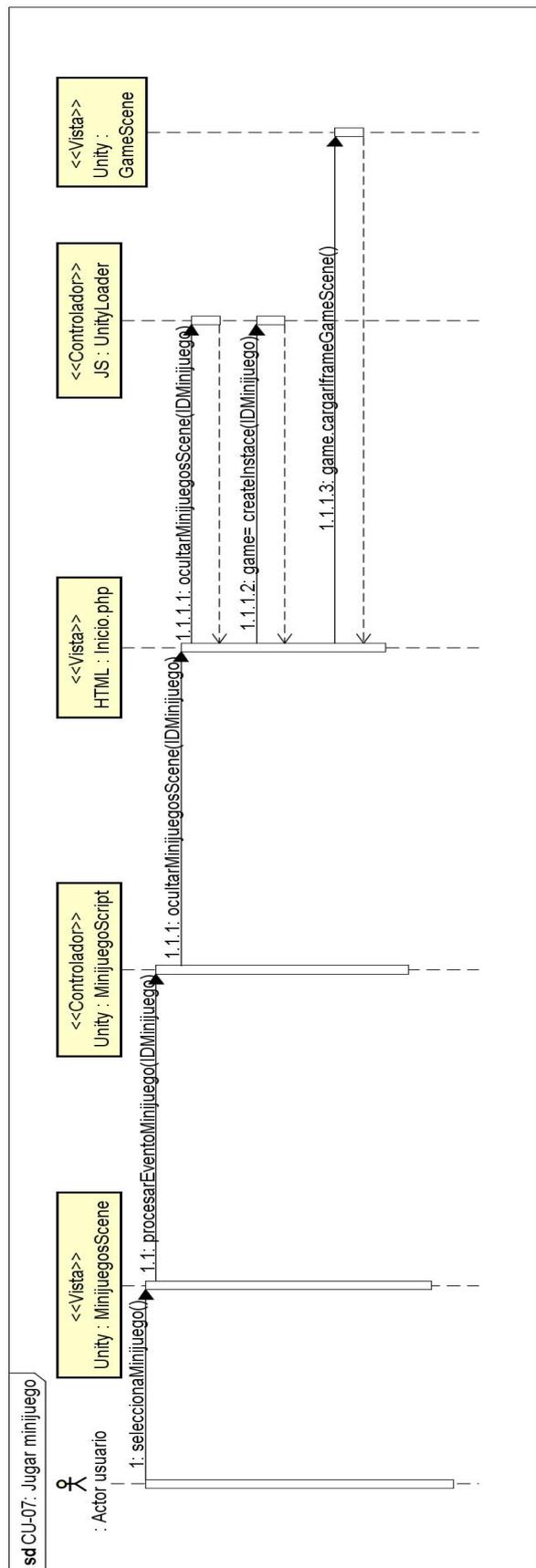


Figura 15: Jugar minijuego

5.4.6 CU-08 Añadir minijuego

En este diagrama se reflejan los casos de uso de añadir proyecto al minijuego (CU-09), modificar minijuego (CU-11), modificar proyecto del minijuego (CU-12), añadir mundo (CU-20), modificar mundo (CU-21), añadir usuario (CU-28), añadir lista de usuarios (CU-29) y modificar usuario (CU-31).

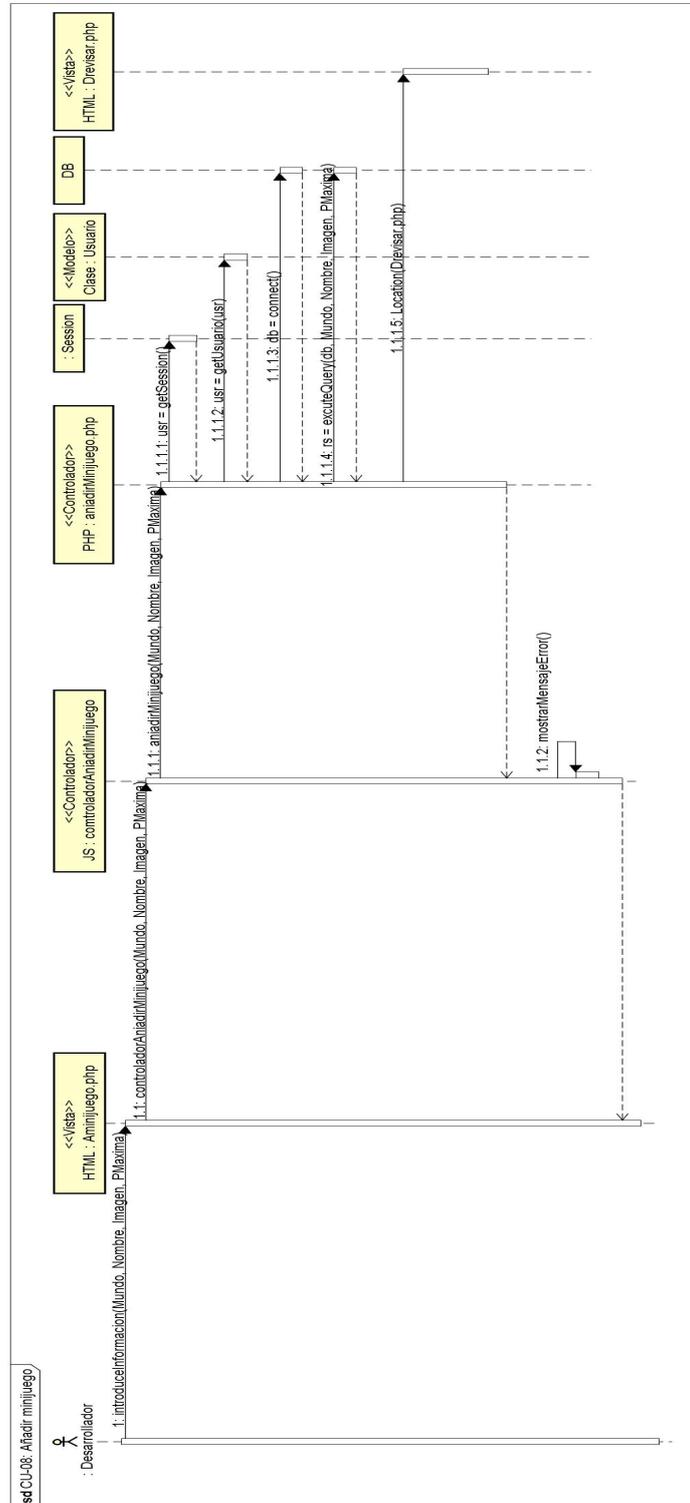


Figura 16: Añadir minijuego

5.4.7. CU-12 Eliminar minijuego

En este diagrama se reflejan los casos de uso abrir mundo (CU-25), cerrar mundo (CU-13), hacer visible mundo (CU-23), ocultar mundo (CU-24), cambiar posición mundo (CU-27), abrir minijuego (CU-17), cerrar minijuego (CU-18), hacer visible minijuego (CU-15), ocultar minijuego (CU-16), cambiar posición minijuego (CU-19), eliminar mundo (CU-22), eliminar usuario (CU-32) y eliminar todos los usuarios (CU-33).

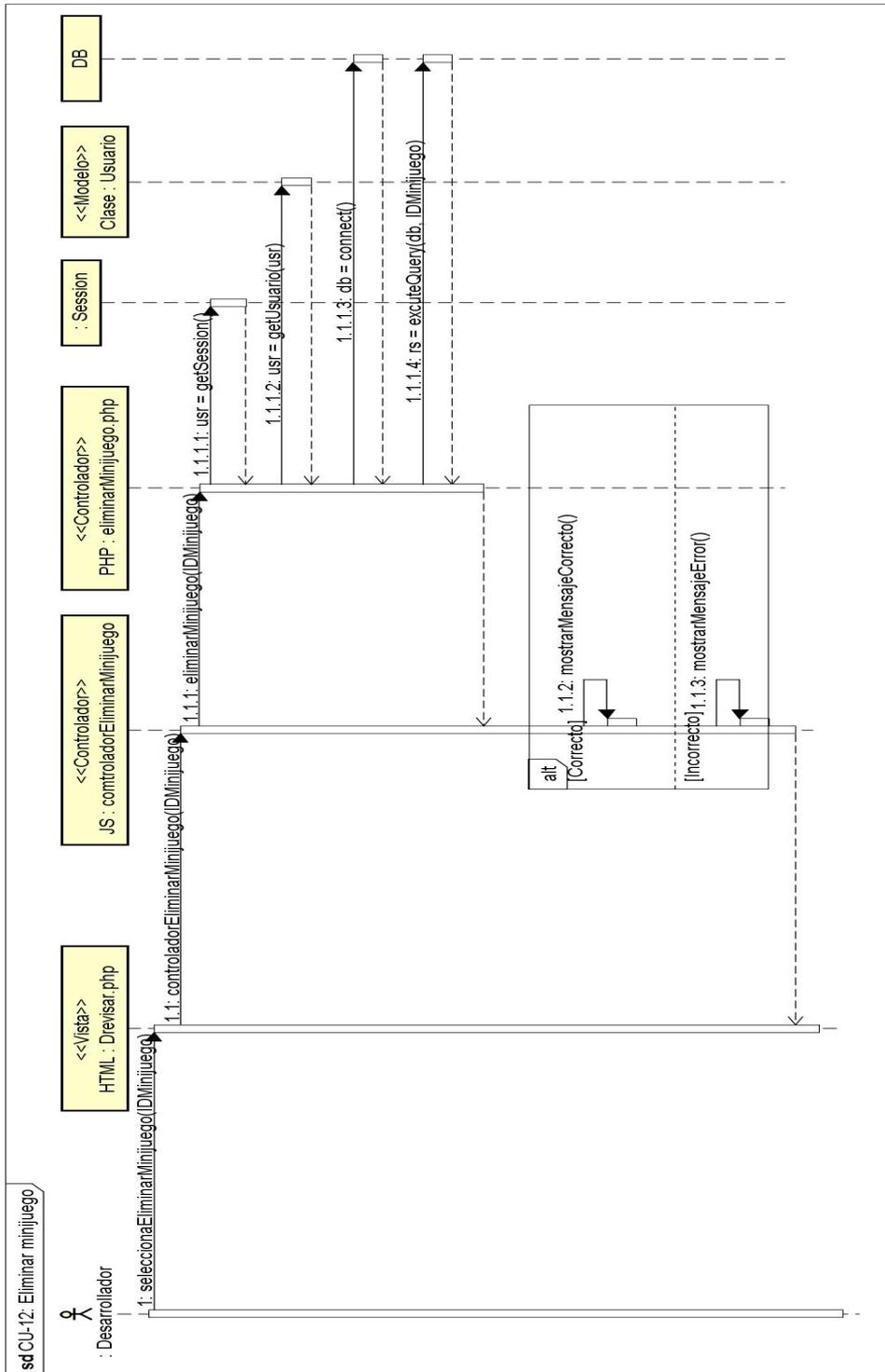


Figura 17: Eliminar minijuego

Capítulo 6

Implementación

6.1 Entorno de desarrollo

Se ha utilizado **Unity** junto a **Visual Studio** para el desarrollo de la parte gráfica de la aplicación y para la parte web se ha utilizado el editor de código fuente gratuito **Notepad++** y el entorno de desarrollo web **WampServer** simulando un entorno cliente-servidor en local.

6.2 Implementación de la Base de Datos

Como Sistema Gestor de la Base de Datos (**SGBD**), se ha optado por **MySQL** [11] en su versión 8 proporcionada por **WampServer** 3.2.

6.3 Copias de seguridad

Se ha utilizado un programa propio para la realización de copias de seguridad. El programa copia todos los archivos especificados a una carpeta con fecha y hora en una carpeta compartida de **google drive**, cada 24 horas.

6.4 Organización del código

Al ser una plataforma web todos los archivos y directorios están en la carpeta pública del servidor. Dentro del directorio público del servidor se pueden encontrar los subdirectorios:

- **css**: guarda los archivos **CSS** que se usan en toda la aplicación.
- **images**: guarda todas las imágenes usadas en la aplicación incluidas la de los minijuegos y mundos.
- **minijuegos**: guarda los proyectos de los minijuegos que cargará la aplicación.
- **texto**: guarda el fichero con la lista de usuarios introducida.
- **unity**: guarda el proyecto **Unity** principal.

6.5 Integración de elementos

Durante el desarrollo de este proyecto han surgido diversos problemas, algunos con poca relevancia y sencillos de resolver y otros no tanto, que han provocado la ampliación temporal del proyecto. En este apartado se expondrán esos problemas así como su posterior resolución.

6.5.1 Configuración del servidor

A pesar de que la configuración del servidor parecía sencilla inicialmente, ha generado algunos problemas a partir de su instalación que se detallan en las tablas 102 y 103.

WampServer necesita para su funcionamiento estándar que sus tres servicios básicos (Apache **Tomcat**, **PHP** y **SQL**) estén activos, esto lo muestra con un código de colores en su icono (Verde: todos activos, Naranja: uno o dos inactivos o Rojo: todos los servicios parados). Esta característica ha facilitado la detección del problema detallado en la tabla 101.

ID-01	Problema con los servicios
Tiempo invertido	Bajo.
Problema	El problema radicaba en el servicio del SQL pues en mi caso ya disponía de un MySQL instalado en mi maquina por lo que por defecto ambos SQL estaban escuchando en el mismo puerto.
Solución	Configurar el servidor cambiando el puerto de instalación y escucha de la base de datos.
Solución alternativa	Asociar la base de datos ya instalada previamente al WampServer para que utilice esta en lugar de instalar la de por defecto en el servidor.

Tabla 102: Problema con los servicios

Otro problema relacionado con la configuración del servidor es que la aplicación requiere de poder transferir archivos al servidor con un peso elevado.

ID-02	Problema con imágenes y archivos pesados
Tiempo invertido	Alto.
Problema	El problema radicaba en que WampServer tiene por defecto una limitación máxima de ficheros y de peso que puede subir. Esta limitación no genera ningún aviso en el código donde se esta ejecutando por lo que detectar el problema ha sido complicado.
Solución	Configurar el servidor aumentando la limitación máxima por defecto.
Solución alternativa	Subir estos archivos pesado por otros métodos como por ejemplo FireZilla.

Tabla 103: Problema con imágenes y archivos pesados

6.5.2 Comunicación Web – Unity

Como ya se ha comentado a lo largo de este proyecto, **Unity** es muy rígido cuando se saca de su entorno lo que ha generado dificultades mas importantes que se han dado en este proyecto. De hecho, esta característica ha provocado un problema de tal envergadura que ha llegado a amenazar la consecución de uno de los objetivos principales de este proyecto, que era lograr la conexión entre la parte **Unity** y la parte Web.

Posiblemente Unity se podría encargar de todo, pero eso supondría usar su plataforma y la opción de almacenamiento en la nube. Los problemas de esta opción son: no se está seguro de que se pueda hacer todo lo que queremos, lleva coste (se tendría que usar su plataforma multijugador) y se pierde el control del servidor, que puede ser importante para la gestión de la información almacenada.

Inicialmente, este proyecto se estructuró a partir de dos grandes bloques, uno Web y otro **Unity**, unidos a través de un controlador (esta estructura se detalla esquemáticamente en la figura 18). Este controlador diseñado como un fichero **JavaScript** sería el encargado de decidir que se mostraría en cada momento, la Web o **Unity**. En este punto surge el principal problema de desarrollo de la aplicación.

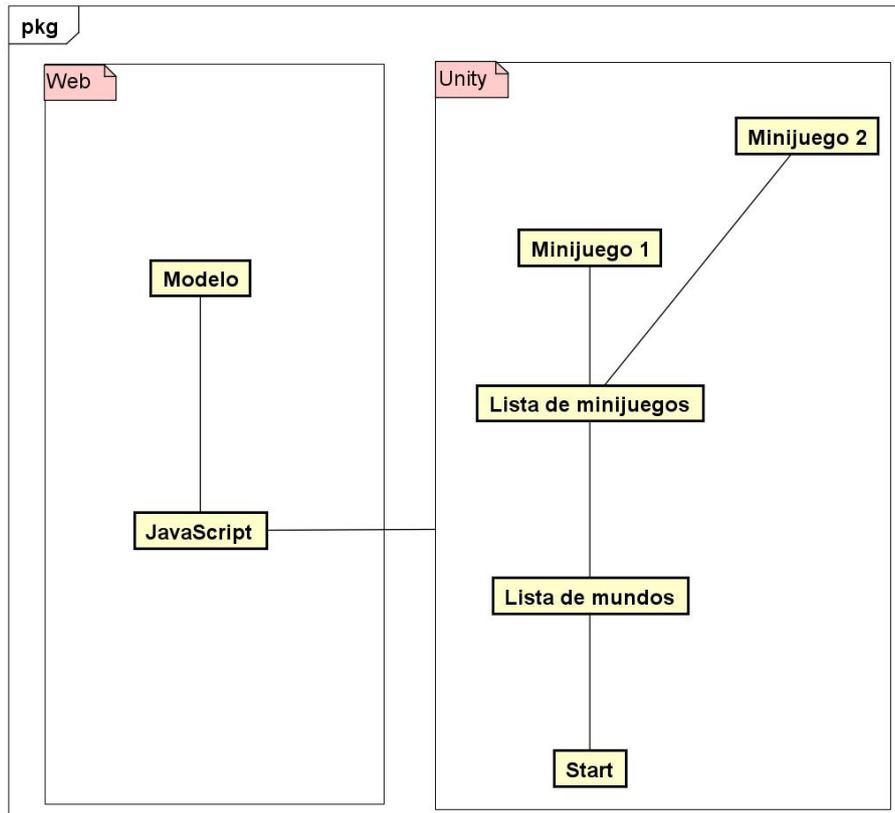


Figura 18: Comunicación Web – Unity, planteamiento inicial.

Al tratarlos como dos bloques independientes, una vez exportado el bloque **Unity** si se desea añadir un nuevo minijuego se necesitaría:

- Parar el servicio.
- Incorporar a **Unity** ese nuevo minijuego y compilarlo.
- Añadir el código generado a la aplicación web.

Estos requisitos no son deseables ya que la intención original era que la aplicación gestionase la administración de los minijuegos de forma independiente sin necesidad de parar el servicio. Para solventar este problema se decidió descartar la idea de los dos bloques separados, de modo que el nuevo planteamiento constaba de un **HTML** que a través de un elemento “*iframe*” cargará el componente principal de **Unity** (a partir de ahora se denominará como **controlador principal Unity**). Este componente pasa a ser el controlador de los minijuegos y los minijuegos se los trata como componentes o aplicaciones independientes acoplables al **HTML**. Por lo que el funcionamiento quedaría resumido en que, en el **HTML** se carga el **controlador principal Unity**, desde este se selecciona que minijuego se desea cargar y a través de **JavaScript** se encargará de cambiar el **controlador principal Unity** por el minijuego seleccionado.

ID-03	Problema con los minijuegos en la unión Web – Unity
Tiempo invertido	Medio.
Problema	Si se trata a Unity como un todo es imposible añadir minijuegos una vez esta exportado y subido a la Web sin para el servicio.
Solución	Tratar a los minijuegos como independientes acoplables a través de un controlador.

Tabla 104: Problema con los minijuegos en la comunicación Web – Unity

Esta solución, definida en la tabla 104, resuelve el problema pero genera una dificultad añadida inesperada. Dado que desde **JavaScript** no se puede saber que minijuego se ha pulsado en el **controlador principal Unity**.

Como primera aproximación a la solución, se planteó que, dado que es posible utilizar **PHP** para carga y descarga de datos en el **controlador principal Unity**, se podría pasar el controlador de local en el cliente (**JavaScript**) a remoto en el servidor (**PHP**). De esta manera cualquier cambio puede ser enviado al servidor que es donde reside la base de datos y los ejecutables de los juegos.

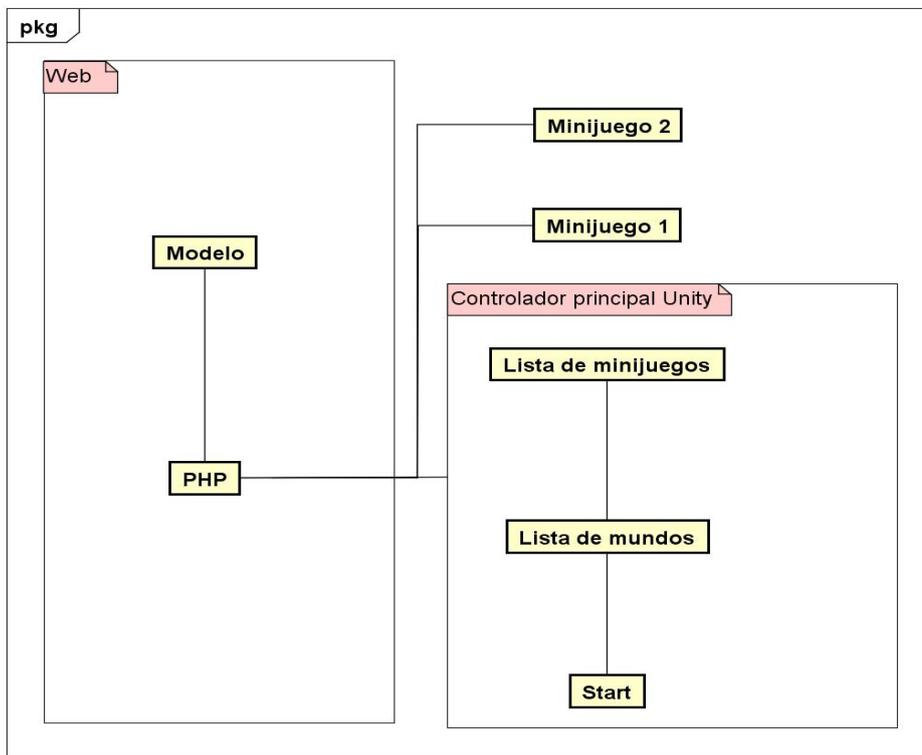


Figura 19: Comunicación Web – Unity, primera solución.

Esta solución, mostrada esquemáticamente en la figura 19, se implementa siguiendo esta secuencia de funciones:

- En el cliente, en el **controlador principal Unity** se pulsa el minijuego que se desea cargar.
- Se ejecuta un fichero **PHP** a través de un método *POST* desde el **controlador principal Unity** con argumento el identificador del minijuego que se quiere cargar.
- En el servidor el controlador **PHP**, consulta en la base de datos el minijuego por el identificador que ha facilitado el **controlador principal Unity**.

- En el servidor el controlador **PHP** facilita la información necesaria sobre el minijuego que tiene que mostrar en el **HTML**.
- En el cliente el controlador del **HTML**, el **JavaScript**, actualiza el contenido del **HTML** cambiando el **controlador principal Unity** por el minijuego seleccionado.

En el caso de pasar alguna información de **HTML** a **Unity** se tendría que seguir la secuencia de pasos inversa, apoyándose en la base de datos.

Una vez cerrado el minijuego se carga de nuevo el **controlador principal Unity**.

ID-04	Problema con el controlador y carga de un minijuego
Tiempo invertido	Bajo
Problema	Desde el controlador JavaScript no se puede saber que minijuego se ha pulsado en el controlador principal Unity.
Solución	Cambiar la función del controlador de local en el cliente (JavaScript) a remoto en el servidor (PHP).

Tabla 105: Problema con el controlador y carga de un minijuego

Con estas modificación la aplicación funciona correctamente resolviendo los requisitos principales.

El único requisito que esta versión no contempla es cargar los datos del usuario actualmente conectado al **controlador principal de Unity**, requisito imprescindible. Ya que si no se podía suministrar la información del usuario conectado actualmente entonces era imposible determinar que minijuegos había jugado, que puntuación había obtenido, que progreso había conseguido,... en definitiva el propósito de este trabajo. Lo que derivó en el replanteamiento total de la solución planteada.

Para implementar esta funcionalidad se propuso utilizar la *variable sesión* de **PHP** (una cookie generada cuando el inicia sesión en la aplicación la cual guarda datos del usuario). Las sesiones son globales, lo que significa que se pasan la información entre los ficheros **PHP** que la soliciten mediante la *variable sesión*.

Esta solución se tuvo que descartar ya que al invocar cualquier **PHP** desde el **controlador principal Unity** ejecutándose en el cliente, tal cual como se venia haciendo en la solución al problema anterior, la *variable sesión* se perdía. Se probó con el uso de bases de datos, archivos de texto plano, archivos *JSON*, funciones en **Unity** que estuvieran constantemente escuchando, entre otros tipos de intermediarios pero ninguna resolvía el problema satisfactoriamente.

Tras mucha investigación acompañada de muchas pruebas de ensayo y error, la solución a este gran problema vino de la mano de la investigación del componente al que se exporta **Unity** para web, **WebGL**, puesto que **WebGL** esta implementada en **JavaScript** entonces ahí tendría que estar la clave. Efectivamente, si se tiene el componente que utiliza **JavaScript** para mostrar **Unity** en **HTML** (que se puede obtener al exportar el proyecto) existe una función "*SendMessage*" que permite ejecutar un método de ese proyecto **Unity** pasándole como argumentos los parámetros necesarios. Una vez localizado este método solo hizo falta buscar la contraparte en **Unity** "*ExternalCall*" obteniendo así la solución al problema.

ID-05	Problema con la unión Unity – Web
Tiempo requerido	Muy alto.
Problema	El problema radicaba en que Unity una vez exportado tiene que poder comunicarse con el controlador del cliente (JavaScript).
Solución	Utilizar JavaScript como controlador entre controlador principal Unity y el controlador de la web (JavaScript) para enviar información. Utilizar en Unity la función ExternalCall para ejecutar una función en JavaScript con unos argumentos. Utilizar la función SendMessage en JavaScript para ejecutar una función en Unity.

Tabla 106: Problema con la comunicación Unity – Web

Una vez solucionado este problema solo hizo falta readaptar las problemáticas anteriores (ID-03 e ID-04) a la solución actual, cambiando el controlador de **PHP** a **JavaScript** y utilizando las funciones “SendMessage” y “ExternalCall” para la comunicación. La estructura final obtenida se muestra esquemáticamente en la figura 20:

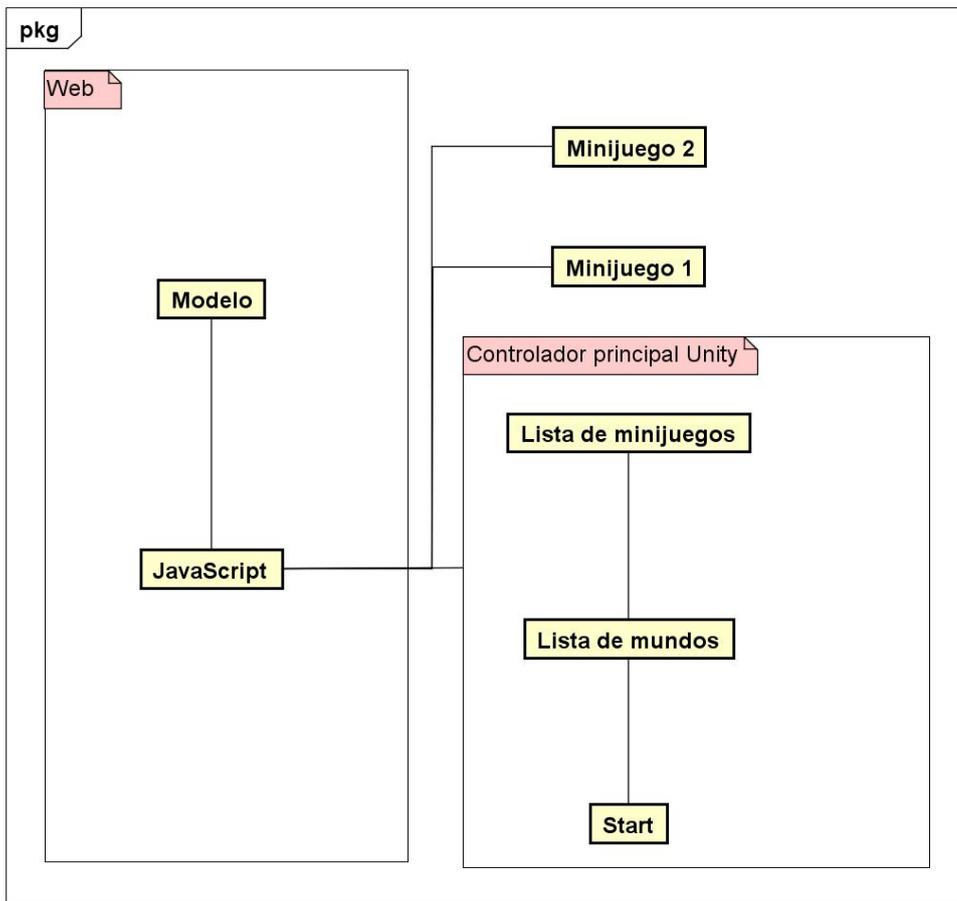


Figura 20: Comunicación Web – Unity, solución final.

La secuencia de funciones implementada quedaría:

- En el cliente, en la pagina de inicio **HTML** se carga el **controlador principal Unity**.
- En el cliente, el controlador JavaScript ejecuta la función “SendMessage” al controlador principal de Unity pasando la información del usuario conectado actualmente.

- Desde el **controlador principal Unity** pide al servidor la lista de mundos y la lista de minijuego disponibles para el usuario actual conectado.
- Se pulsa en el minijuego que se quiere cargar para jugar.
- El **controlador principal Unity** ejecuta la función “*ExternalCall*” al controlador **JavaScript** del **HTML** pasando la información del minijuego pulsado.
- El **JavaScript** solicita al servidor el minijuego especificado.
- El servidor, devuelve la dirección del minijuego solicitado.
- En el cliente, el controlador **JavaScript** actualiza el **HTML** cambiando el **controlador principal Unity** por el minijuego.
- Una vez terminado el minijuego, se ejecuta la función “*ExternalCall*” al controlador **JavaScript** del **HTML** pasando la información de los resultados.
- En el cliente, el controlador **JavaScript** recibe la información, se la enviá al controlador del servidor **PHP** y cambia e minijuego por el **controlador principal Unity**.
- Por ultimo en el servidor, el controlador **PHP** accede a la base de datos para actualizar los valores del minijuego jugado por el usuario actual.

6.5.3 Otro problema

Otro problema mas pequeño pero que también fue relevante es el que se detalla en la tabla 107:

ID-06	Problema con la carga de imágenes de forma dinámica
Tiempo requerido	Bajo
Problema	A la hora de cargar las imágenes si estas son muy pesadas Unity no las cargaba.
Solución	Forzar a que termine de cargar las imágenes para pasar a la siguiente escena.

Tabla 107: Problema con la carga de imágenes de forma dinámica

Capítulo 7

Plan de pruebas y evaluación

En este capítulo se describe la batería de pruebas realizada para comprobar el correcto funcionamiento de la aplicación, así como los resultados obtenidos. Se trata de una serie de pruebas de caja negra guiadas por los casos de uso. Se han realizado pruebas de todas las funcionalidades desarrolladas, en diferentes navegadores, pero dado el volumen de datos generado solo se describirán los casos de prueba más relevantes y que han servido para detectar fallos en la aplicación.

7.1 Casos de prueba

7.1.1 Usuarios

CP-01	Añadir un nuevo usuario
Descripción	Añadir un nuevo usuario al sistema con rol "Desarrollador"
Entrada	Desde el panel de administrador se selecciona "Añadir usuario" y en el formulario de "Añadir nuevo usuario" se introduce: <ul style="list-style-type: none"> • Nombre: juarodr • Contraseña: j546 • Rol: Desarrollador Se selecciona "Añadir".
Resultado esperado	El desarrollador se ha añadido correctamente y se puede ingresar con el sin problemas.

Tabla 108: Añadir un nuevo usuario

CP-02	Añadir lista de usuarios
Descripción	Añadir una lista de usuarios al sistema.
Entrada	<p>Desde el panel de administrador se selecciona “Añadir usuario” y en el formulario de “Añadir lista de usuarios nuevos” se introduce un fichero con estas características:</p> <ul style="list-style-type: none"> • Nombre: datos • Extensión: csv • Formato: Grupo;Rol;Nombre;Contraseña: <ol style="list-style-type: none"> 1. L1;U;fragome;f873 2. L2;U;marmule;m224 3. L2;U;fragome;f849 4. L3;D;andlope;a958 Se selecciona “Añadir”.
Resultado esperado	<p>Los usuarios 1 y 2 se han introducido correctamente. El usuario 1 se modifica con los datos del 3 por lo que se actualiza y el usuario 1 deja de existir. El desarrollador 4 se introduce correctamente y su grupo se cambia del L3 a D (grupo de desarrolladores) El desarrollador 4 se introduce correctamente y su grupo se cambia del L3 a D (grupo de desarrolladores)</p>

Tabla 109: Añadir lista de usuarios

CP-03	Modificar un usuario
Descripción	Modificar un usuario ya ingresado.
Entrada	<p>Desde el panel de administrador se selecciona “Revisar usuarios” se selecciona la opción “Modificar” del usuario y en el formulario se introduce:</p> <ul style="list-style-type: none"> • Nombre: juarodr • Nueva contraseña: (dejamos el campo en blanco) • Rol: Administrador <p>Se selecciona “Modificar”.</p>
Resultado esperado	El usuario pasará a ser administrador manteniendo la misma contraseña.

Tabla 110: Modificar un usuario

CP-04	Eliminar un usuario
Descripción	Eliminar un usuario ya ingresado.
Entrada	Desde el panel de administrador se selecciona “Revisar usuarios” se selecciona la opción “Eliminar” el usuario “juarodr”.
Resultado esperado	El usuario será eliminado de la base de datos

Tabla 111: Eliminar un usuario

7.1.2 Mundos y minijuegos

CP-05	Añadir un nuevo mundo
Descripción	Añadir un nuevo mundo al sistema.
Entrada	Desde el panel de administrador se selecciona “Añadir mundo” y en el formulario se introduce: <ul style="list-style-type: none"> • Nombre: Tema 1 • Imagen del mundo: Introducimos una imagen de un planeta. • Imagen de fondo: Introducimos una imagen de un fondo con nubes para el interior de ese mundo. Se selecciona “Añadir”.
Resultado esperado	El mundo se creará en el sistema.

Tabla 112: Añadir un nuevo mundo

CP-06	Modificar un mundo
Descripción	Modificar un mundo previamente creado.
Entrada	Desde el panel de administrador se selecciona “Mundos y minijuegos” y en el formulario se introduce: <ul style="list-style-type: none"> • Nombre: Tema 2 • Imagen del mundo: Introducimos una imagen de un planeta distinto. • Imagen de fondo: (No introducimos nada) Se selecciona “Modificar”.
Resultado esperado	Se modifica el nombre del mundo, la imagen del mundo y se eliminará la anterior imagen del mundo.

Tabla 113: Modificar un mundo

CP-07	Añadir un nuevo minijuego a un mundo
Descripción	Añadir un nuevo minijuego a un mundo ya creado previamente.
Entrada	Desde el panel de administrador se selecciona “Añadir minijuego” y en el formulario se introduce: <ul style="list-style-type: none"> • Seleccionar mundo: Tema 1 • Nombre: Minijuego 1 • Imagen: Introducimos una imagen de una isla. • Puntuación máxima: 2000. Se selecciona “Añadir”.
Resultado esperado	El minijuego se crea en el mundo designado.

Tabla 114: Añadir un nuevo minijuego a un mundo

CP-08	Añadir un proyecto a un minijuego
Descripción	Añadir un proyecto a un minijuego ya creado previamente.
Entrada	Desde el panel de administrador se selecciona “Mundos y minijuegos” y se selecciona la opción modificar del minijuego. Se obtiene el nombre al que hay que exportar el proyecto: <ul style="list-style-type: none"> • ID: 75 • Zip: 75.zip y los archivos internos deben empezar por 75. En caso contrario indicará que se ha exportado el proyecto mal y fallará. Se seleccionamos “Subir”.
Resultado esperado	Se añade el código al minijuego y ya es posible ejecutarlo en la aplicación.

Tabla 115: Añadir un proyecto a un minijuego

CP-09	Modificar un proyecto de un minijuego
Descripción	Modificar un proyecto de un minijuego por otro con el mismo ID.
Entrada	Desde el panel de administrador se selecciona “Mundos y minijuegos” y se selecciona la opción modificar del minijuego. Se obtiene el nombre al que hay que exportar el proyecto: <ul style="list-style-type: none"> • ID: 75 • Zip: 75.zip y los archivos internos deben empezar por 75. En caso contrario indicará que se ha exportado el proyecto mal y fallará. Se selecciona “Subir”.
Resultado esperado	Se modifica el código del minijuego.

Tabla 116: Modificar un proyecto de un minijuego

CP-10	Modificar un minijuego
Descripción	Modificar un minijuego sin que afecte al proyecto asociado.
Entrada	Desde el panel de administrador se selecciona “Mundos y minijuegos”, se selecciona la opción modificar del minijuego e introducimos: <ul style="list-style-type: none"> • Nombre: Minijuego 2 • Imagen: (lo dejamos como esta) • Puntuación máxima: 5000. Se selecciona “Subir”.
Resultado esperado	Se modifica el minijuego sin cambiar la imagen.

Tabla 117: Modificar un minijuego

CP-11	Eliminar un minijuego
Descripción	Eliminar un minijuego junto a su código.
Entrada	Desde el panel de administrador se selecciona “Mundos y minijuegos”, se selecciona la opción “Eliminar” del minijuego.
Resultado esperado	Se ha eliminado el minijuego, tanto su código como las imágenes asociadas a él.

Tabla 118: Eliminar un minijuego

CP-12	Eliminar un mundo
Descripción	Eliminar un mundo sin minijuegos asociados.
Entrada	Desde el panel de administrador se selecciona “Mundos y minijuegos”, se selecciona la opción “Eliminar” del mundo.
Resultado esperado	Se ha eliminado el mundo y las imágenes asociadas a él.

Tabla 119: Eliminar un mundo

7.1.3 Jugar

CP-13	Iniciar un minijuego
Descripción	Se inicializará un minijuego.
Entrada	Desde el inicio se selecciona el mundo con nombre "Tema 1" y dentro del mundo se selecciona el minijuego con nombre Minijuego 1.
Resultado esperado	Se cargará el minijuego 1 y en la base de datos se generará un registro en la tabla "usariominijuego" con IDUsuario, IDMinijuego y FechaInicio.

Tabla 120: Iniciar un minijuego

CP-14	Terminar un minijuego
Descripción	Termina un minijuego.
Entrada	Dentro del minijuego se hacen los requerimientos pertinentes para conseguir completar el minijuego.
Resultado esperado	Se vuelve a la lista de los minijuegos, se actualizarán los puntos y las estrellas que se han obtenido. Y en la base de datos se actualizará el registro creado anteriormente con Puntuación, Tiempo, Completado (si, no) y FechaFin

Tabla 121: Terminar un minijuego

CP-15	Salir de un minijuego
Descripción	Salir sin completar de un minijuego.
Entrada	Dentro del minijuego se selecciona la opción atrás.
Resultado esperado	Se vuelve a la lista de los minijuegos.

Tabla 122: Salir de un minijuego

7.2 Resultado de las pruebas

En este punto, se muestran los resultados obtenidos al ejecutar esta serie de pruebas en diferentes navegadores. Por cada uno, se muestra una tabla con los resultados, los fallos que se han producido y su solución. Cabe destacar que algunos fallos producidos han servido para corregir fallos similares en posteriores casos de prueba, por lo que los fallos encontrados por empezar las pruebas en un determinado navegador no indica que ese navegador sea más inestable a la hora de ejecutar la aplicación.

7.2.1 Google Chrome

En la tabla 53, se detallan los resultados de las pruebas realizadas en el navegador **Google Chrome** desde un ordenador.

Prueba	Resultado	Solución
CP-01	Correcto.	-
CP-02	Fallo: Se agregan correctamente pero no funcionaba el inicio de sesión.	- El problema estaba en la última columna de la lista hay que agregarle el separador también a esta última columna. EJ: L1;U;fragome;f873;
CP-03	Correcto.	-
CP-04	Correcto.	-
CP-05	Fallo: El mundo no se guarda en la BD.	- Al insertar hay que especificar todas las columnas, incluidas aquellas que su valor quieres que sea por defecto.
CP-06	Correcto.	-
CP-07	Correcto.	-
CP-08	Fallo: No detecta ningún proyecto.	- Este error ha sido bastante difícil pues no había ningún indicio de que podía estar pasando. No obstante el error no provenía del código si no de la configuración del servidor local. Solo admitía una subida de archivos con un tamaño de 8MB como máximo, por lo que la solución fue cambiar la limitación de tamaño de las subidas.
CP-09	Correcto.	-
CP-10	Correcto.	-
CP-11	Correcto.	-
CP-12	Correcto.	-
CP-13	Correcto.	-
CP-14	Correcto.	-
CP-15	Correcto.	-

Tabla 123: Resultado de las pruebas en Google Chrome

7.2.2 Firefox

En la tabla 54, se detallan los resultados de las pruebas realizadas en el navegador **Firefox** desde un ordenador

Prueba	Resultado	Solución
CP-01	Correcto.	-
CP-02	Correcto.	-
CP-03	Correcto.	-
CP-04	Correcto.	-
CP-05	Correcto.	-
CP-06	Correcto.	-
CP-07	Correcto.	-
CP-08	Correcto.	-
CP-09	Correcto.	-
CP-10	Correcto.	-
CP-11	Correcto.	-
CP-12	Correcto.	-
CP-13	Correcto.	-
CP-14	Correcto.	-
CP-15	Correcto.	-

Tabla 124: Resultado de las pruebas en Firefox

7.2.3 Microsoft Edge

En la tabla 55, se detallan los resultados de las pruebas realizadas en el navegador **Microsoft Edge** desde un ordenador.

Prueba	Resultado	Solución
CP-01	Correcto.	-
CP-02	Correcto.	-
CP-03	Correcto.	-
CP-04	Correcto.	-
CP-05	Correcto.	-
CP-06	Correcto.	-
CP-07	Correcto.	-
CP-08	Correcto.	-
CP-09	Correcto.	-
CP-10	Correcto.	-
CP-11	Correcto.	-
CP-12	Correcto.	-
CP-13	Correcto.	-
CP-14	Correcto.	-
CP-15	Correcto.	-

Tabla 125: Resultado de las pruebas en Microsoft Edge

Capítulo 8

Conclusiones y trabajo futuro

8.1 Conclusiones

Lo primero y mas importante a indicar es que se han logrado todos los objetivos inicialmente propuestos. Se ha construido una plataforma que permite integrar distintos minijuegos, desarrollados de manera independiente. Esta plataforma guarda información de las acciones de los usuarios así como su evolución en los distintos juegos, por lo que podrá ser usada tanto para gamificar la asignatura planteada, como para investigar acerca de su uso y su repercusión en el aprendizaje del alumno. Desde el comienzo se tuvo en mente crear algo abierto, y se ha logrado, por lo que el proyecto desarrollado está preparado para ponerse a disposición de la comunidad académica para ser usado más allá de la asignatura planteada.

Este trabajo tiene un carácter integrador de numerosas asignaturas de la carrera. En concreto: asignaturas de planificación y gestión de proyectos, análisis y diseño de software, programación orientada a objetos, sistemas web, análisis y diseño de bases de datos y la asignatura de programación de aplicaciones gráficas. Una primera aproximación al mundo de los motores gráficos y en definitiva al de los videojuegos, un campo de gran interés para mi, pero en el cual nunca había trabajado hasta la fecha.

La realización de este proyecto ha sido debido a la gran rigidez de **Unity** al salirse de su ecosistema. Esto ha supuesto que una gran parte del tiempo se haya dedicado a buscar una manera de comunicar el **HTML** y el **controlador principal Unity** cumpliendo los objetivos planteados para ella. Teniendo que, durante el desarrollo y pruebas de funcionamiento, duplicar el **controlador principal Unity** como dos proyectos diferentes, uno preparado para que funcionara en el entorno de programación **Unity**, y otro con las funcionalidades preparadas para conectarse con el **HTML**, con algunas configuraciones distintas y algunos métodos pensados única y exclusivamente para conectarse al **HTML**.

Aunque ya había hecho pequeños proyectos en **WebGL** y web, este proyecto me ha permitido aprender dos nuevos lenguajes de programación, cosa que es siempre positiva. Así como relacionar dos lenguajes que aparentemente no tienen relación como **C#** y **JavaScript** que ha sido muy complicado de solucionar.

El trabajo realizado me ha permitido profundizar en el conocimiento de **Unity** que es uno de los motores gráficos más populares hoy en día, ser capaz de adaptar aplicaciones inconclusas por requisitos del usuario y fundamentalmente aprender a gestionar problemas aparentemente irresolubles. Considero esta etapa de mi vida como de gran aprendizaje y crecimiento personal.

8.2 Trabajo futuro

En este punto, se incluyen ciertas funcionalidades que sería interesante añadirlas en futuras versiones:

- Mejorar el apartado visual de la aplicación, añadiendo animaciones, transiciones y trasfondo, algo más propio de los videojuegos.
- Implementar una API que permita a los desarrolladores probar sus minijuegos sin necesidad de subirlo a la aplicación previamente. Y subirlo a la aplicación como última acción, cuando el minijuego esté listo para ser jugado.
- Implementar una funcionalidad para los administradores que permita ver y analizar los datos de la base de datos mostrando gráficos que permitan usarse con fines de investigación.
- Añadir una nueva funcionalidad de “pasar de curso” para permitir guardar los datos de cursos anteriores sin necesidad de eliminarlos.
- Permitir minijuegos de distintas entornos, no únicamente de **Unity**. En un principio se planteó para albergar minijuegos creados exclusivamente por un entorno **Unity**, pero se ha implementado de tal forma que con unas modificaciones de configuración en la base de datos y en unos cuantos ficheros, se permitirían minijuegos de otro lenguaje siempre y cuando tengan compatibilidad web.

Bibliografía

[1] Johan Huizinga, Homo ludens.

Disponible en: <https://www.ekilikua.com/blog/homo-ludens,-dos-visiones-fascinantes-del-juego>

[Ultimo acceso: 23-septiembre-2020]

[2] Fundación Cotec, Mi empleo mi futuro.

Disponible en: <https://cotec.es/proyecto/miempleomifuturo/>

[Ultimo acceso: 23-septiembre-2020]

[3] Virginia Gaitán. Gamificación: el aprendizaje divertido.

Disponible en: <https://ebot.es/beneficios-gamificacion-aula/>

[Ultimo acceso: 23-septiembre-2020]

[4] Alma María Pisabarro Marrón y Carlos Enrique Vivaracho Pascual, Gamificación en el aula: gincana de programación. Disponible en:

<http://www.aenui.net/ojs/index.php?journal=revisión&page=article&op=view&path%5B%5D=402&path%5B%5D=593>

[Ultimo acceso: 23-septiembre-2020]

[5] David Erosa García, ¿Que es Unity?.

Disponible en: <https://openwebinars.net/blog/que-es-unity/>

[Ultimo acceso: 23-septiembre-2020]

[6] Unity Technologies, Manual Unity.

Disponible en: <https://docs.unity3d.com/Manual/index.html>.

[Ultimo acceso: 23-septiembre-2020]

[7] Khronos Group, WebGL.

Disponible en: <https://www.khronos.org/webgl/>

[Ultimo acceso: 23-septiembre-2020]

[8] Ionos by 101, Modelo en cascada. Disponible en:

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>

[Ultimo acceso: 23-septiembre-2020]

[9] Universidad de Alicante, MVC. Disponible en:

<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

[Ultimo acceso: 23-septiembre-2020]

[10] Marketi Web, Ventajas de usar MVC. Disponible en:
<https://marketiweb.com/empresa/blog/item/114-que-es-la-arquitectura-mvc-y-cuales-son-sus-ventajas>

[Ultimo acceso: 23-septiembre-2020]

[11] Oracle Corporation, MySQL.

Disponible en: <https://www.mysql.com/>

[Ultimo acceso: 23-septiembre-2020]

Anexos

Manual de usuario

Con este manual le permitirá aprender a utilizar todas las funcionalidades de la aplicación.

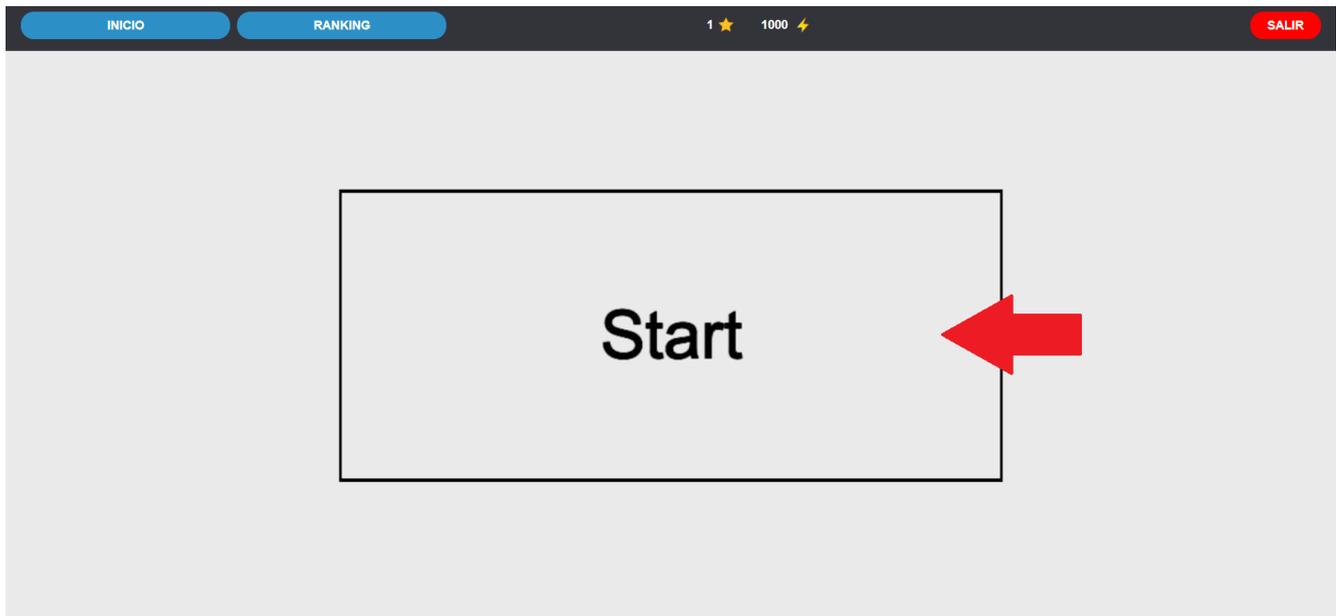
1.- ¿Como acceder a la aplicación?

Acceda a la página principal, introduzca su usuario y contraseña y presione el botón entrar. En el caso de no disponer de credenciales póngase en contacto con el administrador.

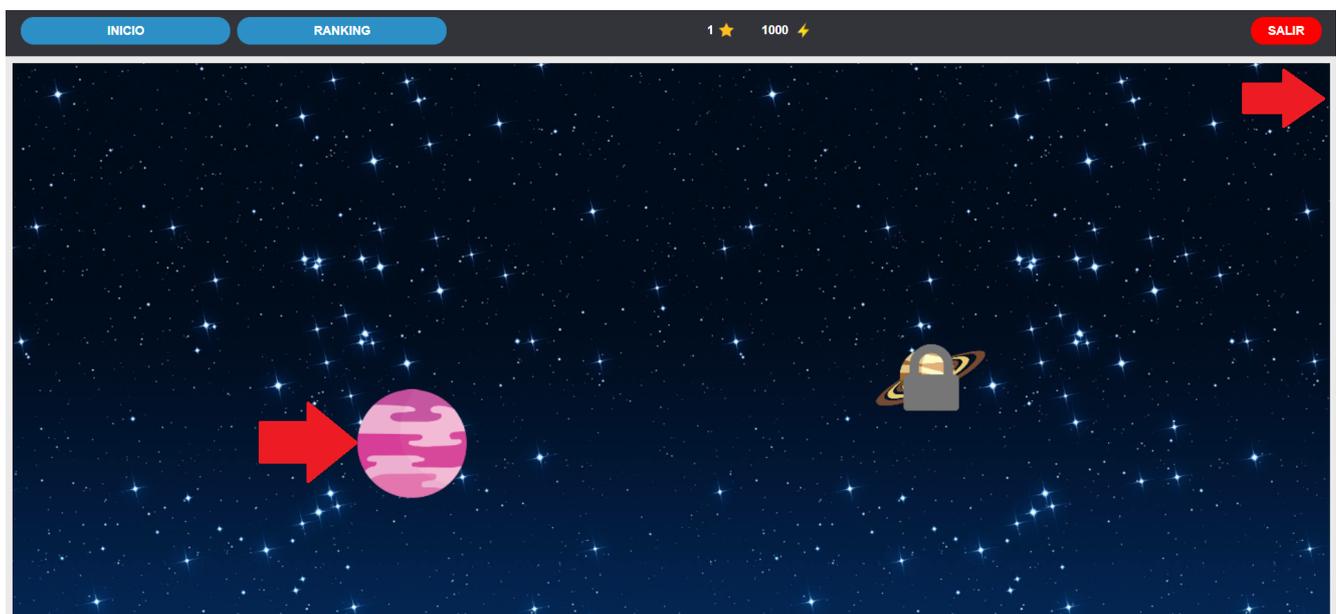


2.- ¿Como jugar a un minijuego?

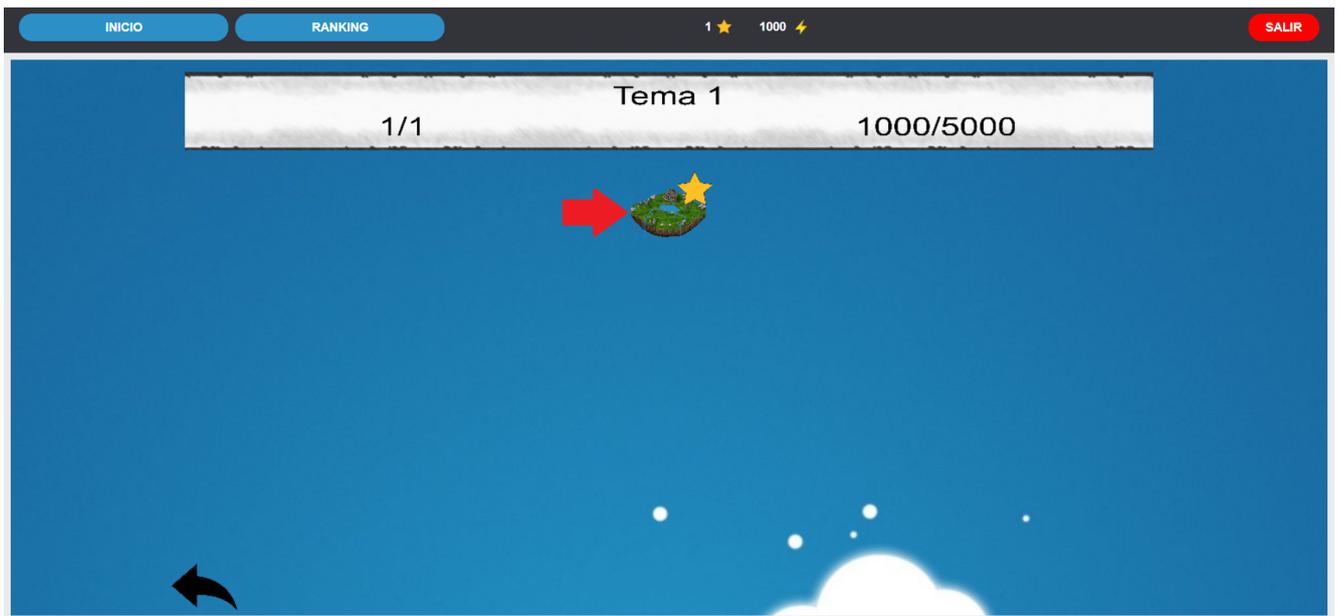
Una vez iniciado sesión se pulsa el botón “Start” que aparece en mitad de la pantalla, nos aparecerá la lista de los mundos disponibles para el usuario conectado.



Se selecciona un mundo de entre los disponibles. Podemos pasar por encima de los mundos con el ratón para conocer información de estos y también podemos desplazarnos hacia el lateral manteniendo pulsado el ratón y moviéndolo horizontalmente.



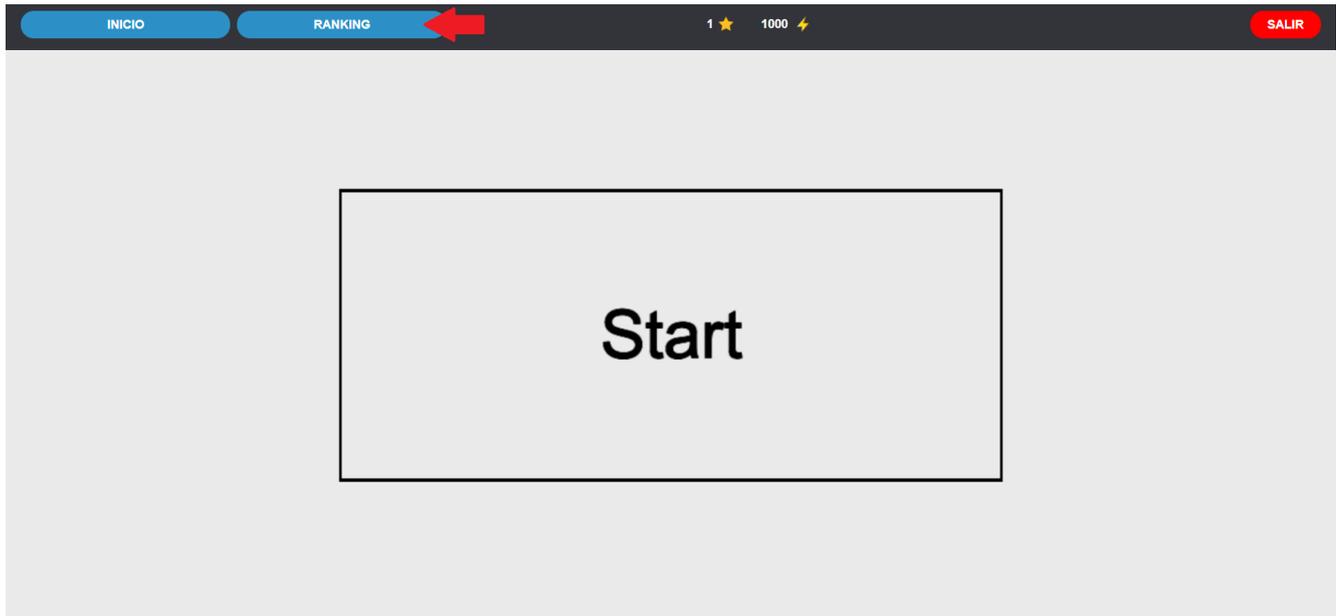
Por ultimo se selecciona el minijuego que se quiera iniciar.



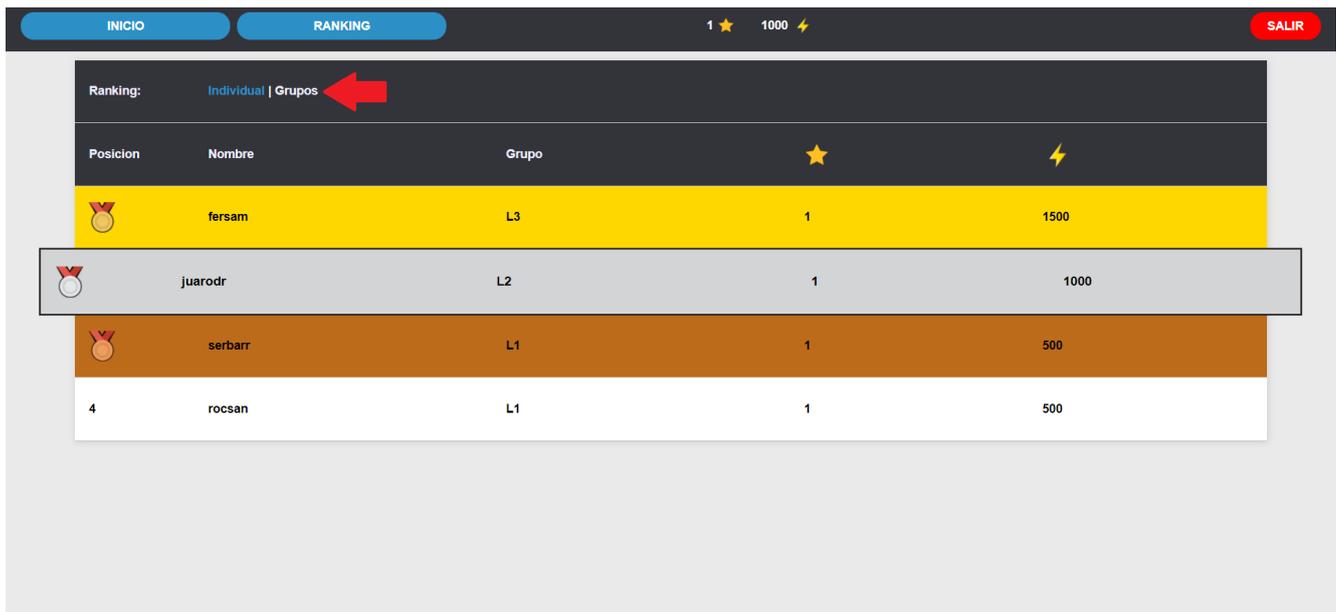
Acto seguido se mostrará el minijuego en la pantalla.

3.- ¿Como ver el ranking de los jugadores?

Una vez iniciado sesión se pulsa el botón "Ranking" situado en la parte superior de la página.



Aquí se mostrarán las distintas posiciones de los usuarios ordenados por la cantidad de estrellas y los puntos que se hayan conseguido.



Por ultimo tienes la opción de ver el ranking de los grupos de usuarios. Que mostrará las distintas posiciones de los grupos de los usuarios ordenados por la cantidad de estrellas y los puntos que se hayan conseguido el grupo completo.

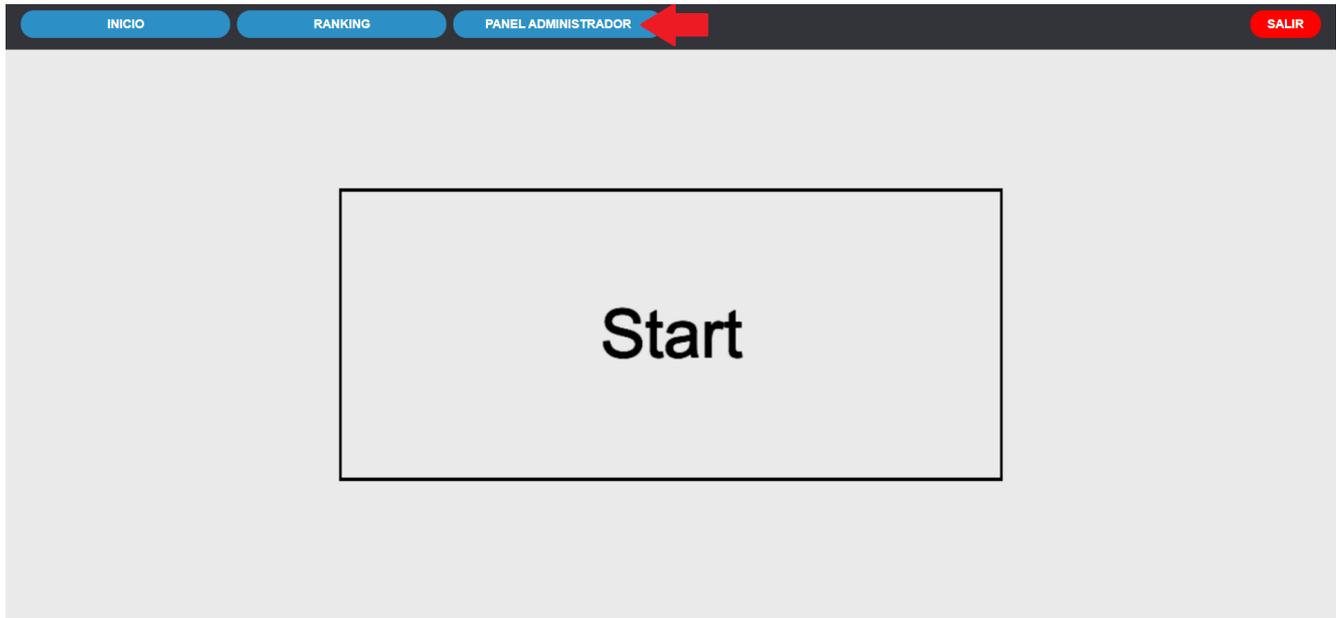
INICIO RANKING 1 ★ 1000 ⚡ SALIR

Ranking: Individual | Grupos

Posición	Grupo	★	⚡
	L1	2	1000
	L3	1	1500
	L2	1	1000

4.- ¿Como añadir un mundo nuevo?

Una vez iniciado sesión como administrador, se pulsa el botón “Panel Administrador” situado en la parte superior de la página.



Aquí se mostrarán los distintos mundos y minijuegos creados actualmente y ordenados por posición. En el panel lateral se pulsa “Añadir mundo” y nos llevara al formulario que nos permitirá añadir un mundo nuevo.



Por ultimo en el formulario se rellenan los campos con: Nombre del mundo, Imagen de la miniatura del mundo (sera el que aparezca en la selección de mundos del controlador principal Unity)

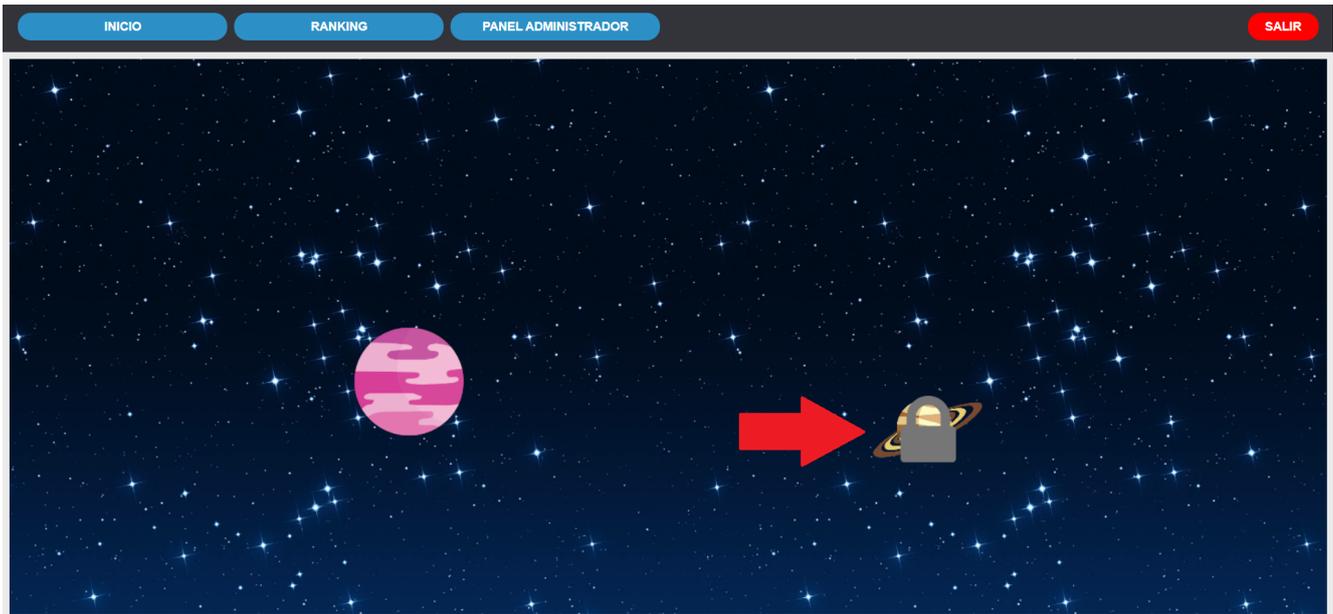
y la imagen de fondo (Imagen que aparecerá como fondo una vez seleccionemos el mundo y se muestren los minijuegos).



Se puede comprobar que se ha añadido correctamente si aparece en el inicio del panel de administrador.

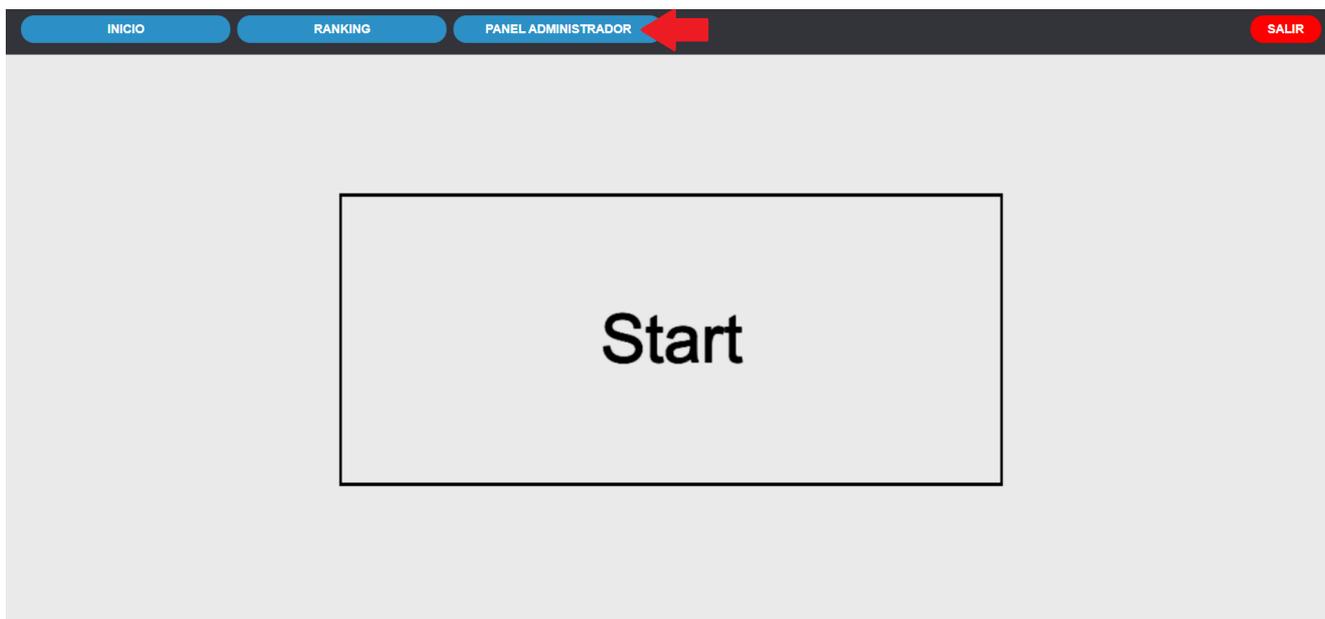


También se puede ver como aparece el mundo nuevo en la lista de mundos a seleccionar.



5.- ¿Como modificar un mundo y otras funcionalidades?

Una vez iniciado sesión como administrador, se pulsa el botón “Panel Administrador” situado en la parte superior de la página.



Aquí se mostrarán los distintos mundos y minijuegos creados actualmente y ordenados por posición.



Hay distintas opciones que se pueden aplicar a un mundo ya creado:

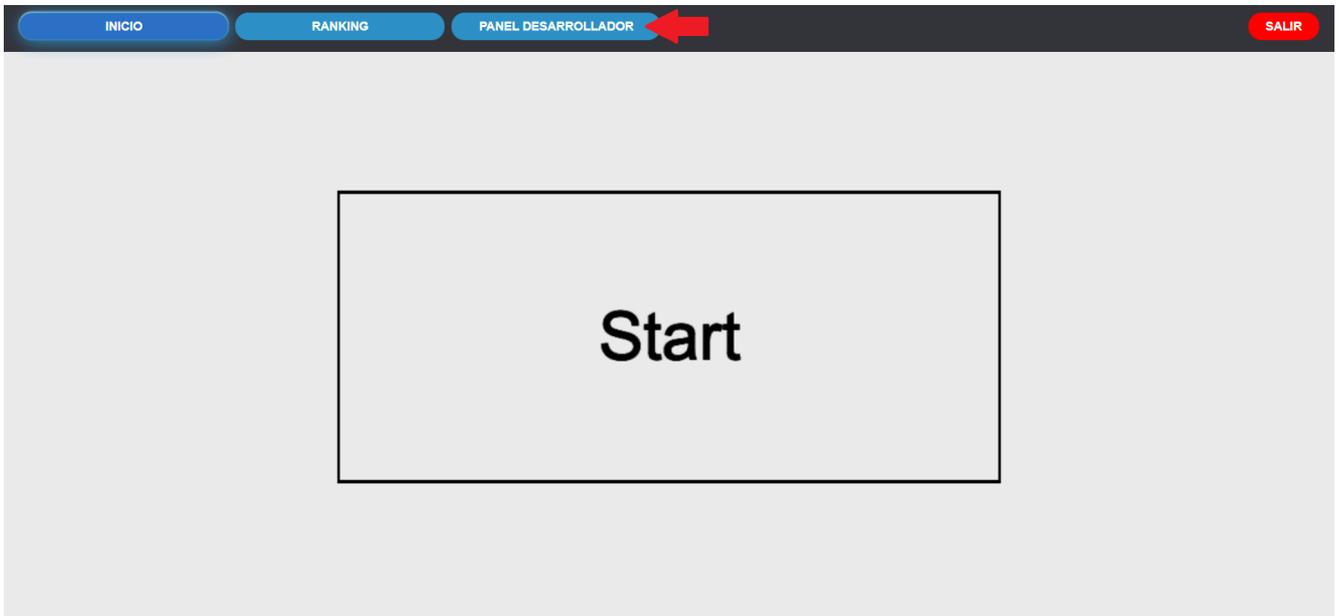
1. Se puede cambiar la disponibilidad del mundo (Cerrado o Abierto). Por defecto aparece cerrado a los usuario por lo que solo es accesible para los administradores.

2. Se puede cambiar la visibilidad del mundo (Visible o Oculto). Por defecto aparece oculto a los usuario por lo que solo es visible para los administradores.
3. Se puede modificar el mundo. Cambiando el nombre, la imagen del mundo y la imagen de fondo.
4. Se puede modificar la posición del mundo. Por defecto los nuevos mundos se añaden al final.
5. Se puede eliminar un mundo. Se eliminara todo lo relacionado con el mundo excepto si tiene minijuegos asociados en ese caso hay que eliminar primero los minijuegos.

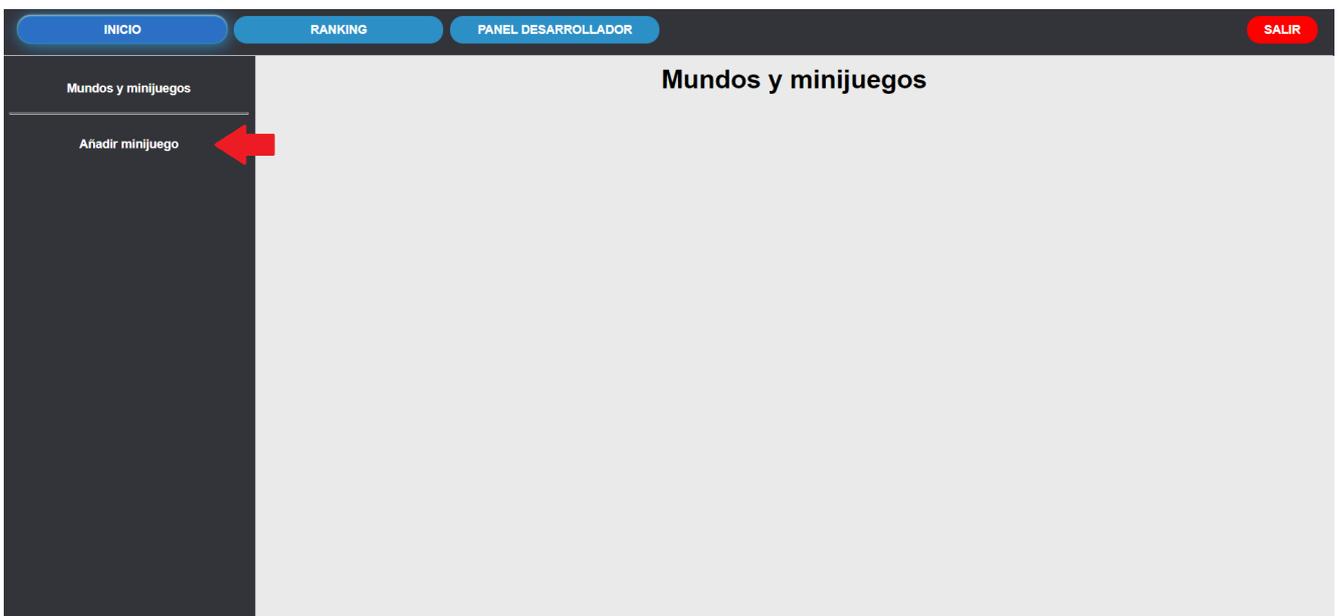


6.- ¿Como añadir minijuego?

Una vez iniciado sesión como administrador o como desarrollador, se pulsa el botón “Panel Administrador” o “Panel Desarrollador” situado en la parte superior de la página.



Aquí se mostrarán los distintos minijuegos asociados a su mundo, creados actualmente por el desarrollador y ordenados por posición. En el panel lateral se pulsa “Añadir minijuego” y nos llevara al formulario que nos permitirá añadir un nuevo minijuego.



En el formulario se rellenan los campos con: mundo al que se va a asociar al minijuego, Nombre del minijuego, Imagen de la miniatura del minijuego (sera el que aparezca en la selección de minijuegos del controlador principal Unity) y la puntuación máxima que se puede obtener en el minijuego.

The screenshot shows the 'Añadir minijuego' form in the 'PANEL DESARROLLADOR' section. The form includes a dropdown menu for 'Tema 1', a text input for 'Nombre' (Minijuego 1), a file selection button for 'Imagen' (isla.png), and a text input for 'Puntuación maxima' (5000). A red arrow points to the 'Tema 1' dropdown, and another red arrow points to the 'Añadir' button at the bottom. There are also red arrows pointing to the 'Nombre' and 'Puntuación maxima' input fields.

Una vez creado el minijuego se puede ver en la lista de mundos creados. Pero aun ha que subir el código fuente del minijuego para poder ejecutarlo. Para ello primero se pulsa el botón "Modificar del minijuego".

The screenshot shows the 'Mundos y minijuegos' list in the 'PANEL DESARROLLADOR' section. The list has a header 'Mundos y minijuegos' and a sub-header 'Desplegar minijuegos'. The list contains two entries: 'Tema 2' and 'Minijuego 1'. The 'Minijuego 1' entry has a score of 5000 and a red arrow pointing to the 'Modificar del minijuego' button (represented by a pencil icon).

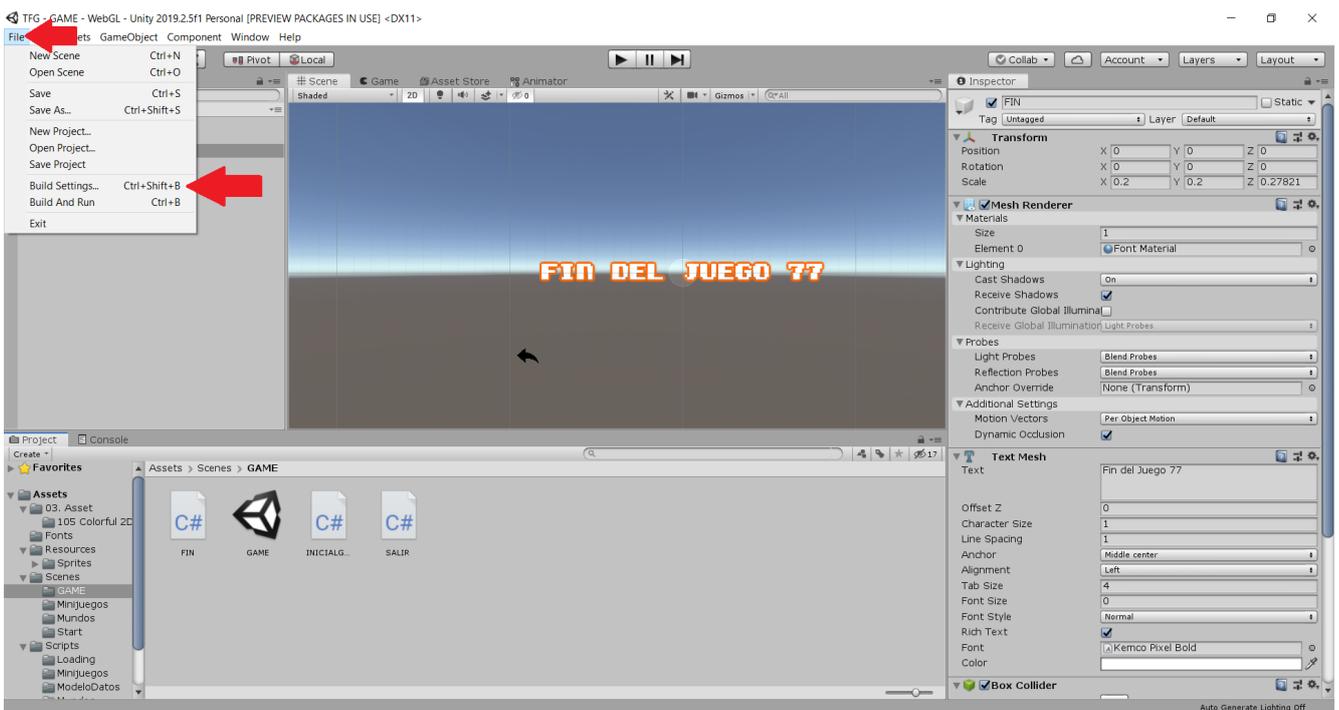
Mundo	Minijuego	Puntuación	Acciones
Tema 2			 
	Minijuego 1	5000	 

Una vez se ha creado el minijuego hay que facilitar el código fuente del proyecto del mismo, bajo el nombre de un número que nos proporciona la aplicación (En el caso del ejemplo el número proporcionado por la aplicación es el 77).



Para la obtención del código fuente de la aplicación hay que seguir los siguientes pasos:

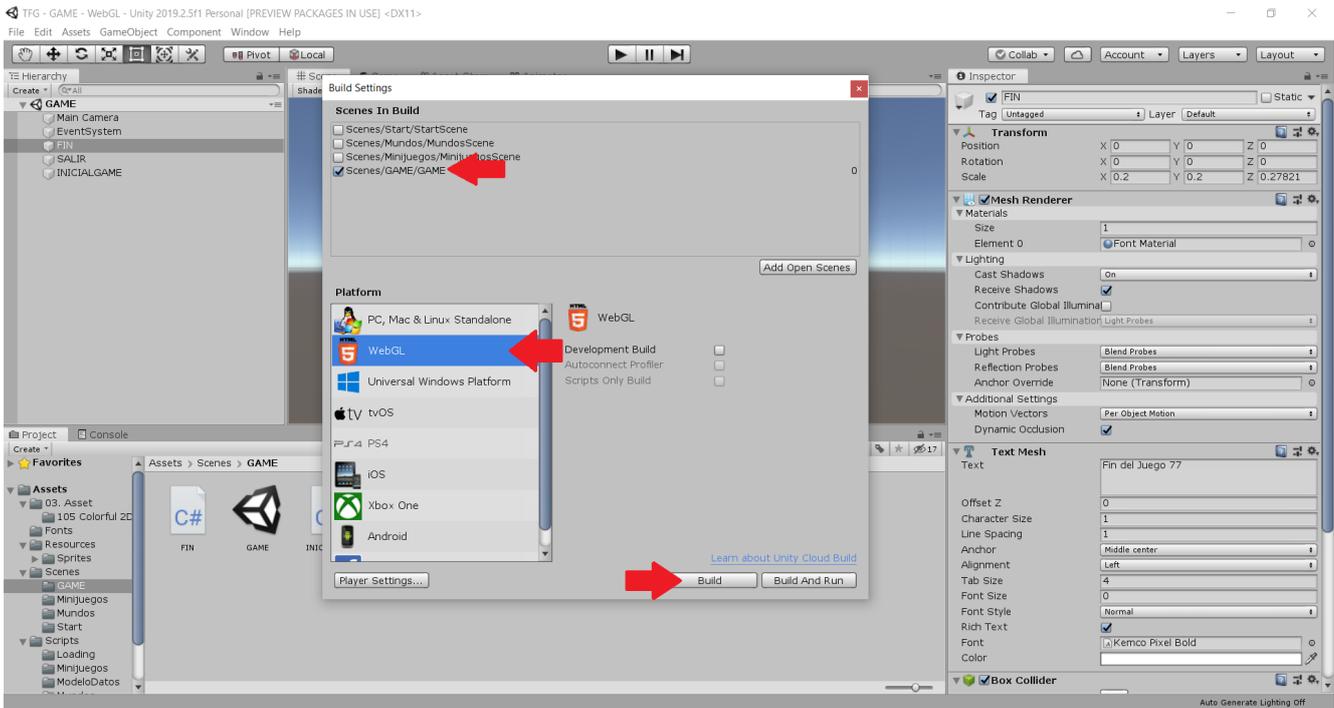
1.- Se abre el entorno Unity, en el menú principal se presiona "File" y "Build Settings"



2.- Se selecciona el conjunto de escenas que se quieren exportar (en el caso del ejemplo solo se quiere exportar una).

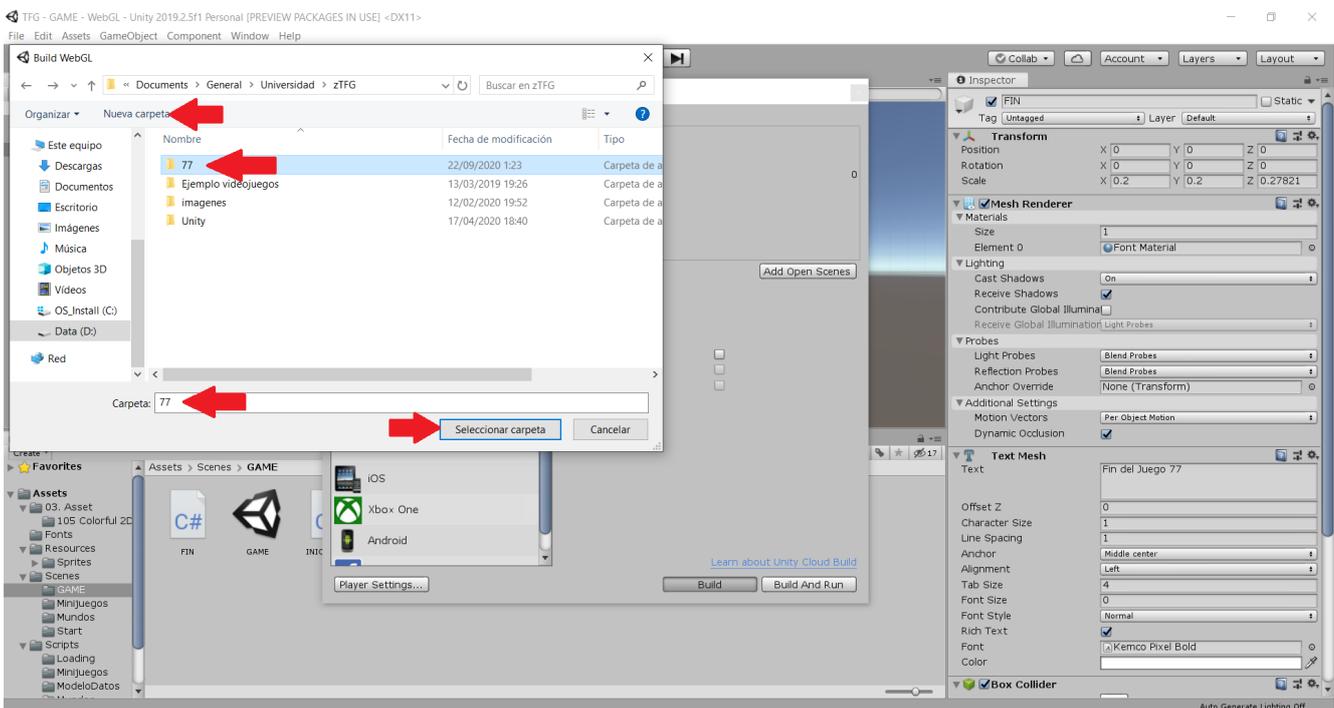
3.- Hay que asegurarse de que el complemento de WebGL está instalado y seleccionado.

4.- Se presiona el botón "build".

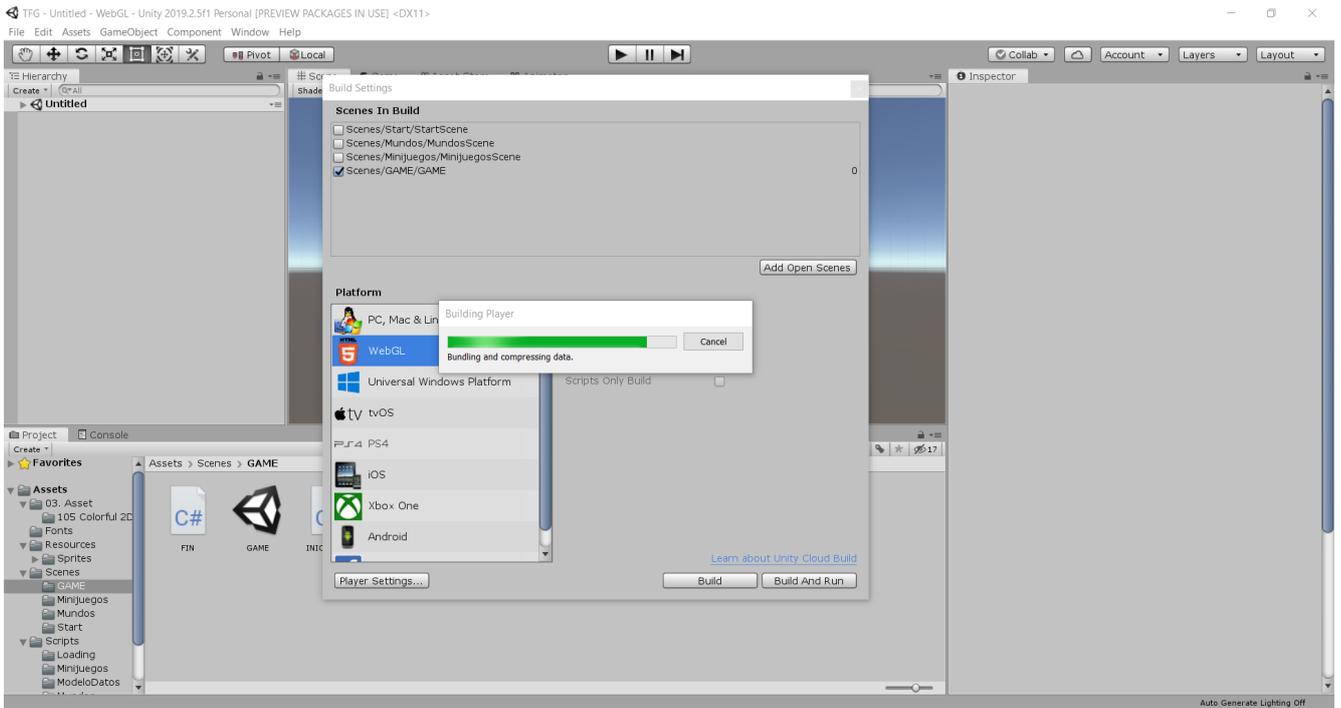


5.- Se selecciona el directorio de destino que queremos y se crea una carpeta nueva pero, importante el nombre de esa carpeta tiene que ser el número facilitado por la aplicación anteriormente (en este caso 77).

6.- Se selecciona esa carpeta.



7.- Esperamos a que Unity termine de exportar.



8.- Una vez haya terminado de exportar, accedemos al directorio creado, y se accede al directorio "Build".

Nombre	Fecha de modificación	Tipo	Tamaño
Build	22/09/2020 1:33	Carpeta de archivos	
TemplateData	22/09/2020 1:33	Carpeta de archivos	
index.html	22/09/2020 1:31	Chrome HTML Do...	1 KB

9.- Hay que asegurarse de que los archivos generados empiezan por el número facilitado por la aplicación (en este caso 77). En caso contrario es importante no modificarlos manualmente y volver al paso 1.

Nombre	Fecha de modificación	Tipo	Tamaño
77.data.unityweb	22/09/2020 1:32	Archivo UNITYWEB	6.269 KB
77.json	22/09/2020 1:31	Archivo JSON	1 KB
77.wasm.code.unityweb	22/09/2020 1:33	Archivo UNITYWEB	3.551 KB
77.wasm.framework.unityweb	22/09/2020 1:33	Archivo UNITYWEB	75 KB
UnityLoader.js	04/09/2019 15:21	Archivo JavaScript	156 KB

10.- Se comprime el directorio con el número facilitado a un comprimido con el mismo nombre y (importante) con extensión .zip.

Nombre	Fecha de modificación	Tipo	Tamaño
 77	22/09/2020 1:33	Carpeta de archivos	
 77.zip	22/09/2020 1:41	Archivo WinRAR ZIP	1 KB

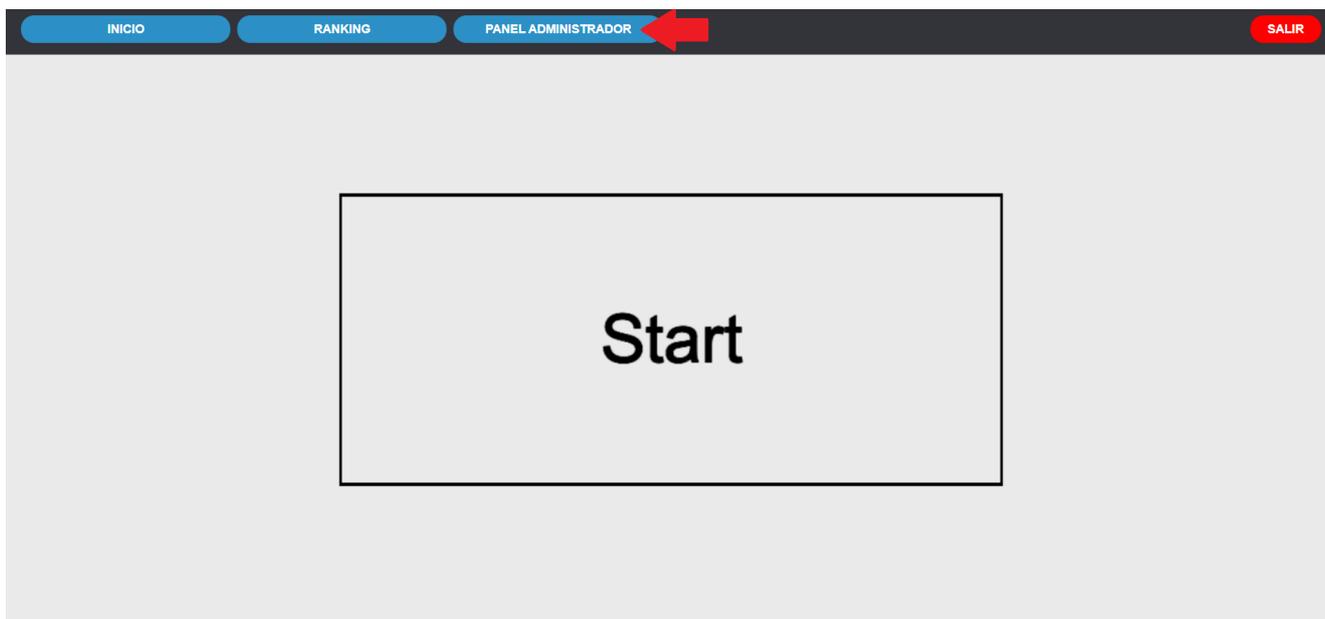
Una vez se ha obtenido el comprimido del código fuente del minijuego volvemos a la página web, se introduce el comprimido y se pulsa el botón “subir.”



Por ultimo se puede revisar que se ha subido correctamente ejecutando el minijuego en la lista de minijuegos del controlador principal Unity.

7.- ¿Como modificar minijuego y otras funcionalidades?

Una vez iniciado sesión como administrador, se pulsa el botón “Panel Administrador” situado en la parte superior de la página.



Aquí se mostrarán los distintos mundos y minijuegos creados actualmente y ordenados por posición.



Hay distintas opciones que se pueden aplicar a un mundo ya creado:

1. Se puede cambiar la disponibilidad del minijuego (Cerrado o Abierto). Por defecto aparece cerrado a los usuario por lo que solo es accesible para los administradores y el desarrollador responsable del desarrollo de ese minijuego.

2. Se puede cambiar la visibilidad del minijuego (Visible o Oculto). Por defecto aparece oculto a los usuario por lo que solo es visible para los administradores y el desarrollador responsable del desarrollo de ese minijuego.

3. Se puede modificar el minijuego. Cambiando el mundo al que pertenece, el nombre, la imagen del mundo y la puntuación máxima que se puede obtener.

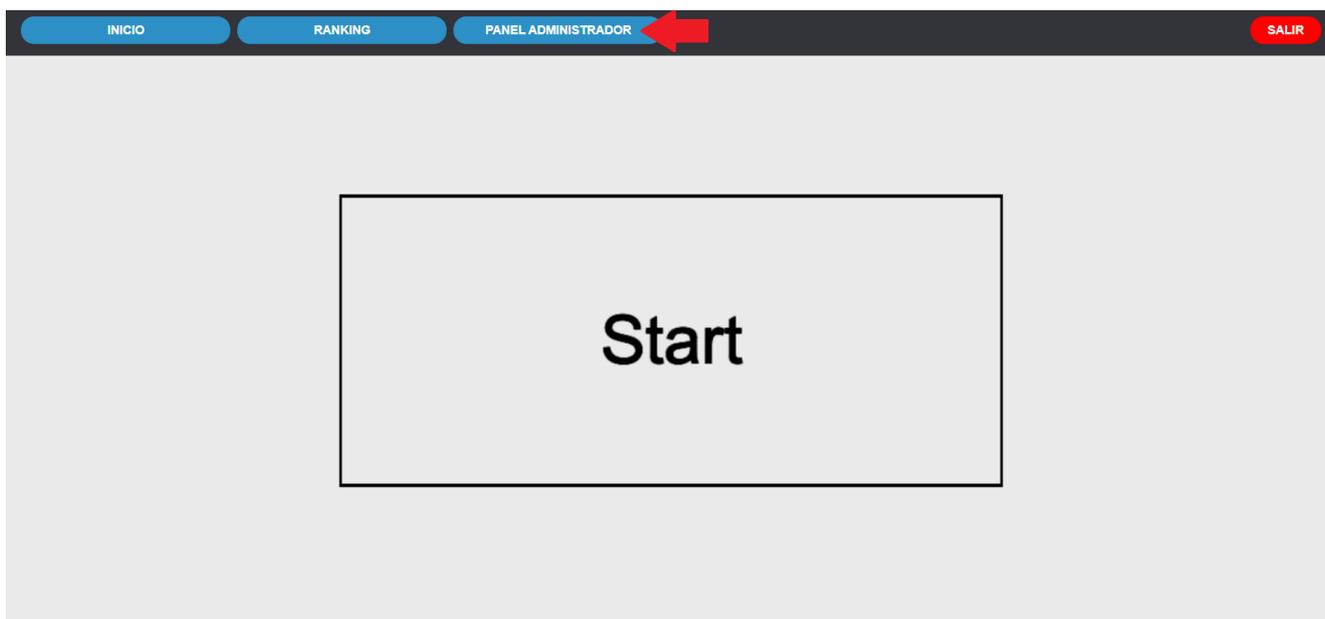
4.- Se puede modificar el código fuente del proyecto sobrescribiendo el anterior.

4. Se puede modificar la posición del minijuego. Por defecto los nuevos mundos se añaden al final.

5. Se puede eliminar un minijuego. Se eliminara todo lo relacionado con el minijuego excepto si tiene usuarios que hayan jugado al menos una vez, en ese caso hay que borrar a esos usuarios primero.

8.- ¿Como añadir un usuario o una lista de usuarios?

Una vez iniciado sesión como administrador, se pulsa el botón “Panel Administrador” situado en la parte superior de la página.



En el panel lateral se pulsa “Añadir usuario”.



Aquí se disponen de dos opciones, o bien agregar un usuario manualmente o bien introducir una lista de usuarios.

En el caso de querer agregar un usuario manualmente hay que introducir los siguientes elementos:

- Nombre de usuario.
- Contraseña.
- Rol (Usuario, Administrador o Desarrollador).
- Grupo de usuarios al que pertenecerá.

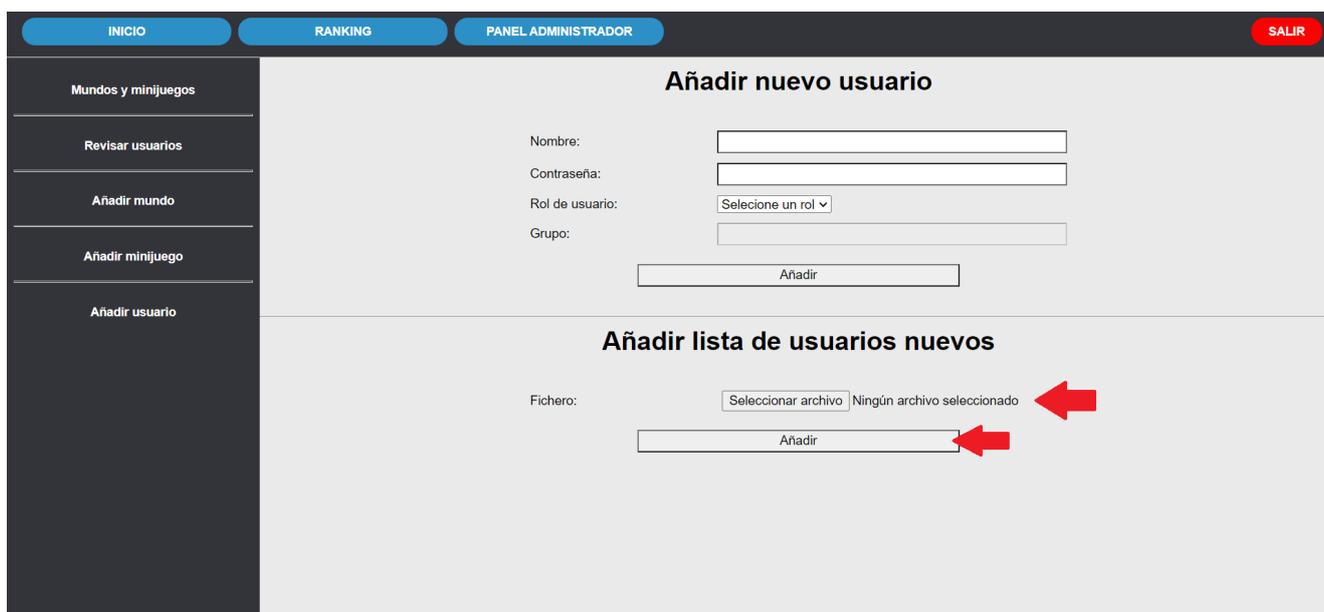
En el caso de tener un rol de Administrador o Desarrollador el grupo se deshabilitara y no se podrá introducir.

En el caso de que exista un usuario con el mismo nombre, si se introduce una contraseña esta se sobrescribirá.

The screenshot shows the 'PANEL ADMINISTRADOR' interface. On the left is a sidebar with options: 'Mundos y minijuegos', 'Revisar usuarios', 'Añadir mundo', 'Añadir minijuego', and 'Añadir usuario'. The main content area has two sections. The top section, 'Añadir nuevo usuario', contains a form with the following fields: 'Nombre:' (text input with 'serbarr'), 'Contraseña:' (password input with masked characters), 'Rol de usuario:' (dropdown menu with 'Usuario' selected), and 'Grupo:' (text input with 'L1'). A red 'Añadir' button is below these fields. The bottom section, 'Añadir lista de usuarios nuevos', has a 'Fichero:' label, a 'Seleccionar archivo' button, and the text 'Ningún archivo seleccionado'. A red 'Añadir' button is below this section. Red arrows in the image point to the input fields in the first section.

En el caso de querer agregar una lista de usuarios hay que introducirla con las siguientes restricciones:

- Formato validos: .txt, .csv, .xls, .doc.
- Cada fila constara de los siguientes campos: Grupo;Rol;Nombre;Contraseña; y tiene que terminar en “;”.
- Roles permitidos: U (Usuario), D (Desarrollador) y A (Administrador).
- Cada columna tiene que estar separada por “;” sin espacios entre columna y columna. EJ: L1;U;fragome;f873;
-



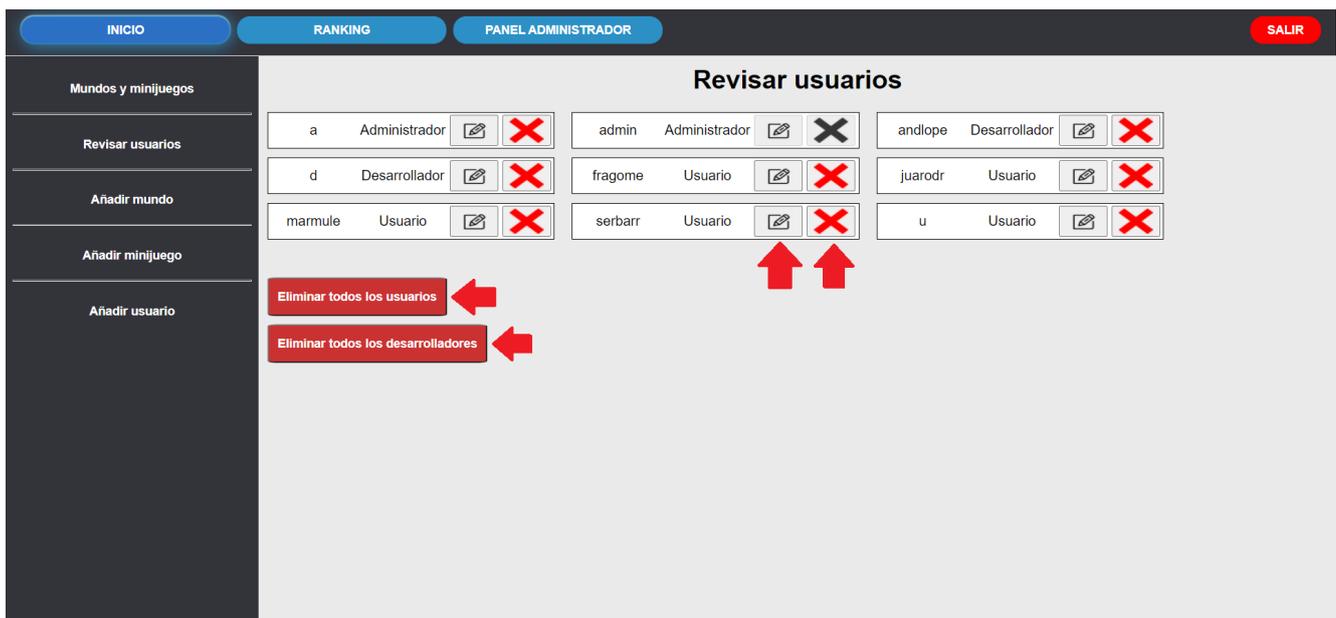
Por ultimo una vez añadido los usuarios se pueden revisar en "Revisar usuarios". Que se encuentra en el panel lateral del "Panel de Administrador".



Aquí se puede revisar la lista de usuarios agregados al sistema, si nombre y su rol.



También se puede modificar un usuario y eliminarlo. Como también se puede eliminar todos los usuarios de la aplicación (excepto a los desarrolladores y administradores) o eliminar unicamente a los desarrolladores.



INICIO RANKING PANEL ADMINISTRADOR SALIR

Modificar usuario

Nombre:

Nueva contraseña:

Nuevo rol de usuario:

Nuevo grupo:

Mundos y minijuegos

Revisar usuarios

Añadir mundo

Añadir minijuego

Añadir usuario