



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA  
Mención en Computación

---

**Análisis y actualización de una plataforma para  
la gamificación de asignaturas**

---

**Alumno:**

**D. Rubén Garduño Barbier**

**Tutores:**

**D.<sup>a</sup> Alma María Pisabarro Marrón  
Dr. D. Carlos Enrique Vivaracho Pascual**



# Agradecimientos

Agradezco de corazón el encomiable trato y atención que he recibido de manos de mis tutores, D.<sup>a</sup> Alma María Pisabarro Marrón y Dr. D. Carlos Enrique Vivaracho Pascual dado su esfuerzo incondicional por enseñarme todo lo posible e involucrarse tanto en el proyecto.

También me gustaría agradecer a todos los profesores y profesoras que me han guiado a lo largo de mi vida y trayectoria académica, puesto que gracias a su labor y dedicación, hoy puedo estar donde estoy.

Mi agradecimiento hacia mis familiares, mis amigos y mi pareja no tiene límites. Gracias por estar ahí en todo momento, confiar en mí y demostrarme el valor del esfuerzo, la constancia y la responsabilidad en cualquier ámbito de la vida.

¡ Muchas gracias !





# Prólogo

La enseñanza es, sin ninguna duda, un arte que ha ido evolucionando a lo largo de los siglos. Ya desde sus inicios, aquellos que recibían educación se distinguían de aquellos que, por una razón u otra, no tenían acceso a la misma. Y es que ésta dota al que la recibe de algo fundamental: capacidad de evolución.

Las personas estamos en un cambio continuo, enfrentándonos a nuevos retos a medida que pasa el tiempo. Estar preparado para ello... depende de nosotros. Depende de nuestra osadía a la hora de enfrentar el cambio, de lo preparados que estemos, de la magnitud del cambio... en definitiva, depende de tantos factores, que lo mejor que podemos hacer es prepararnos a conciencia para ello.

Así como las personas evolucionamos, los métodos de aprendizaje y las técnicas de enseñanza han ido evolucionando de su mano, adaptándose a los recursos disponibles en cada momento y lugar. Las nuevas tecnologías nos han aportado un punto de apoyo hasta ahora desconocido, que abre un nuevo abanico de posibilidades con las que aprender puede volverse tan ameno y divertido como uno desee.

*La educación es el arma más poderosa que puedes usar para cambiar el mundo.*

(Nelson Mandela)



# Índice general

<b>1. Introducción</b>	<b>15</b>
1.1. Motivación . . . . .	16
1.2. Estructura . . . . .	17
<b>2. Objetivos</b>	<b>19</b>
2.1. Objetivo general . . . . .	19
2.2. Objetivos específicos . . . . .	19
<b>3. Gestión del proyecto</b>	<b>21</b>
3.1. Planificación del proyecto . . . . .	21
3.1.1. Planificación inicial . . . . .	21
3.1.2. Modificaciones frente a la planificación inicial . . . . .	22
3.2. Recursos utilizados . . . . .	23
3.3. Costes del proyecto . . . . .	24
<b>4. Plataforma de trabajo y conjunto de datos</b>	<b>27</b>
4.1. Estructura de la plataforma . . . . .	27
4.2. Interfaz de la plataforma . . . . .	28
4.2.1. Página de <i>login</i> . . . . .	28
4.2.2. Página <i>Inicio</i> . . . . .	29
4.2.3. Página <i>Mundos</i> . . . . .	30
4.2.4. Página <i>Minijuegos</i> . . . . .	31
4.2.5. Página <i>Ránking</i> . . . . .	32
4.2.6. Página <i>Panel Administrador</i> . . . . .	33
4.3. Conjunto de datos . . . . .	38
<b>5. Resultados</b>	<b>41</b>
5.1. Eliminación de determinadas dependencias con <i>Unity</i> . . . . .	41
5.1.1. Método de llamada antes de la modificación . . . . .	41
5.1.2. Método de llamada después de la modificación . . . . .	41
5.2. Análisis y corrección de brechas de seguridad en el sistema actual . . . . .	44
5.2.1. Ataque a la base de datos desde software foráneo . . . . .	44
5.2.2. Réplicas de peticiones <i>POST</i> modificando la base de datos . . . . .	48
5.3. Inclusión del apartado <i>Ránking por juego</i> en la plataforma . . . . .	52
5.3.1. <i>Ránking por juego</i> para los usuarios administradores . . . . .	52
5.3.2. <i>Ránking por juego</i> para el resto de usuarios . . . . .	55
5.4. Diseño de un <i>Dashboard</i> . . . . .	57
5.4.1. Ubicación del <i>Dashboard</i> . . . . .	57
5.4.2. Información general . . . . .	58

5.4.3. Pestaña <i>Usuarios</i> . . . . .	60
5.4.4. Pestaña <i>Grupos</i> . . . . .	65
5.4.5. Pestaña <i>Mundos</i> . . . . .	69

<b>6. Conclusiones</b>	<b>73</b>
------------------------	-----------

# Índice de figuras

1.1. Modelo de trabajo en Cascada . . . . .	16
3.1. Planificación inicial del <i>TFG</i> . . . . .	22
3.2. Setup inicial de la <i>VM</i> . . . . .	23
4.1. Estructura de directorios de la plataforma . . . . .	27
4.2. Mapa web de la plataforma . . . . .	28
4.3. Pantalla de acceso a la plataforma . . . . .	29
4.4. Pantalla de inicio para <i>Jugadores</i> . . . . .	30
4.5. Pantalla de inicio para <i>Administradores</i> . . . . .	30
4.6. Pantalla de <i>Mundos</i> . . . . .	31
4.7. Pantalla de <i>Minijuegos</i> . . . . .	31
4.8. Pantalla de <i>Ránking individual</i> . . . . .	32
4.9. Pantalla de <i>Ránking grupal</i> . . . . .	33
4.10. Pestaña <i>Mundos y minijuegos</i> . . . . .	34
4.11. Pestaña <i>Revisar usuarios</i> . . . . .	35
4.12. Fin de la pestaña <i>Revisar usuarios</i> . . . . .	35
4.13. Interfaz <i>Modificar usuarios</i> . . . . .	36
4.14. Pestaña <i>Añadir Mundo</i> . . . . .	36
4.15. Pestaña <i>Añadir Minijuego</i> . . . . .	37
4.16. Pestaña <i>Añadir Usuario</i> . . . . .	38
4.17. Modelo de datos del sistema . . . . .	38
5.1. Modelo de Madurez de Richardson . . . . .	42
5.2. Estructura de la plataforma tras añadir el patrón fachada . . . . .	43
5.3. Vulnerabilidad en la llamada a <i>guardar.php</i> desde el apartado <i>Depurador</i> . . . . .	44
5.4. Comunicación entre <i>guardar.php</i> y otros ficheros . . . . .	45
5.5. Vulnerabilidad en la llamada a <i>guardar.php</i> . . . . .	46
5.6. Petición de tipo <i>POST</i> a <i>guardar.php</i> en el servicio <i>Postman</i> . . . . .	46
5.7. Comunicación entre ficheros tras aplicar el <i>patrón fachada</i> . . . . .	47
5.8. Pestaña <i>Depurador</i> en las <i>Herramientas para desarrolladores</i> tras la aplicación del <i>patrón fachada</i> . . . . .	48
5.9. Vulnerabilidad en el envío de una petición a <i>executeGameRequest.php</i> . . . . .	49
5.10. Opción de edición y reenvío de una petición . . . . .	49
5.11. Forma de edición y reenvío de la petición . . . . .	50
5.12. Cuerpo de la solicitud tras el encriptamiento de los datos enviados . . . . .	51
5.13. Nueva pestaña de <i>Ver clasificación</i> en el Panel Administrador . . . . .	52
5.14. Nueva pestaña de <i>Ver clasificación</i> en el Panel Administrador . . . . .	53
5.15. <i>Ránking individual</i> para el juego <i>Apuntados</i> . . . . .	54
5.16. <i>Ránking individual</i> para el juego <i>Apuntados</i> en el grupo <i>G10</i> . . . . .	55

5.17. Nueva pestaña de <i>Ránking por juego</i> para usuarios no administradores . . . . .	56
5.18. Página <i>Hall of Fame</i> . . . . .	56
5.19. Ránking individual para el juego <i>Caída de datos</i> . . . . .	57
5.20. Ubicación y composición general del <i>dashboard</i> . . . . .	58
5.21. Vista general del dashboard con la subpestaña <i>Media de intentos</i> . . . . .	59
5.22. Vista general del dashboard con la subpestaña <i>Tiempo medio de juego</i> . . . . .	60
5.23. Pestaña <i>Usuarios</i> del <i>Dashboard</i> . . . . .	61
5.24. Pestaña <i>Usuarios</i> del <i>Dashboard</i> con los usuarios desplegados . . . . .	61
5.25. Pestaña <i>Jugadores</i> del dashboard con evolución de puntos por minijuego . . . . .	62
5.26. Pestaña <i>Jugadores</i> del dashboard con evolución de estrellas por minijuego . . . . .	63
5.27. Pestaña <i>Jugadores</i> del dashboard con el selector de mundos desplegado . . . . .	63
5.28. Pestaña <i>Jugadores</i> del dashboard con evolución de puntos por niveles para un minijuego concreto . . . . .	64
5.29. Pestaña <i>Jugadores</i> del dashboard con evolución de estrellas por niveles para un minijuego concreto . . . . .	64
5.30. Pestaña <i>Grupos</i> inicial . . . . .	65
5.31. Pestaña <i>Grupos</i> con el selector de grupos desplegado . . . . .	66
5.32. Pestaña <i>Grupos</i> con la evolución de puntos por minijuego . . . . .	67
5.33. Pestaña <i>Grupos</i> con la evolución de estrellas por minijuego . . . . .	67
5.34. Pestaña <i>Grupos</i> con los jugadores activos por minijuego . . . . .	68
5.35. Pestaña <i>Grupos</i> con la evolución de puntos por niveles para un minijuego concreto . . . . .	68
5.36. Pestaña <i>Grupos</i> con la evolución de estrellas por niveles para un minijuego concreto . . . . .	69
5.37. Pestaña <i>Mundos</i> inicial . . . . .	70
5.38. Pestaña <i>Mundos</i> con el selector de mundos desplegado . . . . .	70
5.39. Pestaña <i>Mundos</i> con la media de horas por minijuego . . . . .	71
5.40. Pestaña <i>Mundos</i> con la media de estrellas por minijuego . . . . .	71
5.41. Pestaña <i>Mundos</i> con la media de puntos por minijuego . . . . .	72
5.42. Pestaña <i>Mundos</i> con los usuarios totales por minijuego . . . . .	72

# Índice de tablas

3.1. Recursos software necesarios para la consecución del proyecto . . . . .	24
3.2. Costes de los recursos hardware necesarios para el proyecto . . . . .	25
3.3. Costes de las licencias de software utilizadas en el proyecto . . . . .	25
3.4. Costes de los recursos humanos utilizados en el proyecto . . . . .	25
3.5. Coste total del proyecto . . . . .	26





# Resumen

La gamificación alberga un conjunto de técnicas y metodologías para fomentar y mejorar el proceso de aprendizaje. Es por ello que su aplicación dentro del plan de estudios de una asignatura puede acarrear múltiples beneficios, tanto para los alumnos, como para los docentes al cargo.

El presente proyecto ha sido realizado como *Trabajo Fin de Grado* en el Grado en Ingeniería Informática en la Universidad de Valladolid. En el mismo, se da solución a un conjunto de requerimientos y mejoras sobre una plataforma creada previamente, para la gamificación de la asignatura *Fundamentos de Valladolid*, demandadas por los docentes de la asignatura.

Con el fin de adaptar a la nueva metodología de *Unity* (motor de desarrollo de videojuegos), se decidió implantar una nueva forma de acceder a los minijuegos desde la plataforma, consiguiendo una forma de proceder más unificada dentro del desarrollo web. También se realizó un estudio de la seguridad de la plataforma, implementando las medidas pertinentes para solventar las vulnerabilidades detectadas.

Otra de las modificaciones realizadas fue la inclusión del acceso al ránking individual y por grupos de cada minijuego, permitiendo su visualización en el navegador a los *alumnos* y *desarrolladores*, además de permitir a los usuarios *administradores* descargar este ránking para el uso particular de esta información.

Por último, se diseñó un *Dashboard* que mostrase numerosos aspectos de la información administrada en la base de datos del sistema.

**Palabras clave:** Gamificación, seguridad, dashboard, ránking, minijuego



# Abstract

Gamification houses a set of techniques and methodologies to promote and improve the learning process. This is why its application within the study plan of a subject can bring multiple benefits, both for the students and for the teachers in charge.

This project has been carried out as *End of Degree Project* in the Degree in Computer Engineering at the University of Valladolid. In it, a solution is given to a set of requirements and improvements on a previously created platform, for the gamification of the subject *Basis of Programming*, demanded by the teachers of the subject.

In order to adapt to the new *Unity* methodology, it was decided to implement a new way of accessing the minigames from the platform, achieving a more unified way of proceeding within web development. A study of the security of the platform was also carried out, implementing the pertinent measures to solve the vulnerabilities detected.

Another of the modifications made was the inclusion of access to the individual and group ranking of each minigame, allowing *students* and *developers* to view it in the browser, in addition to allowing users *administrators* download this ranking for the particular use of this information.

Finally, a *Dashboard* was designed to show numerous aspects of the information managed in the system's database.

**Key words:** Gamification, security, dashboard, ranking, minigame



# Capítulo 1

## Introducción

El proceso de aprendizaje de cualquier tarea en la vida depende de multitud de factores, algunos de ellos medibles, y otros cuantos que son del todo impredecibles. Uno de esos factores, determinantes para alcanzar el éxito, es la motivación [1]. Ese detalle que marca la diferencia, se obtiene fomentando el interés de aquel o aquella que esté aprendiendo, logrando así aumentar su potencial.

Hace años, el término *videojuego* (juego electrónico que se visualiza en una pantalla [2]) se asociaba más comúnmente a fines lúdicos y de divertimento, dejando a un lado otros fines de interés como pueden ser el estudio o el aprendizaje. Con el paso de los años, y a medida que el mundo y la tecnología evolucionaban a la par, este sesgo conceptual fue disminuyendo de forma progresiva, hasta llegar a la realidad actual en la que nos encontramos.

Hoy en día, los videojuegos se crean y desarrollan en multitud de ámbitos que traspasan las barreras del mero entretenimiento, cumpliendo objetivos tan importantes como ayudar en la rehabilitación de accidentes cerebrovasculares sobre la capacidad de caminar y el equilibrio [3], producir mejoras en la memoria y un aumento de la creatividad y la capacidad de resolución de problemas [4], o bien enseñar paso a paso una determinada tarea a partir de retos y juegos.

Así, la *gamificación* puede definirse como el uso de mecánicas y estéticas basadas en juegos para involucrar a la gente, motivar la acción, promover el aprendizaje y resolver problemas [5]. Aunque el término suene distante del concepto de aprender de una forma “seria”, la realidad es que la aplicación de estas técnicas consiguen amenizar la tarea y evitar el fracaso [5].

Enfocando nuestra atención en el ámbito académico, el uso de diversas técnicas de gamificación en el desarrollo de una asignatura puede aumentar su material didáctico, dando a los estudiantes un pilar más sobre el que apoyarse y consiguiendo asentar los conocimientos de los estudiantes con mayor facilidad.

Es por ello que el presente *TFG* se enmarca dentro del contexto de un proyecto más grande, en el que se gamificó la asignatura *Fundamentos de Programación*, obligatoria e impartida en el Grado en Ingeniería Informática y en el Grado en Estadística en la Universidad de Valladolid. Esta iniciativa ya se aplica en la docencia, y ha sido evaluada en alumnos de la asignatura, obteniendo óptimos resultados tras la dinámica.

Tras pasar el proceso de explotación, el fin del proyecto actual consistió en realizar el man-

tenimiento habitual, para resolver o dar respuesta a los requerimientos de los docentes, para un mejor aprovechamiento y resultado de la plataforma y así mejorar el proceso de aprendizaje, tanto en la asignatura trabajada, como en otras futuras asignaturas que puedan sumarse a esta dinámica de trabajo.

## 1.1. Motivación

En el desarrollo de un proyecto de ingeniería del software, aclarar los objetivos definidos en cada fase es necesario, para asegurar así su éxito y correcto funcionamiento del producto o productos desarrollados. Sin embargo, las necesidades de los usuarios pueden cambiar o aumentar a lo largo del tiempo, lo que implica adaptar el producto a estas nuevas necesidades.

En el año 2020, la Escuela de Ingeniería Informática de la Universidad de Valladolid lanzó un proyecto para gamificar la asignatura de *Fundamentos de Programación*, para la que se desarrolló una plataforma de almacenamiento de diferentes minijuegos relacionados con los conceptos básicos de la programación y las estructuras de datos [6]. Este proyecto se realizó siguiendo un modelo en cascada (figura 1.1), siendo la fase de mantenimiento la fase final del modelo de trabajo y en la que se encuentra actualmente.

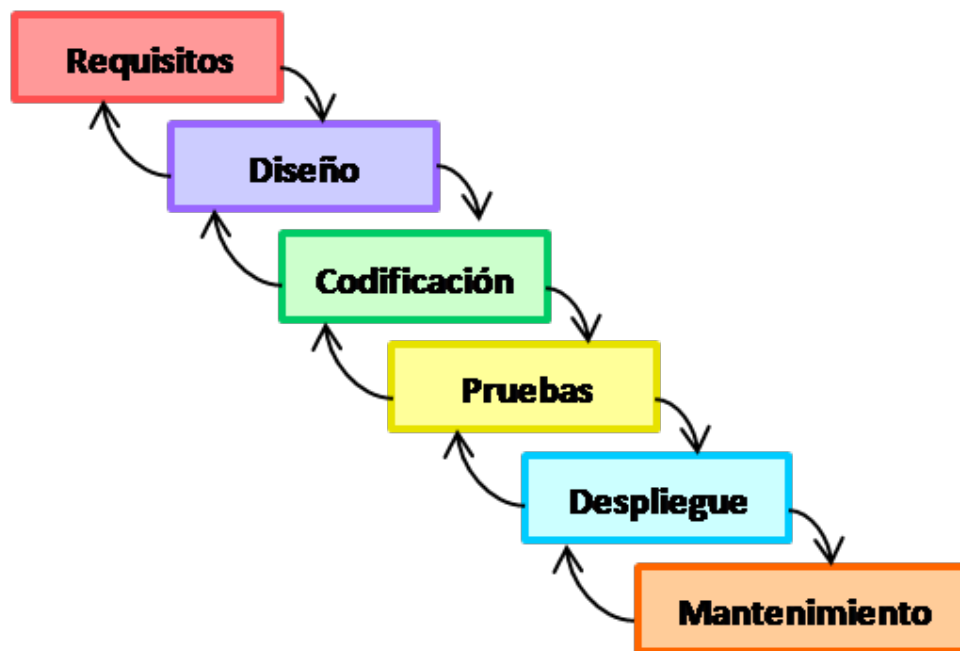


Figura 1.1: Modelo de trabajo en Cascada

Al poner a disposición del alumnado esta herramienta, y no detectar errores en el funcionamiento de la dinámica, se detectaron por parte de los docentes responsables un conjunto de funcionalidades y necesidades concretas para facilitar el control y monitorización, atendiendo a las solicitudes de los usuarios administradores y desarrolladores, así como otras funcionalidades para los propios alumnos, que serán analizadas y descritas a lo largo del presente *TFG*.

## 1.2. Estructura

Esta memoria ha sido estructurada como sigue:

- El *Capítulo 2* recoge el objetivo principal del *TFG*, así como los objetivos específicos perseguidos.
- El *Capítulo 3* describirá la gestión y planificación del proyecto, así como los recursos necesarios para la consecución de los objetivos planteados.
- El *Capítulo 4* presentará la estructura e interfaz de la plataforma de trabajo, así como su conjunto de datos asociado y sus características.
- El *Capítulo 5* describirá los diferentes resultados obtenidos, para cada uno de los objetivos planteados en este *TFG*.
- Por último, a lo largo del *Capítulo 6* se mencionarán y expondrán las conclusiones extraídas a lo largo del desarrollo del *TFG*.





# Capítulo 2

## Objetivos

A lo largo del presente *TFG*, se ha perseguido alcanzar la consecución de un conjunto de objetivos que se enuncian a continuación.

### 2.1. Objetivo general

El objetivo general o principal que se planteó a lo largo de todo el proyecto, y que no se ha perdido de vista durante todo su desarrollo, fue el análisis y estudio de la plataforma existente para la gamificación, analizando y solucionando posibles brechas en su seguridad, así como la adición de nuevas funcionalidades a petición de los docentes.

### 2.2. Objetivos específicos

Para alcanzar el objetivo general propuesto, se plantearon una serie de objetivos específicos que concretasen las distintas necesidades en cada caso. Estos son los que siguen:

1. **Eliminación de determinadas dependencias con *Unity***

Se intentó conseguir la eliminación de la existente dependencia con *Unity* en la realización de los juegos, a petición de los docentes encargados de la gestión y administración de la plataforma.

2. **Análisis y corrección de brechas de seguridad en el sistema actual**

Con este objetivo, se buscaron brechas en la seguridad de la plataforma, evitando fugas de información y posibles modificaciones ilícitas de la base de datos, y se implantaron las correcciones pertinentes para solventarlo.

3. **Inclusión del apartado “Ránking por juego” en la plataforma**

Este tercer objetivo específico consistió en incluir en el sistema la posibilidad de consultar la clasificación individual por juego y grupo así como la clasificación por grupos de cada juego. Se añadió también la opción de descargar un fichero con la clasificación para el caso de los usuarios *Administradores*.

4. **Diseño e implementación de un *Dashboard***

Otro de los objetivos planteados se basó en diseñar e implementar un panel analítico o *dashboard* con información relevante para los investigadores, que pudiese facilitar el seguimiento y evolución de los alumnos. Este objetivo tuvo que ser modificado en alcance dejándolo finalmente en el diseño de dicho Dashboard.



# Capítulo 3

## Gestión del proyecto

Conseguir planificar de manera correcta y eficiente un proyecto de software es una tarea compleja, vital para lograr con éxito la consecución de los objetivos del mismo. La gestión de los tiempos, la anticipación ante posibles contratiempos y la visión global de las metas a conseguir son fundamentales, debiendo así dedicar a ello el tiempo suficiente.

### 3.1. Planificación del proyecto

La duración estimada para la realización del presente *TFG* es de un total de 300 horas de trabajo. Esta cifra se obtiene teniendo en cuenta la cantidad de créditos que supone la realización del mismo en el *Grado en Ingeniería Informática* (12 ECTS) y las horas invertidas en cada uno de estos créditos (25 horas).

Teniendo esta información en cuenta, así como la fecha de inicio del proyecto (7 de Febrero de 2022, aunque la propuesta se lanzó en Diciembre de 2021), se procederá a explicar la planificación inicial planteada y las modificaciones que esta sufrió a lo largo de su ejecución.

#### 3.1.1. Planificación inicial

Una de las primeras tareas a realizar en la ejecución del proyecto fue hacer una planificación y calendarización del mismo, fijando los objetivos de forma clara y reflejando en el calendario las distintas etapas a seguir. La figura 3.1 muestra esta primera planificación.

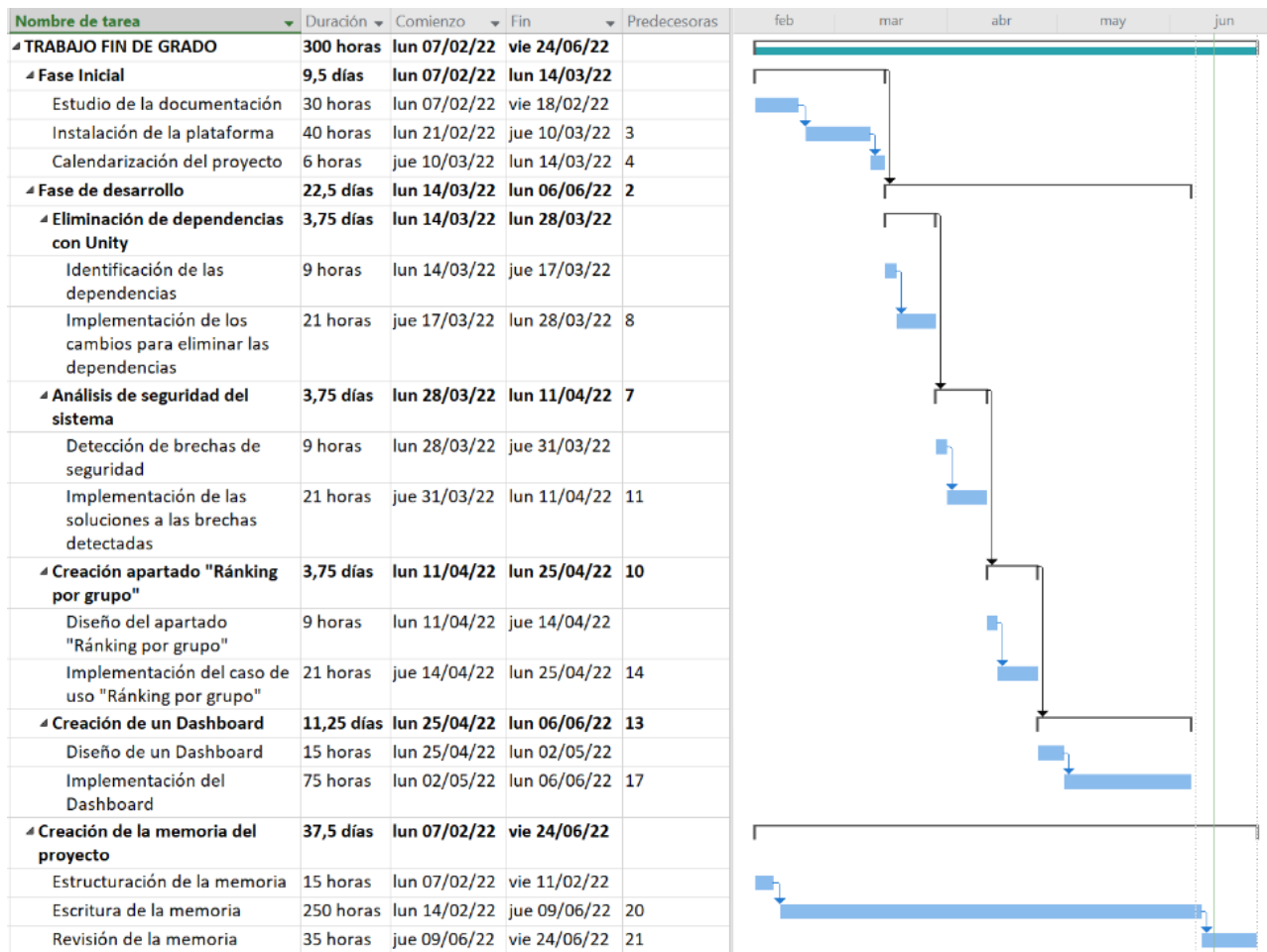


Figura 3.1: Planificación inicial del *TFG*

La tarea con mayor duración prevista fue el desarrollo del informe del proyecto, ya que se planteó su planificación y escritura desde el mismo momento de inicio del trabajo. De manera paralela, se desarrollaron varias fases, con objetivos específicos en cada una de ellas. La fase inicial estaría compuesta por la lectura y comprensión de la documentación, la instalación de la plataforma en la máquina virtual proporcionada por el Departamento de Informática de la UVa y la propia calendarización del proyecto.

Posteriormente, la fase de desarrollo se dividirá en las distintas sub tareas relacionadas con la consecución de los objetivos planteados en la *Sección 2*. Entre estos objetivos, cabe destacar que la creación del *Dashboard* en la plataforma fue la tarea con mayor peso, teniendo un peso total de 90 horas (figura 3.1).

### 3.1.2. Modificaciones frente a la planificación inicial

Aunque se planteó una minuciosa planificación desde los principios del *TFG*, esta se vió condicionada por una serie de imprevistos que implicaron modificar la planificación inicial en varios aspectos.

Dentro de los condicionantes más notorios, se encuentran:

- Al comienzo del trabajo, en la fase inicial del proyecto, en la configuración del puesto de trabajo, dentro de la sub tarea de *Instalación de la plataforma*, la máquina virtual

facilitada por el Departamento de Informática fue dotada con las siguientes capacidades (5 Gby de Ram, 4 CPU y 15 Gby) las cuales, tras varias pruebas, no cubrían los requisitos necesarios para el uso y manejo de la plataforma, lo que provocó una fase de ajuste de los requisitos, que duró en total dos semanas hasta encontrar la configuración ideal, evitando el sobre dimensionamiento, en (8Gby de Ram, 4 CPU y 15 Gby). Tras configurar este y otros elementos necesarios del setup de trabajo, se pudo continuar con el desarrollo del proyecto.

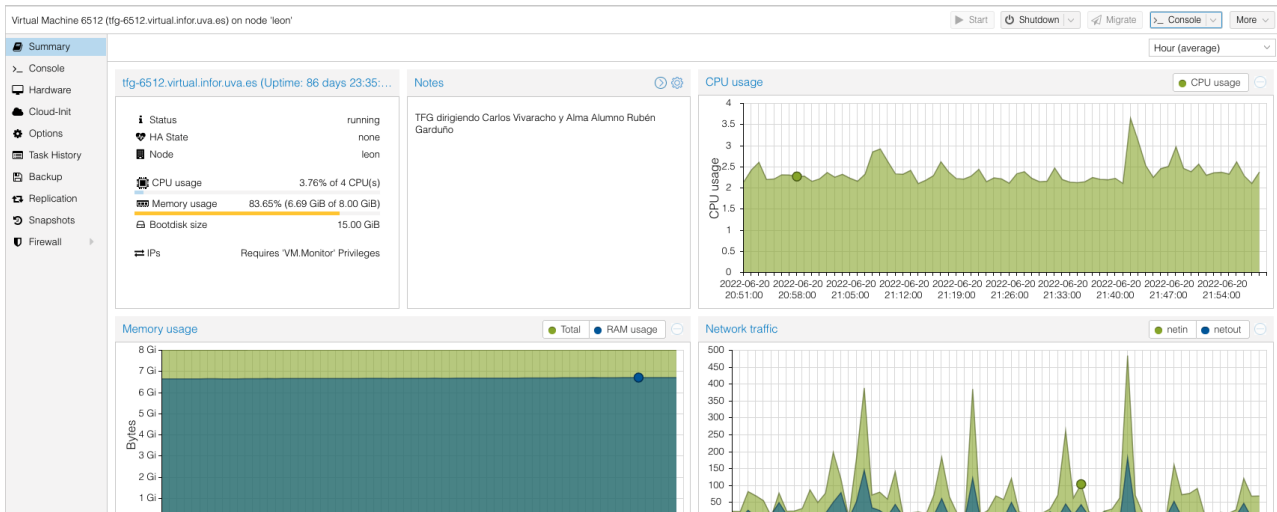


Figura 3.2: Setup inicial de la VM

- Tras el análisis de las brechas de seguridad se localizaron más brechas de seguridad de las previstas inicialmente en la plataforma lo que motivó una reestimación del esfuerzo inicialmente calculado para su resolución, con un considerable impacto en la planificación global. La nueva estimación agregó 60 horas más a esta tarea, lo que provocó un ajuste del alcance de los objetivos iniciales, reduciendo el alcance de la tarea cuarta adaptando el objetivo al tiempo restante y quedando marcado en el diseño del *Dashboard*. La premisa o requisito aplicado para esta toma de decisión fue la priorización de los aspectos de seguridad sobre el resto.
- Como consecuencia a esta circunstancia, se decidió ocupar parte del tiempo asignado a la tarea “Diseño e implementación de un Dashboard”, lo que resultó en una reducción del tiempo de 90 a 30 horas. Por lo tanto, en esta tarea se pudo alcanzar únicamente el hito del diseño, sin llegar a implementar el Dashboard.

En último lugar, es necesario remarcar que, aunque el presente documento se comenzó a escribir desde el inicio del proyecto, la mayor parte del mismo ha sido escrito durante el último mes del proyecto.

## 3.2. Recursos utilizados

Asociada directamente a la gestión de tiempos en un proyecto de ingeniería, se encuentra la planificación y gestión de los recursos del proyecto. Entre estos recursos, se distinguirán:

- **Recursos Hardware:** Se utilizó la computadora personal del alumno, así como la máquina virtual asignada por el Departamento de Informática de la Universidad de Valladolid.

- **Recursos Software:** Se hizo uso de varias herramientas software, algunas de uso gratuito, y otras con un coste asociado, reflejados en la tabla 3.1.

Nombre	Descripción
Visual Studio Code	IDE (Entorno de Desarrollo Integrado) usado para el desarrollo de las soluciones.
MySQL	Sistema gestor de bases de datos relacionales utilizado para almacenar y gestionar los datos de la plataforma.
PHPMyAdmin	Aplicación web para gestionar la base de datos MySQL del sistema.
Overleaf	Editor de textos en formato LaTeX utilizado para redactar la memoria del proyecto.
Creately	Espacio de trabajo visual con el que se realizó el mapa web de la plataforma.
Diagrams.net	Aplicación web gratuita con la que se modeló la estructura de directorios del sistema.
Postman	Herramienta de prueba de APIs de tipo REST con la que se realizaron varias de las pruebas de seguridad del sistema.
Microsoft Project	Software de administración de proyectos utilizado para la planificación y administración del proyecto.
Máquina virtual UVa	Máquina virtual de tipo Linux utilizada para realizar las modificaciones pertinentes.
Balsamiq	Herramienta de creación de Mockups utilizada para el diseño del <i>dashboard</i> .

Tabla 3.1: Recursos software necesarios para la consecución del proyecto

- **Recursos humanos:** Se hizo una estimación del coste que abarcaría la totalidad del proyecto en cuanto a la gestión del personal involucrado en el mismo. Dado que se trata de un Trabajo Fin de Grado, se consideró la necesidad de un único desarrollador.

### 3.3. Costes del proyecto

Directamente relacionados con los recursos necesarios, se encuentran los costes asociados a estos, que servirán para construir un presupuesto total para la realización del proyecto.

En lo referente a los recursos hardware, la tabla 3.2 recoge este coste asociado, teniendo en cuenta el ordenador personal del alumno.

Concepto	Cantidad	Nº Horas	Pre- cio/hora	Coste Total
Ordenador portátil personal (AMD Ryzen7, 16 Gby Ram, 1 TBy HD, Windows 11)	1	300	0,10 €	30,00 €
<b>TOTAL</b>				30,00 €

Tabla 3.2: Costes de los recursos hardware necesarios para el proyecto

La tabla 3.3 contiene el resumen de los costes asociados a las licencias de software necesarias. Se han priorizado el uso de herramientas OpenSource y las licencias utilizadas son gratuitas a excepción de Microsoft Project y Balsamiq.

Nombre	Cantidad	Coste
Visual Studio Code	1	0,00 €
MySQL	1	0,00 €
PHPMyAdmin	1	0,00 €
Overleaf	1	0,00 €
Creately	1	0,00 €
Diagrams.net	1	0,00 €
Postman	1	0,00 €
Microsoft Project	1	1.509,00 €
Máquina Virtual UVa	1	0,00 €
Balsamiq	1	85,48 €
<b>Coste total</b>		<b>1.594,48 €</b>

Tabla 3.3: Costes de las licencias de software utilizadas en el proyecto

Por otra parte la tabla 3.4 muestra los costes asociados a recursos humanos. El sueldo medio anual de un desarrollador de software en España es de 26.000,00 euros [7]. Sumando los gastos empresariales (un 33 % del total), esta cifra ascendería a 34.580,00 euros anuales que, divididos en 12 pagas, y calculando el precio por hora, será de 18,01 euros por hora.

Puesto	Cantidad	Precio/hora	Nº Horas	Coste total
Desarrollador de software	1	18,01€	300	5.403,00 €

Tabla 3.4: Costes de los recursos humanos utilizados en el proyecto

Teniendo en cuenta las 300 horas previstas al inicio del proyecto, el coste por personal previsto sería de **5.403,00 €**. Así, el presupuesto total para el presente *TFG* se muestra en la tabla 3.5.

<b>Concepto</b>	<b>Coste total</b>
Recursos Hardware	30,00 €
Licencias Software	1.594,48 €
Recursos Humanos	5.403,00 €
<b>TOTAL</b>	<b>7.027,48 €</b>

Tabla 3.5: Coste total del proyecto

Así, el coste total previsto para la consecución del proyecto ascendería a **7.027,48 €**.



# Capítulo 4

## Plataforma de trabajo y conjunto de datos

Este capítulo describirá en detalle la plataforma “Programa Jugando”, así como sus características y posibilidades y su base de datos asociada, al comienzo del desarrollo de este proyecto. Por lo tanto, no se reflejarán los cambios ni las nuevas funcionalidades (que serán expuestos en el Capítulo 5).

### 4.1. Estructura de la plataforma

Con el fin de poder trabajar y realizar cambios en modo local, para posteriormente implementar estos cambios en la plataforma creada anteriormente, se trabajó en una máquina virtual prestada temporalmente por el Departamento de Informática. La estructura de directorios de la plataforma viene dada en la figura 4.1.

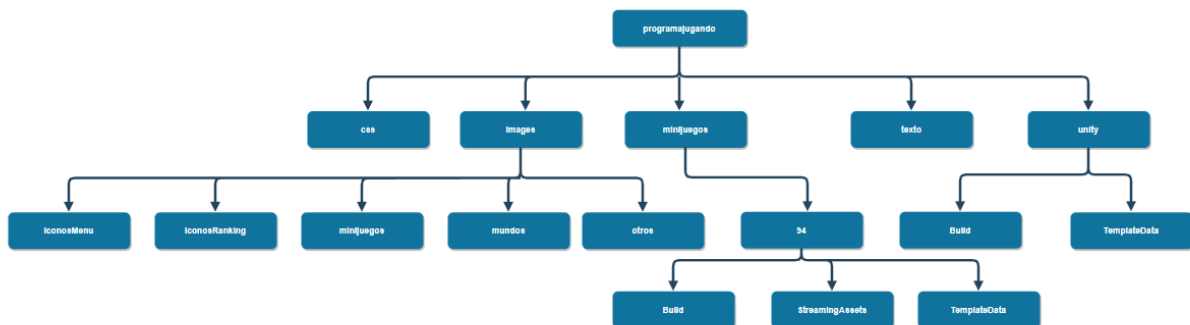


Figura 4.1: Estructura de directorios de la plataforma

Esta estructura consta de un directorio padre (*programajugando*) que contiene al resto de directorios de la plataforma. En esta ubicación se encuentran también la mayoría de ficheros y documentos que dan funcionalidad al sistema. Dentro de los subdirectorios presentes, se encuentran:

- **css**: Contiene los ficheros relacionados con los estilos o estéticas de la plataforma.
- **images**: Alberga las distintas imágenes utilizadas en el sistema, organizándolas en los subdirectorios correspondientes.

- **minijuegos**: Contiene la información concreta relativa a cada minijuego. Esta se encuentra a su vez en un subdirectorio, etiquetado con el identificador del minijuego correspondiente (la figura 4.1 representa el ejemplo del minijuego 94).
- **texto**: Almacena ficheros de datos en formato de texto plano.
- **unity**: Este último directorio contiene los ficheros correspondientes al *framework* Unity.

El sistema de ficheros, ya con las modificaciones pertinentes, se encuentra disponible en la siguiente dirección: [https://gitlab.inf.uva.es/minijuegos\\_gamificacion/plataforma\\_v2](https://gitlab.inf.uva.es/minijuegos_gamificacion/plataforma_v2)

## 4.2. Interfaz de la plataforma

La plataforma original, ubicada en la dirección <https://greidi.infor.uva.es/programajugando>, está compuesta por una serie de interfaces características que cumplen una determinada función en cada caso. El mapa web de la plataforma se presenta en la figura 4.2, y refleja una visión de alto nivel de las páginas por las que es posible navegar en el sistema.

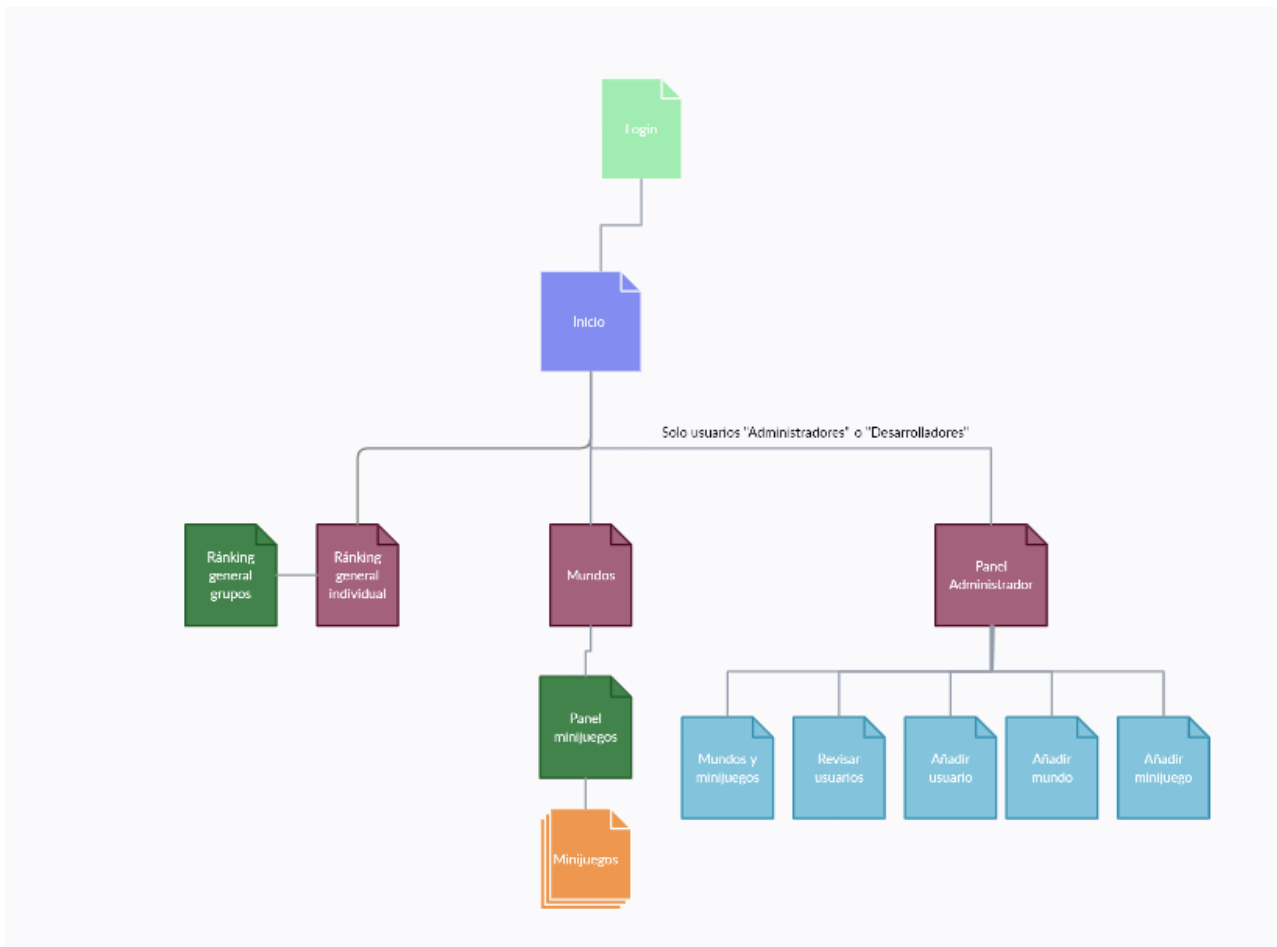


Figura 4.2: Mapa web de la plataforma

### 4.2.1. Página de *login*

La primera pantalla que un usuario se encuentra al entrar en la aplicación es una ventana de *login*, reflejada en la figura 4.3:

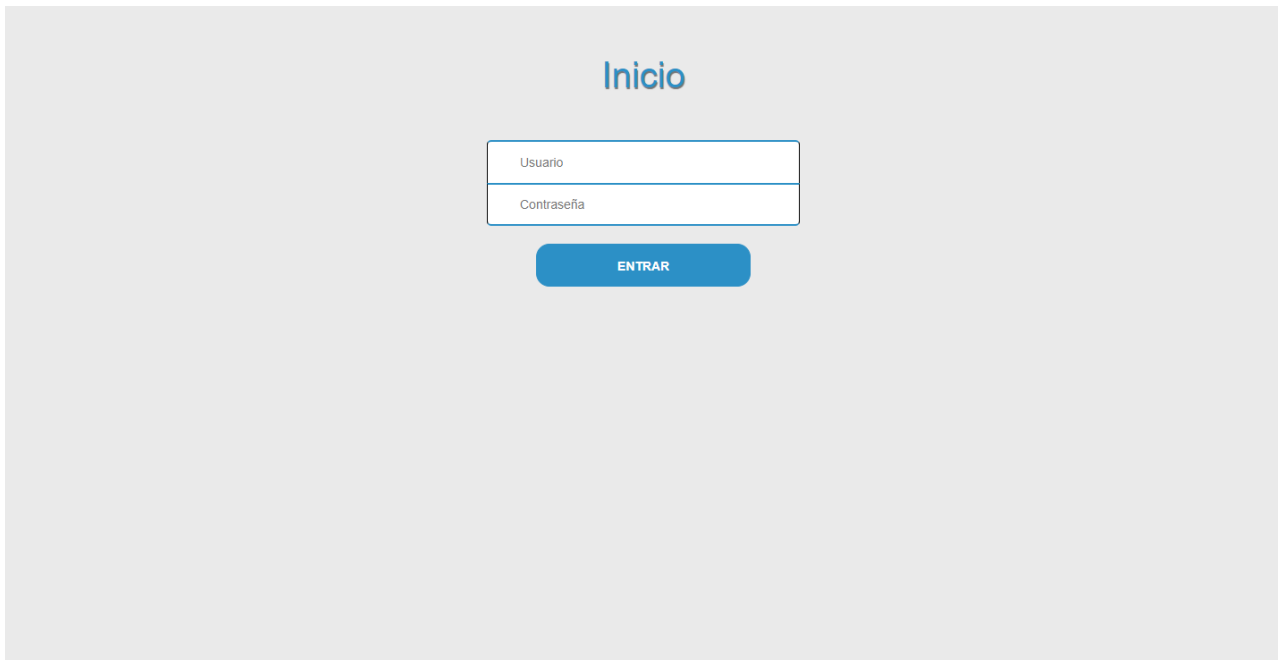


Figura 4.3: Pantalla de acceso a la plataforma

Esta se compone de dos campos, *Usuario* y *Contraseña*, en las que el usuario puede introducir sus credenciales para identificarse y poder así acceder a la plataforma.

#### 4.2.2. Página *Inicio*

Esta página se presenta inmediatamente después de registrarte en la página de *login*, y varía en función del tipo de usuario que acceda, ya que existen 3 tipos de usuarios:

- ***Usuario jugador***: Tiene acceso a la sección de mundos y minijuegos y al ránking global, tanto individual como grupal.
- ***Usuario desarrollador***: Tiene acceso a la sección de mundos y minijuegos y al ránking global, tanto individual como grupal. También podrá añadir minijuegos al sistema.
- ***Usuario administrador***: Tiene acceso a la sección de mundos y minijuegos, al ránking global (tanto individual como grupal) y a las opciones de administración de la plataforma.

#### Página *Inicio* para *Jugadores* y *Desarrolladores*

En el caso del inicio para los jugadores y desarrolladores (figura 4.4), la página se compone de los siguientes elementos:

- Un botón de *Start*, que da acceso a la visualización de los distintos mundos.
- Una cabecera situada en la parte superior de la página, que se mantiene visible durante toda la navegación en la aplicación, y contiene el acceso a la página de inicio (es decir, la presente), el acceso al ránking general, el número de estrellas y de puntos obtenidos, y el acceso para cerrar la sesión en la plataforma.

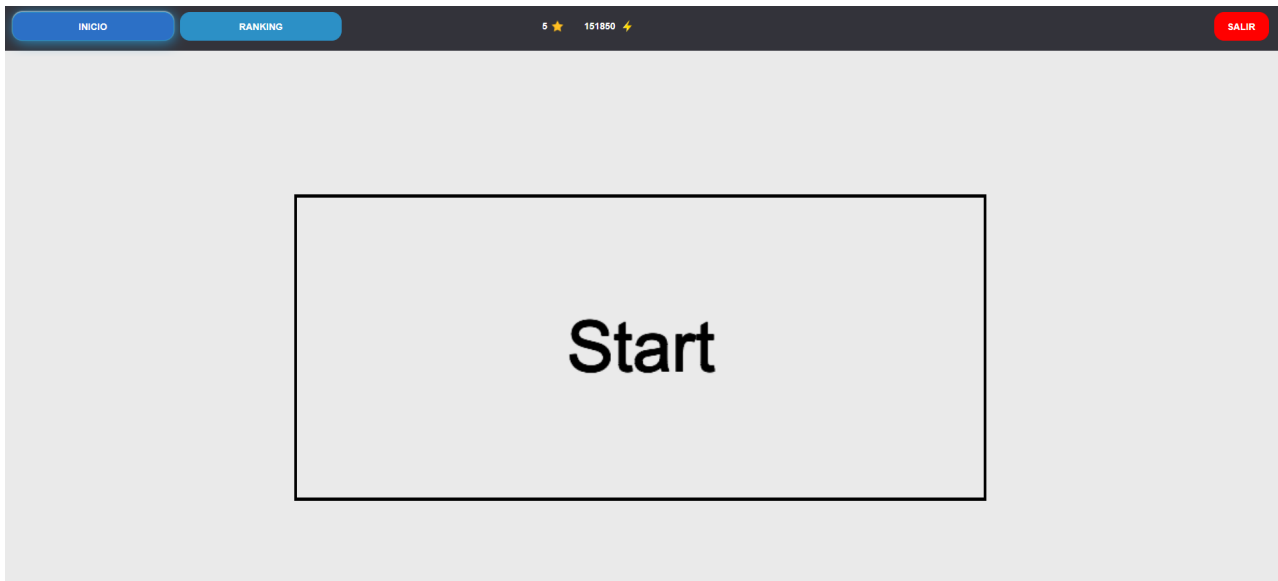


Figura 4.4: Pantalla de inicio para *Jugadores*

### Página *Inicio* para *Administradores*

En este caso, la vista es similar al caso de los usuarios jugadores, incluyendo la opción en la cabecera para acceder al panel administrador (figura 4.5).

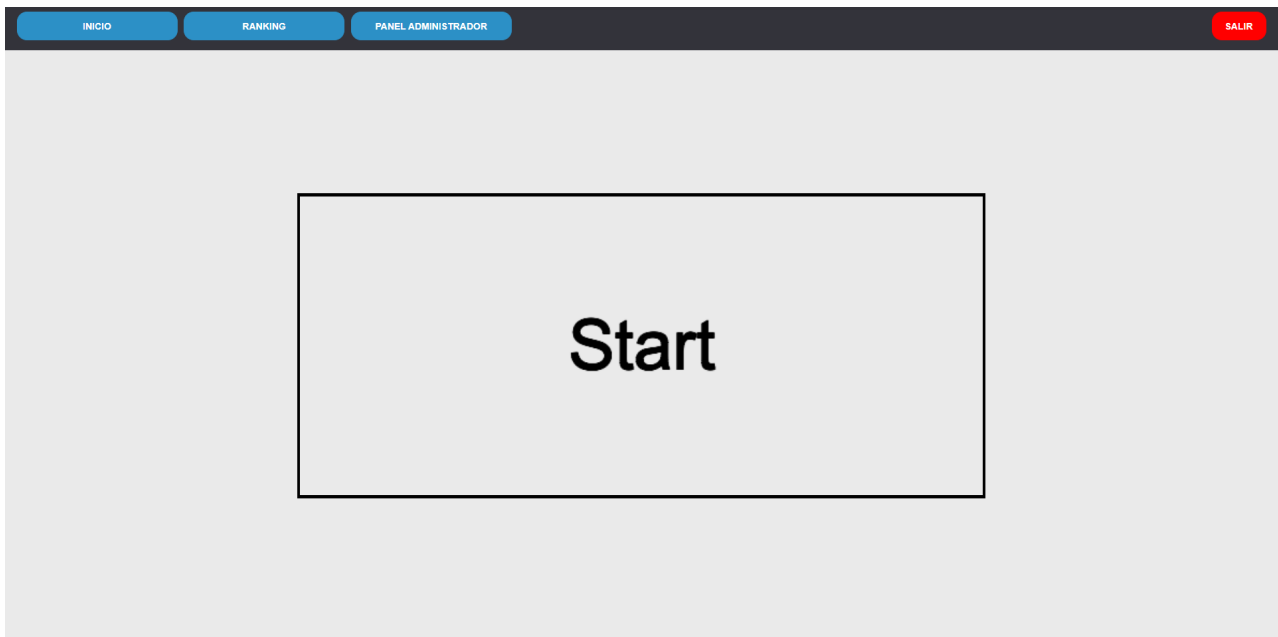


Figura 4.5: Pantalla de inicio para *Administradores*

En ambos casos, al pulsar en *Start*, se accederá a la visualización de los mundos.

### 4.2.3. Página *Mundos*

Tal y como se aprecia en la figura 4.6, esta ventana es una representación del “universo” donde se encuentran los diversos mundos. Pulsando sobre la imagen del mundo correspondiente, se accede a la siguiente página a estudiar, *Minijuegos*.

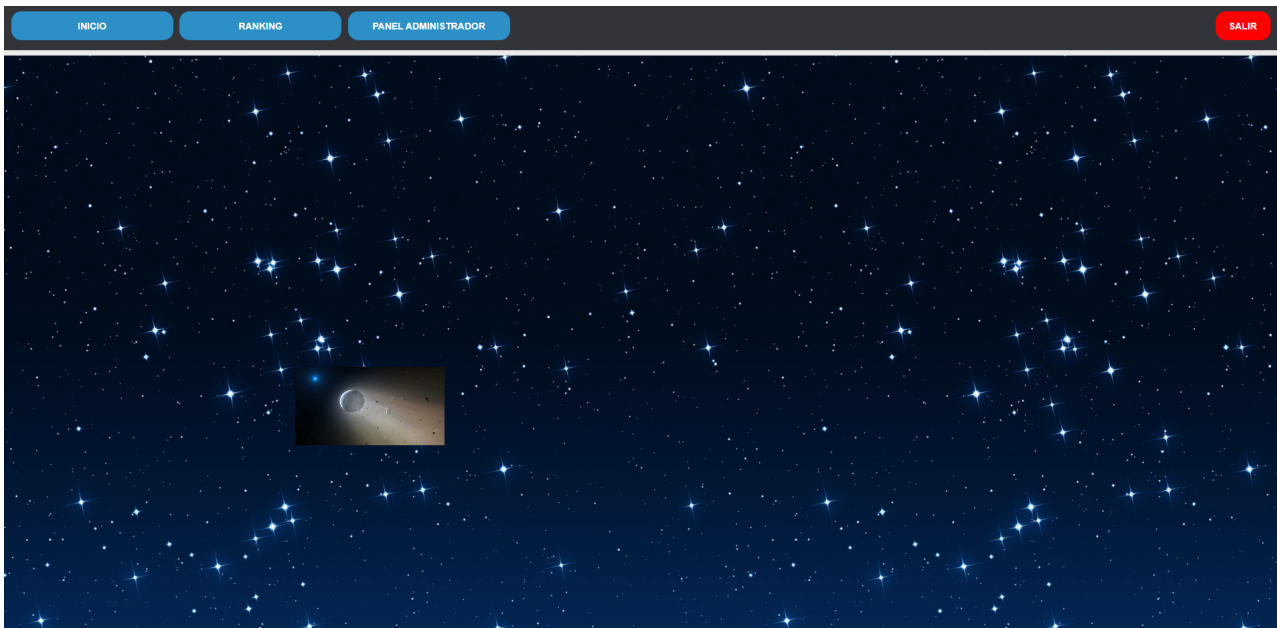


Figura 4.6: Pantalla de *Mundos*

#### 4.2.4. Página *Minijuegos*

Ya dentro de un mundo concreto, la página *Minijuegos* contiene el acceso a cada uno de los minijuegos (a través de su imagen asociada, tal y como presenta la figura 4.7).



Figura 4.7: Pantalla de *Minijuegos*

Además del acceso a estos minijuegos, también contiene un panel superior, con el nombre del mundo, el número de juegos superados, y la puntuación conseguida hasta el momento; y, en la parte inferior izquierda, una flecha para volver a la página de los mundos.

## 4.2.5. Página *Ránking*

La página *Ránking* contiene la clasificación general de los jugadores y de los grupos en el conjunto de juegos. Para navegar entre ambas opciones, la página cuenta con dos etiquetas (*Individual* y *Grupos*) ubicadas en la parte superior (figura 4.8). Esta figura refleja el ránking individual:

Posición	Nombre	Grupo	★	⚡
1	probador2	Pr	7	202800
2	probador1	Pr	5	151850
3	sansimo	Z7	5	40000
4	asitorb	Z7	5	40000
5	davmore	Z1	5	40000
6	alecugn	Z1	5	40000
7	jordela	Z8	5	40000

Figura 4.8: Pantalla de *Ránking individual*

Esta clasificación se presenta en forma de tabla, en la que se recoge:

- **Posición:** Clasificación general del jugador. Para las 3 primeras posiciones, las filas correspondientes se pintan de amarillo, gris y marrón, simulando la correspondencia oro-plata-bronce. Además, esta posición se representa con la imagen del metal correspondiente.
- **Nombre:** Nombre de usuario del jugador.
- **Grupo:** Grupo de pertenencia del jugador.
- **Estrellas:** Número total de estrellas conseguidas por el jugador. Tal y como se aprecia en la figura 4.8, el nombre de la columna se representa con la imagen de un dibujo de una estrella.
- **Puntos:** Representados por el dibujo de un rayo, esta columna almacena la suma total de puntos obtenidos por el jugador.

Por otro lado, el ránking grupal (figura 4.9) cuenta con los siguientes campos:

- **Posición:** Clasificación general del grupo.
- **Grupo:** Nombre del grupo.
- **Estrellas:** Número total de estrellas conseguidas por todos los miembros el grupo.

- **Puntos:** Suma total de los puntos de todos los integrantes del grupo en todos los mini-juegos.

Posicion	Grupo	★	⚡
1	Z1	89	612930
2	Z7	63	456080
3	Z8	39	278460
4	Z10	26	194860
5	Z2	25	178460
6	Z6	22	166195
7	Z4	14	107820

Figura 4.9: Pantalla de *Ránking grupal*

Al tratarse de un ránking, esta lista muestra únicamente (tanto en el caso individual como en el grupal) las 10 primeras posiciones de la clasificación.

#### 4.2.6. Página *Panel Administrador*

La sección *Panel Administrador* contiene todas las opciones de administración de la plataforma. Está estructurada en forma de barra lateral, situada en la parte izquierda de la pantalla, permitiendo así la navegación entre pestañas.

##### *Pestaña Mundos y minijuegos*

La pestaña *Mundos y minijuegos* (figura 4.10) contiene las opciones para la gestión de los mundos, y sus minijuegos asociados. Para cada mundo, se visualiza (de izquierda a derecha):

- El nombre del mundo.
- La imagen que identifica al mundo.
- La imagen dentro del mundo (que contiene el conjunto de minijuegos pertenecientes a ese mundo).
- La opción de cerrar o abrir el mundo, representada con el icono de un candado.
- La opción de ocultar o hacer visible el mundo, que se representa con el icono de un ojo.
- La opción de modificar el mundo, representada con el icono de un lápiz.

- La opción de eliminar el mundo, que se representa con el icono de una cruz roja.

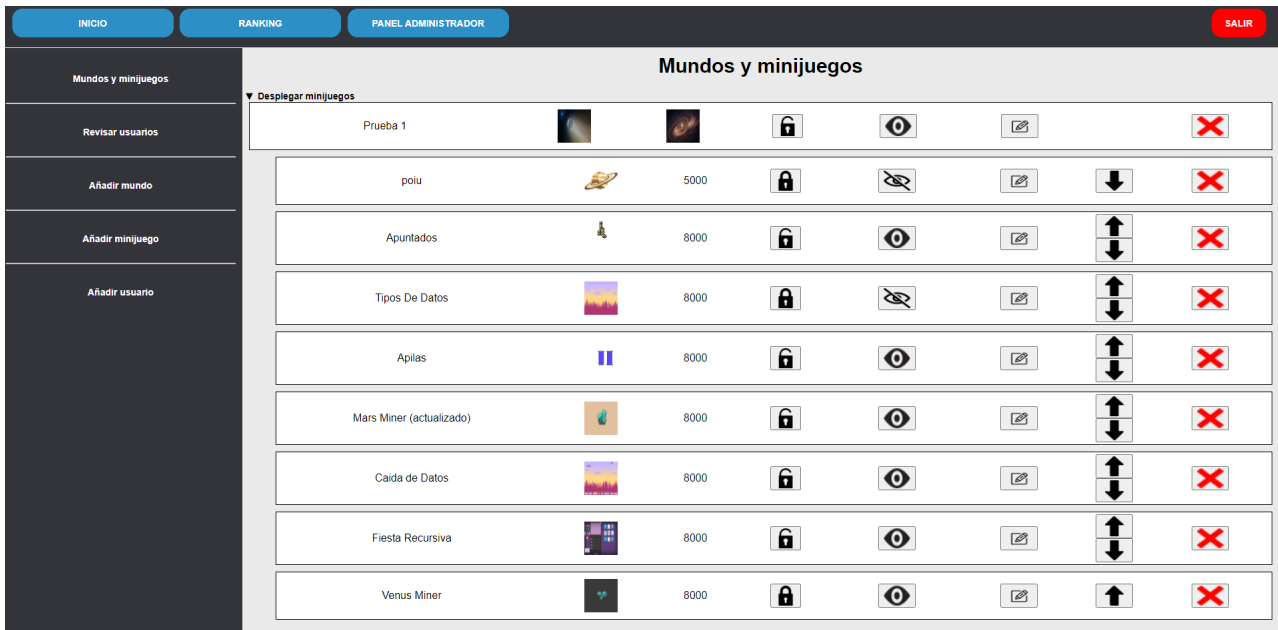


Figura 4.10: Pestaña *Mundos y minijuegos*

Desplegando la lista de minijuegos de cada mundo, se puede encontrar:

- El nombre del minijuego.
- La imagen del minijuego.
- La puntuación máxima del minijuego.
- La opción de cerrar o abrir el minijuego.
- La opción de ocultar o hacer visible el minijuego.
- La opción de modificar el minijuego.
- La opción de subir o bajar el minijuego dentro de la lista (representada con flechas ascendentes y descendentes).
- La opción de eliminar el minijuego.

Para todas estas opciones (tanto en el caso de los mundos como en el de los minijuegos), se dispone de un evento de ratón para el cual, al posicionarlo sobre cada opción, aparezca una etiqueta con la opción correspondiente, facilitando así la usabilidad de la aplicación.

### Pestaña *Revisar usuarios*

La presente pestaña (figura 4.11) está destinada a visualizar la lista de usuarios que componen la plataforma. Además, se permite modificar los datos del usuario, o eliminarlo de la base de datos.



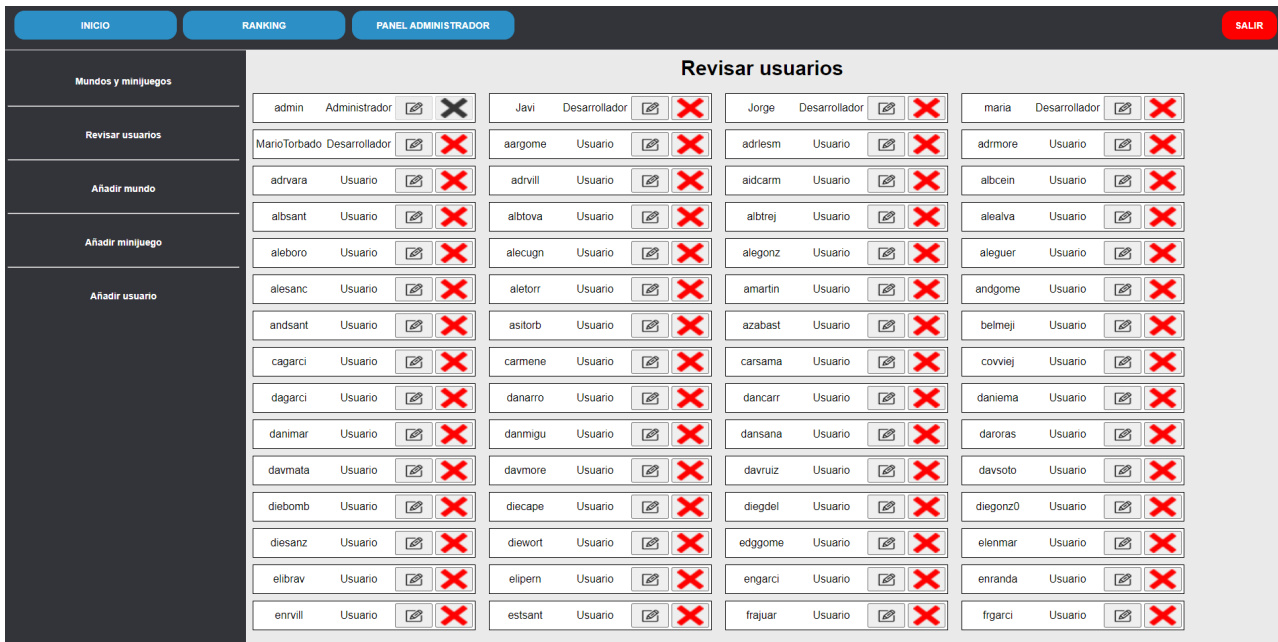


Figura 4.11: Pestaña *Revisar usuarios*

Al final de la lista de usuarios, se dispone de 2 operaciones adicionales: eliminar todos los usuarios y eliminar todos los desarrolladores (4.12).

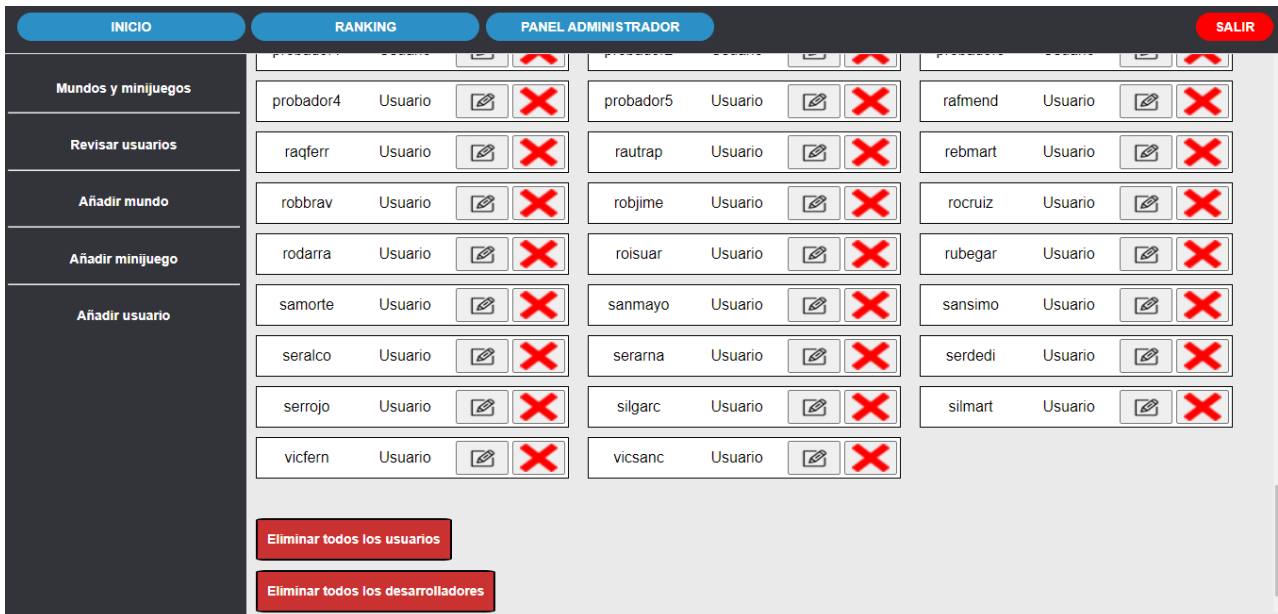


Figura 4.12: Fin de la pestaña *Revisar usuarios*

En el caso de desear modificar los datos de un usuario, las opciones se reflejan en la figura 4.13:

The screenshot shows the 'Modificar usuario' interface. At the top, there are navigation tabs: 'INICIO', 'RANKING', 'PANEL ADMINISTRADOR', and a red 'SALIR' button. On the left, a sidebar contains menu items: 'Mundos y minijuegos', 'Revisar usuarios', 'Añadir mundo', 'Añadir minijuego', and 'Añadir usuario'. The main content area is titled 'Modificar usuario' and contains the following form elements:
 

- 'Nombre': A text input field with the value 'probador2'.
- 'Nueva contraseña': An empty text input field.
- 'Nuevo rol de usuario': A dropdown menu with 'Usuario' selected.
- 'Nuevo grupo': A text input field with the value 'Pr'.
- 'Modificar': A button at the bottom of the form.

Figura 4.13: Interfaz *Modificar usuarios*

Para este usuario, es posible:

- Actualizar su contraseña de acceso a la plataforma.
- Cambiar su rol.
- Cambiar el grupo al que pertenece.

Pinchando en el botón *Modificar*, estos cambios se hacen efectivos.

### Pestaña *Añadir mundo*

Con esta pestaña, reflejada en la figura 4.14, es posible añadir un nuevo mundo a la base de datos y, por consiguiente, a la plataforma.

The screenshot shows the 'Añadir nuevo mundo' interface. It features the same navigation and sidebar as Figure 4.13. The main content area is titled 'Añadir nuevo mundo' and contains the following form elements:
 

- 'Nombre': A text input field.
- 'Imagen del mundo': A file selection area with a 'Seleccionar archivo' button and a 'Ninguno archivo selec.' label.
- 'Imagen de fondo': A file selection area with a 'Seleccionar archivo' button and a 'Ninguno archivo selec.' label.
- 'Añadir': A button at the bottom of the form.

Figura 4.14: Pestaña *Añadir Mundo*

Para añadir un nuevo mundo, basta con incluir esta información:

- El nombre del mundo.
- La imagen con la que identificarlo.
- La imagen de fondo de este mundo.

Al pulsar en *Añadir*, estos cambios se hacen efectivos y se añade el mundo en cuestión.

## Pestaña *Añadir minijuego*

En este caso, la pestaña está dedicada a la inclusión de nuevos minijuegos en el sistema (figura 4.15). Así, los elementos que la componen son los siguientes:

- Un selector del mundo al que pertenecerá el minijuego.
- Un campo para incluir el nombre del minijuego.
- Una imagen que lo identifique visualmente.
- Un campo para incluir su puntuación máxima.



The screenshot shows a web interface for an administrator. At the top, there is a navigation bar with three tabs: 'INICIO', 'RANKING', and 'PANEL ADMINISTRADOR'. A red 'SALIR' button is located in the top right corner. On the left side, there is a vertical sidebar menu with five items: 'Mundos y minijuegos', 'Revisar usuarios', 'Añadir mundo', 'Añadir minijuego', and 'Añadir usuario'. The main content area is titled 'Añadir minijuego' and contains a form with the following elements: a dropdown menu labeled 'Seleccione un mundo', a text input field for 'Nombre:', a file selection button labeled 'Seleccionar archivo' and a status 'Ninguno archivo selec.', a text input field for 'Puntuación máxima:', and a large 'Añadir' button at the bottom.

Figura 4.15: Pestaña *Añadir Minijuego*

Al igual que en la pestaña anterior, los cambios se hacen efectivos al pulsar en *Añadir*.

## Pestaña *Añadir usuario*

Por último, la pestaña *Añadir usuario* permite añadir un usuario concreto, o bien un conjunto de usuarios. Tal y como se aprecia en la figura 4.16, para incluir un nuevo usuario, es necesario:

- Añadir su nombre de usuario.
- Añadir su contraseña de acceso al sistema.
- Seleccionar su rol en la plataforma.
- Incluir el grupo al que pertenece.

En el caso de disponer de una lista ya preparada en el formato correcto, basta con añadir el fichero para incluir a estos usuarios, haciéndolo efectivo al pulsar en *Añadir*.

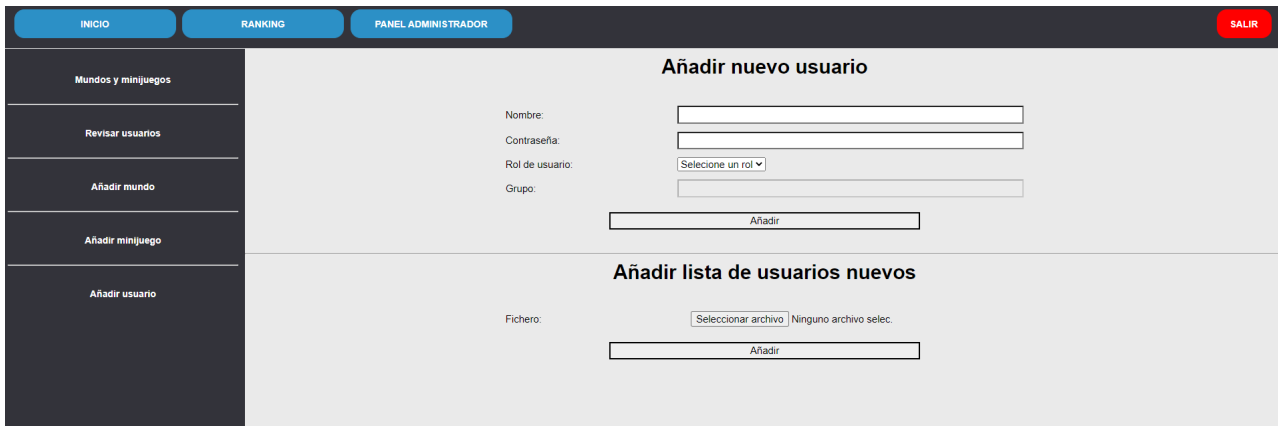


Figura 4.16: Pestaña *Añadir Usuario*

### 4.3. Conjunto de datos

Con el fin de mantener la información estructurada en el sistema, esta se almacenó en una base de datos *MySQL* sincronizada con la plataforma de trabajo. La disposición de esta información se representa en la figura 4.17.

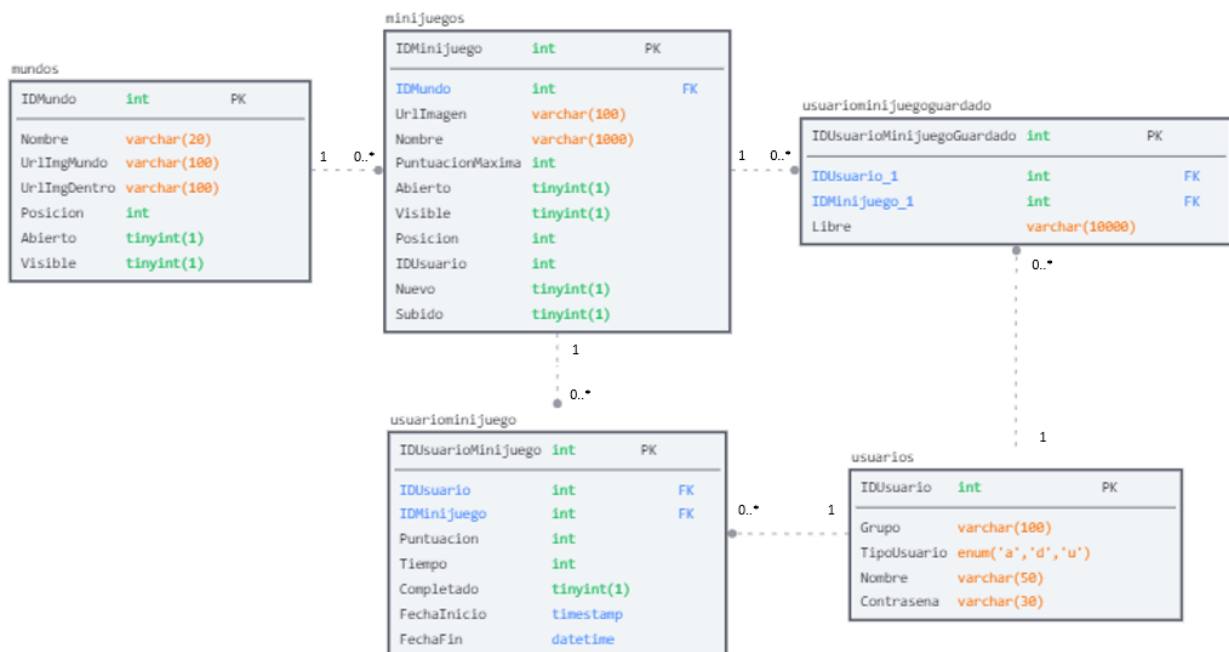


Figura 4.17: Modelo de datos del sistema

#### Tabla *mundos*

Esta se encarga de almacenar la información relativa a un mundo concreto. Los atributos que almacena serán:

- **IDMundo:** Actúa como clave primaria de la tabla, siendo el identificador único de cada instancia.

- **Nombre:** Representa el nombre proporcionado para el mundo.
- **UrlImgMundo:** Almacena la ruta relativa a la imagen con la que se identificará el mundo dentro del sistema.
- **UrlImgDentro:** Al igual que el atributo anterior, guarda información sobre una imagen, pero esta vez haciendo referencia a la imagen de dentro del mundo, que se usa como fondo para presentar los minijuegos disponibles en ese mundo concreto.
- **Posición:** Representa la posición del mundo.
- **Abierto:** Con un valor lógico, este atributo indica si el mundo se encuentra en modo *abierto* o *cerrado*.
- **Visible:** Guarda el estado de visibilidad del mundo (*visible* o *no visible*).

### Tabla *minijuegos*

La tabla *minijuegos* es la encargada de almacenar la información relativa a los minijuegos disponibles en el sistema. Los atributos que la componen son:

- **IDMinijuego:** Es el identificador único (en forma de número entero) del minijuego, y actúa como clave primaria del mismo.
- **URLImagen:** Contiene una cadena con la dirección relativa de la imagen que representa al minijuego en el sistema.
- **Nombre:** Almacena el nombre del minijuego.
- **PuntuacionMaxima:** Este campo representa el valor máximo que se puede alcanzar jugando al minijuego en cuestión.
- **Abierto:** Es un indicador de que el juego está abierto o cerrado, en formato lógico.
- **Posición:** Posición en la que se encuentra el minijuego.
- **IDUsuario:** Identificador del usuario que añadió el juego a la plataforma.
- **Nuevo:** Indicador de la novedad del minijuego.
- **Este último atributo indica si el juego se ha subido o no a la plataforma**

### Tabla *usuarios*

Esta tabla guarda, como su propio nombre indica, la información relativa a los usuarios del sistema. Dentro del conjunto de datos que la forman, se encuentran:

- **IDUsuario:** Valor en formato entero que actúa como clave primaria de la tabla, identificando de forma unívoca a cada usuario del sistema.
- **Grupo:** Cadena de texto que almacena el nombre del grupo al que pertenece el usuario en cuestión.
- **TipoUsuario:** Enumeración que contiene los valores *A* (Administradores), *D* (Desarrolladores) y *U* (Usuarios), correspondientes al rol del usuario en el sistema.

- **Nombre:** Cadena de texto que representa su nombre de usuario.
- **Contraseña:** Es la cadena de texto que almacena la contraseña del correspondiente usuario del sistema.

### Tabla *usuariominijuego*

Los registros de actividad en los minijuegos por parte de los distintos usuarios se almacenan en esta la tabla. Para ello, contiene los siguientes atributos:

- **IDUsuarioMinijuego:** Número entero que identifica de forma unívoca (actuando como clave primaria) cada partida de un usuario concreto en un minijuego determinado.
- **IDUsuario:** Clave foránea (referenciando a la tabla *usuarios*) que identifica el usuario que ha jugado la partida.
- **IDMinijuego:** Clave foránea (referenciando a la tabla *minijuegos*) que identifica al minijuego.
- **Puntuacion:** Puntuación obtenida en esa partida (representado por un número de tipo entero).
- **Tiempo:** Tiempo (en segundos) que ha durado la partida.
- **Completado:** Indicador binario de si el usuario ha superado o no el nivel al que ha jugado.
- **FechaInicio:** Momento (fecha y hora) de entrada al minijuego.
- **FechaFin:** Momento (fecha y hora) de salida del minijuego.

### Tabla *usuariominijuegoguardado*

Por último, esta tabla mantiene un registro de las entradas a las diferentes puntuaciones obtenidas en cada uno de los niveles de cada partida jugada en cada minijuegos.

- **IDUsuarioMinijuegoGuardado:** Identificador único (clave primaria) de cada una de las entradas de la tabla.
- **IDUsuario:** Clave foránea (referenciando a la tabla *usuarios*) que identifica al usuario que ha jugado la partida.
- **IDMinijuego:** Clave foránea (referenciando a la tabla *minijuegos*) que identifica al minijuego.
- **Libre:** Array de valores enteros que indica, para un usuario concreto en un juego determinado, las puntuaciones obtenidas para cada uno de los niveles del juego.

# Capítulo 5

## Resultados

### 5.1. Eliminación de determinadas dependencias con *Unity*

La primera modificación de mejora que se planteó con respecto a la plataforma inicial fue eliminar determinadas dependencias con el *software Unity* que con el tiempo complicarían el desarrollo de nuevos minijuegos y funcionalidades. Más concretamente, estas dependencias estaban relacionadas con la forma de acceder al código de un minijuego.

#### 5.1.1. Método de llamada antes de la modificación

Antes de llevar a cabo las modificaciones que se expondrán más adelante, el sistema accedía al código de cada minijuego (para así cargarlo y permitir al usuario jugar) a través del fichero *games.php*. Para ello, se hacía uso de la siguiente orden:

```
var unityInstanceGame = UnityLoader.instantiate("unityContainer2",  
    minijuegos/<?php echo $IDMinijuego; ?>/Build/<?php echo  
    $IDMinijuego; ?>.json", {onProgress: UnityProgress});
```

Estos minijuegos se cargaban a partir de la orden *UnityLoader.instantiate()*, recibiendo entre los parámetros necesarios un archivo con formato *.json* con el identificador del juego, ubicado en el subdirectorio dedicado a dicho juego, en la carpeta *Build*.

#### 5.1.2. Método de llamada después de la modificación

Con el fin de unificar, estandarizar y facilitar el método de acceso a los juegos, se decidió cambiar la metodología antigua por lo siguiente. En lugar de cargar el fichero *.json* del juego, bastará con cargar el *index.html* del juego en cuestión, método más estándar para acceder a aplicaciones web.

Esta llamada se hace desde el fichero *inicio.php*, de la siguiente forma:

```
document.getElementById("game").src = "./minijuegos/" + IDMinijuego + "/  
index.html";
```

Así, el acceso a los minijuegos en cuestión se simplifica y unifica, ayudando al futuro crecimiento de la plataforma.

Por otro lado, se planteó la estandarización de la interface entre el juego y la plataforma. Para ello se diseñó y construyó un API (Application Program Interface) de forma que se trasladaron las funciones ubicadas en el fichero *games.php* (que establecían comunicación entre el juego y la plataforma) a una nueva librería *Javascript*. Para ello, se creó un nuevo fichero, *IntegrationApi.js*, que actúa como método de comunicación entre la plataforma y los minijuegos y alberga todas estas funcionalidades. Agrupando la funcionalidad y posibilitando su acceso con la API.

## Diseño del consumo del *API* de la Plataforma

Dado que la plataforma expone un API para ser consumida, se procedió a diseñar e implementar una librería de funciones javascript que hicieran un consumo seguro de las mismas.

Tras analizar las APIs expuestas se vio que la madurez de las mismas, en base al modelo de madurez de Richardson estaba encuadrado en el Level 2, es decir API con URI de consumo y verbos asociados a la operación a realizar [8].

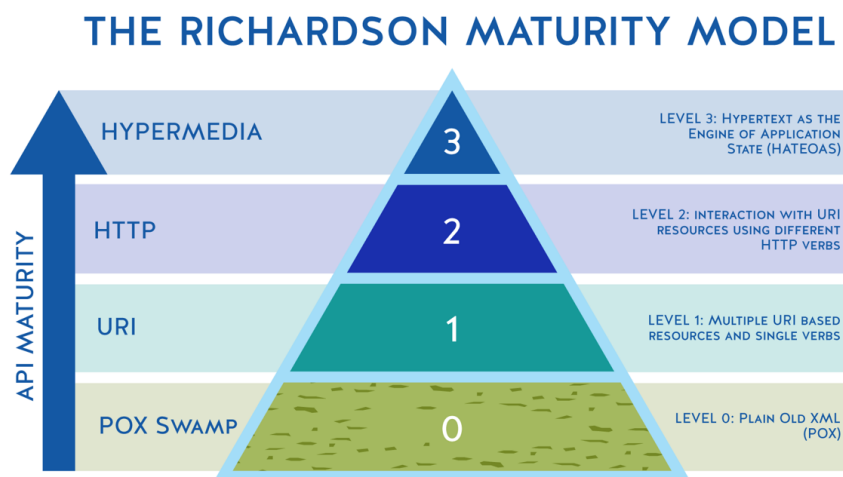


Figura 5.1: Modelo de Madurez de Richardson

Las APIs usadas usan el método REST POST para grabar contenido que se envía en formato estandarizado JSON y como caso excepcional justificado, se usa también para la lectura de datos cuando se requiere enviar datos vía body.

## Implementación de la librería Javascript: Fichero *IntegrationApi.js*

El fichero *IntegrationApi.js* contiene la definición de las funciones que comunican cada juego con la plataforma. Estas funciones son:

- ***exit()***: Llamada al salir del juego, para regresar al controlador.
- ***save\_update(data)***: Función invocada cada vez que se necesite actualizar los datos del juego (puntuación, y completado o no completado) con un *Array* de 2 valores enteros:



- La puntuación obtenida en el juego.
- Un valor 1 o 0. que indicará si el juego ha sido completado o no, respectivamente.

Estos datos vienen dados en el parámetro de función *data*. Se hace uso de la operación POST /

- ***getOpenField()***: Función invocada para obtener el campo valor libre de la base de datos, donde el jugador guardará cualquier información que considere referente al estado del juego para el jugador.
- ***setOpenField(openField)***: Sirve para almacenar la información en el campo de valor libre, referente al desarrollo del juego para el jugador.

## Nueva estructura del sistema

Al añadir este API de comunicación al sistema, la nueva estructura de la plataforma se presenta en la figura 5.2.

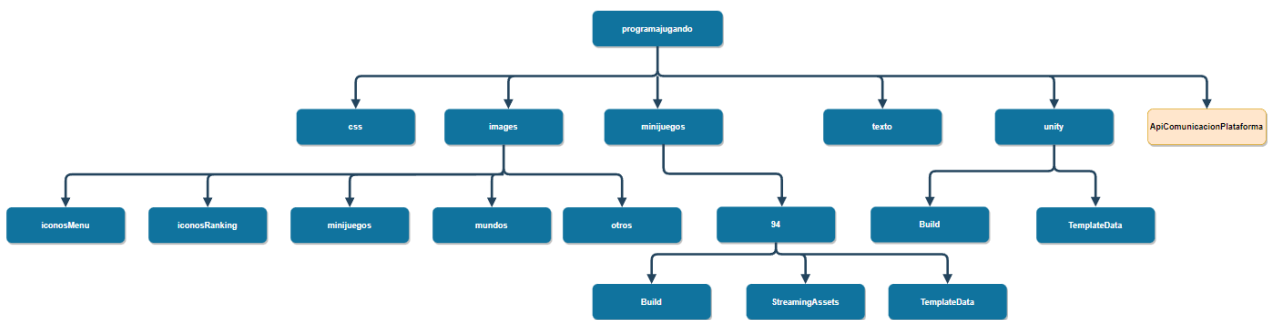


Figura 5.2: Estructura de la plataforma tras añadir el patrón fachada

Tal y como se aprecia en la figura 5.2, a la estructura de directorios existentes, se añadió el directorio *ApiComunicacionPlataforma*, que albergará el fichero librería *IntegrationApi.js* (5.2.1).

## Cambios a realizar por parte de los desarrolladores de minijuegos

Por ello, para hacer efectivos estos cambios y evitar problemas futuros, los desarrolladores que quieran incluir un nuevo minijuego deben hacer lo siguiente:

- Crear una aplicación *web*, cuyo fichero de entrada se llame *index.html*.
- Añadir, en la cabecera del fichero *index.html*, el API de comunicación entre el juego y la plataforma, a partir del siguiente comando:

```
<script src="../../ApiComunicacionPlataforma/IntegrationApi.js"></script>
```

- Llamar a las funciones incluidas en *IntegrationApi.js* desde el fichero *index.html*, cada vez que se quiera comunicar al juego con la plataforma y viceversa, haciendo uso de los comandos correspondientes.

Incluyendo estos cambios en sus minijuegos, esta modificación se hizo efectiva, cumpliendo así uno de los primeros propósitos del *TFG*.

## 5.2. Análisis y corrección de brechas de seguridad en el sistema actual

Con el fin de reforzar la seguridad de la plataforma, se hizo un estudio de los posibles fallos de seguridad del sistema. Y, por cada vulnerabilidad encontrada, se propuso un cambio para solventarla. Se localizaron dos vulnerabilidades críticas.

### 5.2.1. Ataque a la base de datos desde software foráneo

La primera vulnerabilidad que se detectó hacía referencia a la seguridad del acceso y modificación de la base de datos del sistema. A través de las *Herramientas para desarrolladores* del navegador (Mozilla Firefox en nuestro caso, ya que fue el navegador utilizado a lo largo del desarrollo del proyecto), es posible inspeccionar parte del código que se ejecuta, así como visualizar las diferentes peticiones que se realizan a la plataforma.

Indagando en estas opciones para desarrolladores, se encontró que, dentro de la pestaña *Depurador* (figura 5.3, flecha 1), era posible acceder, navegando por el panel izquierdo al fichero *games.php* (flecha 2), que contenía llamadas directas a otros ficheros que alteraban el contenido de la base de datos del sistema. Esta vulnerabilidad vino muy relacionada con la creación del fichero *IntegrationApi.js*, ya que fue detectada durante el desarrollo de la modificación vista en la sección anterior.

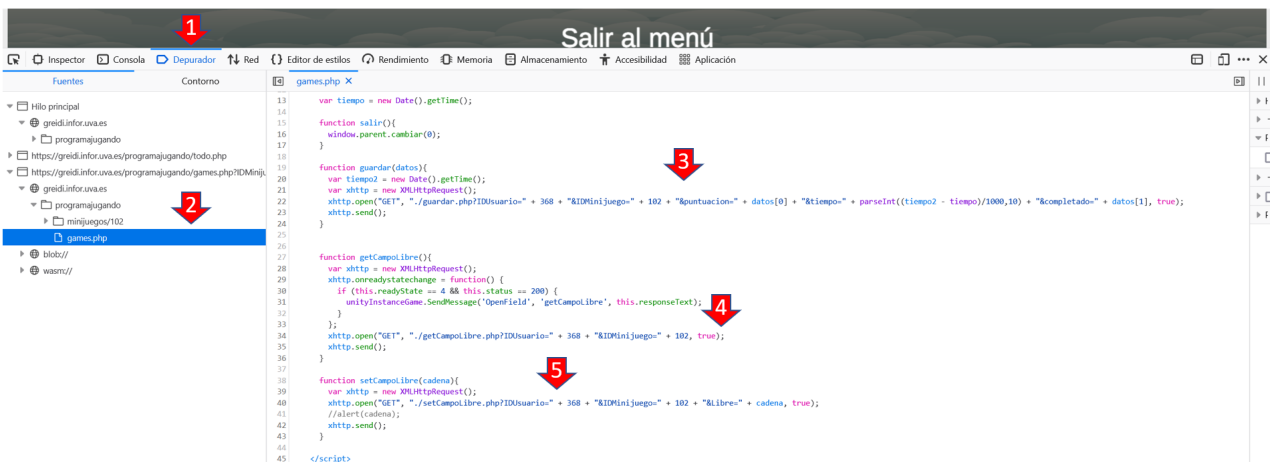


Figura 5.3: Vulnerabilidad en la llamada a *guardar.php* desde el apartado *Depurador*

Más concretamente, estas funciones eran las siguientes:

- **Función *guardar(datos)***: Hará una llamada al fichero *guardar.php*, incluyendo en las cabeceras la puntuación obtenida en un juego determinado o el tiempo que ha tardado, que se pasarán como parámetro a la función (flecha 3).
- **Función *getCampoLibre()***: Invocará al fichero *getCampoLibre.php*, enviando también en sus cabeceras información como el identificador del minijuego o del propio usuario (flecha 4).

- **Función *setCampoLibre()***: Llamará al fichero *setCampoLibre.php*, enviando también en sus cabeceras el identificador del minijuego, el del usuario y el parámetro *libre* que recibirá la función como parámetro (flecha 5).

Así, la comunicación entre los ficheros se refleja en la figura 5.4.

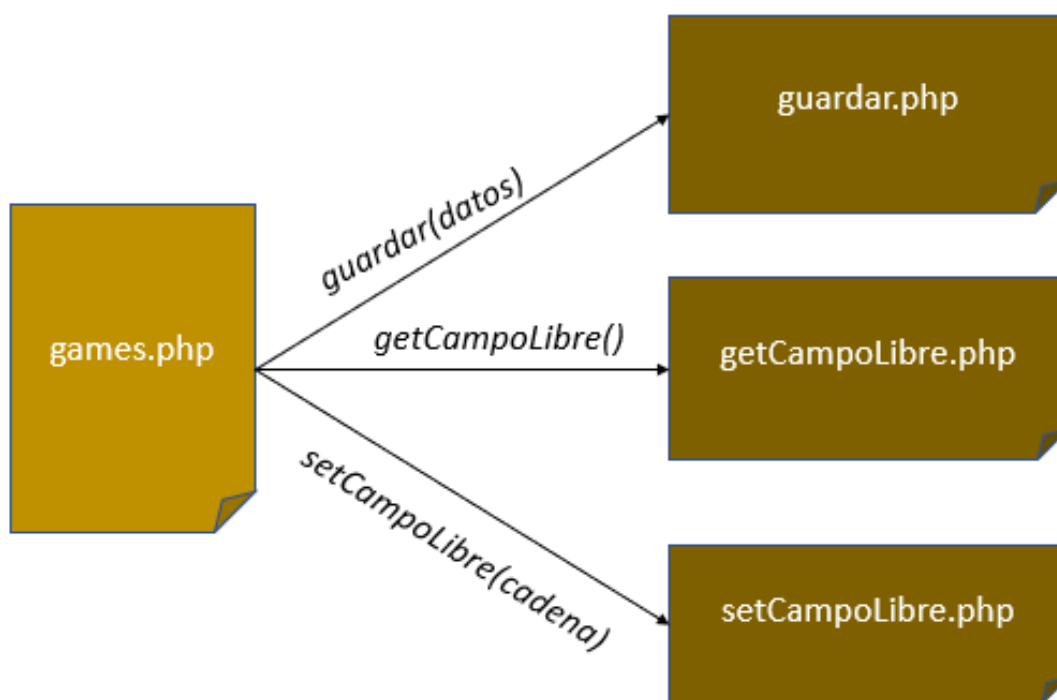


Figura 5.4: Comunicación entre *guardar.php* y otros ficheros

Este agujero de seguridad también se encontraba al entrar en la pestaña *Red*, al inspeccionar la página (figura 5.5, flecha 1), tras haber jugado a un minijuego y haber superado cualquiera de sus niveles. Al entrar en esta pestaña, es posible acceder a las diferentes peticiones enviadas al servidor (flecha 2) y visualizar la petición (flecha 3). Tal y como se aprecia en la figura 5.5, es posible visualizar la petición, mostrándose el fichero (*guardar.php*) y todas las cabeceras asociadas a la petición.



Figura 5.5: Vulnerabilidad en la llamada a *guardar.php*

Esta era una brecha en la seguridad crítica, ya que, haciendo uso de cualquier herramienta para testear APIs de tipo REST, sería posible hacer una llamada de tipo *POST* que introduzca datos erróneos en la base de datos a petición del usuario (figura 5.6).

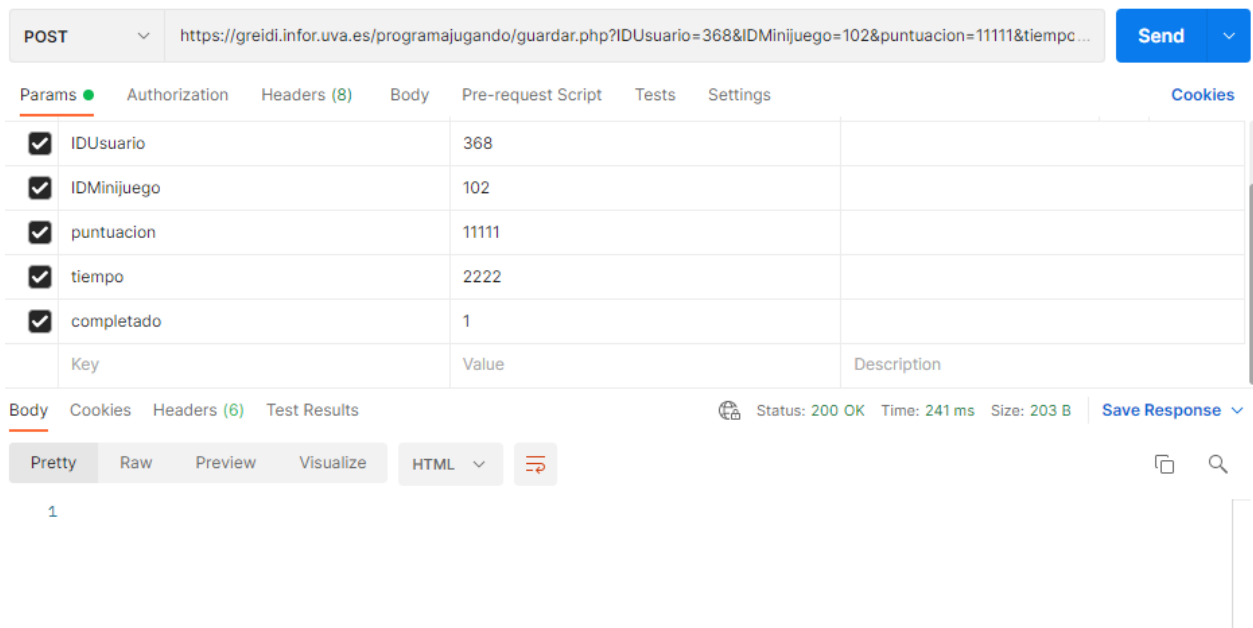


Figura 5.6: Petición de tipo *POST* a *guardar.php* en el servicio *Postman*

Al hacer la petición, el servicio no ofrece ningún tipo de problema, accediendo y modificando la base de datos sin necesidad de haber jugado siquiera al juego en cuestión. Esto ponía en grave peligro la seguridad e integridad de los datos, con lo que fue necesario tomar acciones para solventarlo y hacer seguro el acceso y modificación de los datos del sistema.

## Aplicación del *patrón fachada* como solución propuesta

La solución que se decidió implementar fue la que sigue. Para ocultar la información y llamadas realizadas y conseguir una comunicación cliente-servidor, se creó un fichero PHP que conformase un patrón conocido comúnmente como *patrón fachada* en la estructura del sistema, haciendo las llamadas pertinentes para que la información no sea accesible para un usuario cualquiera.

Para ello, se decidió ocultar la funcionalidad ubicada en el fichero *games.php* (figura 5.3), de forma que, introduciendo llamadas entre ficheros intermedios, se lleven a cabo las mismas operaciones, sin revelar las llamadas realizadas. Tras realizar las modificaciones vistas en la Sección 5.1, estas funcionalidades se encontraban reubicadas en el fichero *IntegrationApi.js*, lo que implicó realizar cambios en el mismo.

La estructura propuesta aparece en la figura 5.7:

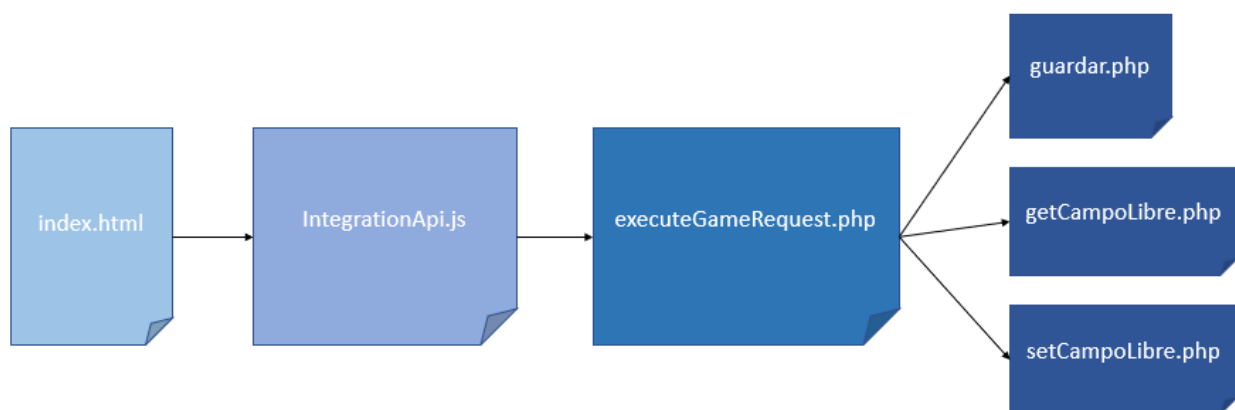


Figura 5.7: Comunicación entre ficheros tras aplicar el *patrón fachada*

Con esta nueva estructura, los ficheros *index.html* de cada minijuego invocan (para cada una de las funcionalidades vistas) las funciones disponibles en el fichero librería *IntegrationApi.js*, que a su vez hacen llamadas al fichero *executeGameRequest.php*, que hace las llamadas correspondientes que alteran y modifican la base de datos.

### Ficheros *index.html* de cada minijuego

Los ficheros *index.html* correspondientes a cada minijuego contendrán funciones específicas que invoquen a las funciones correspondientes del API, *IntegrationApi.js*. Tal y como se comentó en la sección 5.1.2, este debe ser incluido en la cabecera del fichero.

### Fichero *IntegrationApi.js*

El fichero *IntegrationApi.js* fue modificado, de forma que las nuevas comunicaciones con la base de datos se establecen a partir de llamadas al fichero *executeGameRequest.php*, ocultando así las llamadas claves en el sistema.

## Fichero *executeGameRequest.php*

Por último, este será el fichero encargado de actuar como “fachada”. Así, estará estructurado de forma que, en función de la opción que se necesite, hará las llamadas que antes contenía el fichero *games.php*

La figura 5.8 mostrará el efecto de este cambio en la pestaña *Depurador* (flecha 1) de las *Herramientas para desarrolladores*. Ahora, será posible acceder al contenido del API de comunicación (*IntegrationApi.js*), pero no al contenido de *executeGameRequest.php*, que será quien albergue las llamadas que editen el contenido de la base de datos.

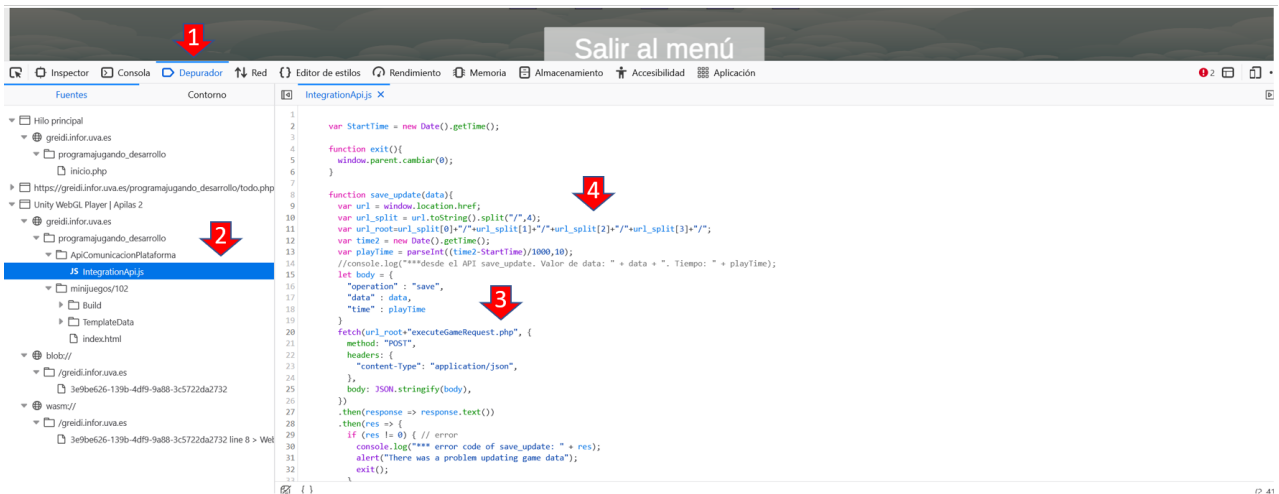


Figura 5.8: Pestaña *Depurador* en las *Herramientas para desarrolladores* tras la aplicación del *patrón fachada*

La flecha 3 indica un ejemplo de petición a este fichero. Se podrá ver cuál es el fichero al que se hace la petición, pero no será posible acceder a la funcionalidad del mismo, complicando así la manipulación indebida de los datos del sistema.

## Comprobación de la dirección de origen de las peticiones

Como medida de seguridad adicional, se decidió aplicar numerosas comprobaciones para verificar el origen de las peticiones *GET* y *POST*, de forma que, si la petición no se realiza desde la plataforma, esta petición será rechazada.

Así, la flecha 4 de la figura 5.8 mostrará como esta comprobación se realiza antes de llegar a hacer ninguna llamada más en el sistema.

### 5.2.2. Réplicas de peticiones *POST* modificando la base de datos

La siguiente vulnerabilidad que se encontró durante el proceso de mejora de la seguridad, fue la posibilidad del reenvío y modificación de algunas llamadas capaces de modificar la base de datos de manera ilícita.

Esta brecha surge tras haber jugado una partida en cualquier juego disponible. Al finalizar el juego, accediendo a las *Herramientas para desarrolladores*, en la pestaña *Red* (figura 5.9,

flecha 1) era posible acceder a una petición de tipo *POST* al fichero *executeGameRequest.php* (flecha 2). Pinchando en esta petición, se puede consultar la petición en la pestaña *Cabeceras* (flecha 3).



Figura 5.9: Vulnerabilidad en el envío de una petición a *executeGameRequest.php*

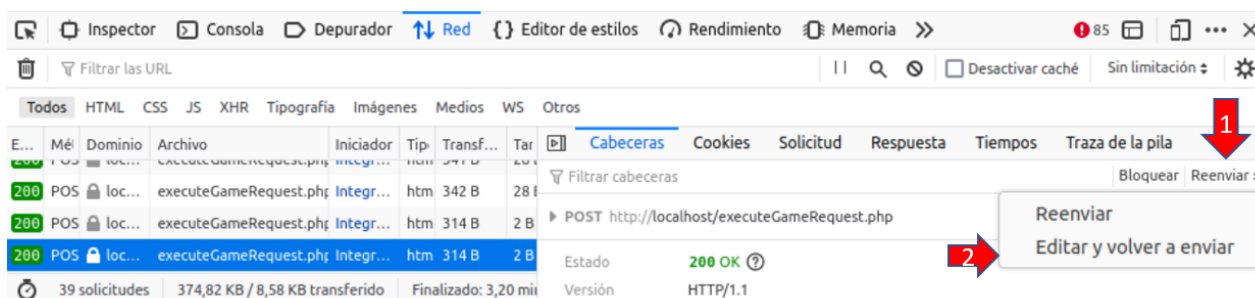


Figura 5.10: Opción de edición y reenvío de una petición

Dentro de esta pestaña, existe una opción que permite la edición y reenvío de una petición existente (figura 5.10, flechas 1 y 2). La figura 5.11 muestra como muestra la dirección que recibe la llamada (flecha 1), y cómo es posible editar el cuerpo de la petición (flecha 2) y enviar de nuevo la petición (flecha 3), siendo visibles datos sensibles del sistema.

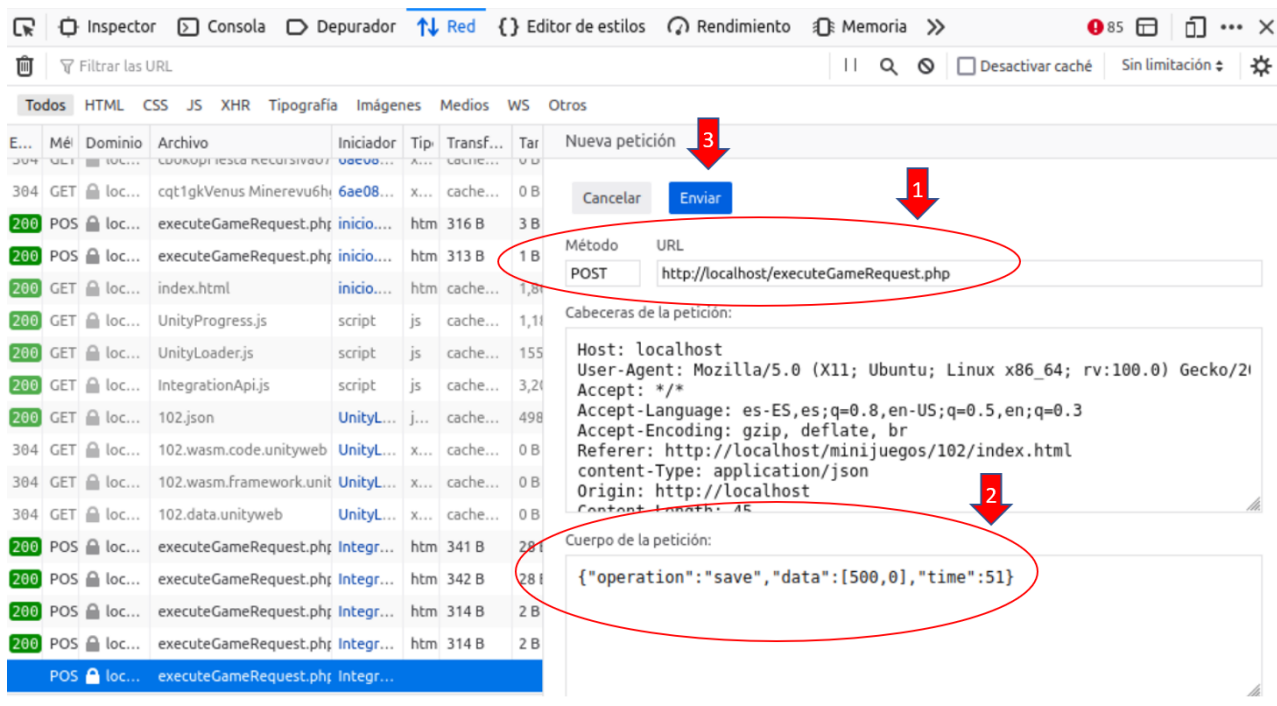


Figura 5.11: Forma de edición y reenvío de la petición

Esto abriría gran cantidad de posibilidades a los atacantes que deseen introducir datos no verídicos de forma ilícita.

### Solución: Encriptación de los datos enviados

La solución propuesta a esta posible vía de entrada de los atacantes, consistió en encriptar aquellos datos que se envían de forma pública, codificando la información y dificultando así la edición de la petición.

Para ello, tras barajar diferentes posibilidades se determinó elegir el algoritmo AES para realizar la encriptación del mensaje en tránsito [9] a través de la librería javascript CryptoJS con clave secreta .

Aunque la opción de reenviar la petición no se elimina, al no permitir la edición de la misma, el sistema recibirá peticiones duplicadas, lo cual no será un problema de alteración de la información.



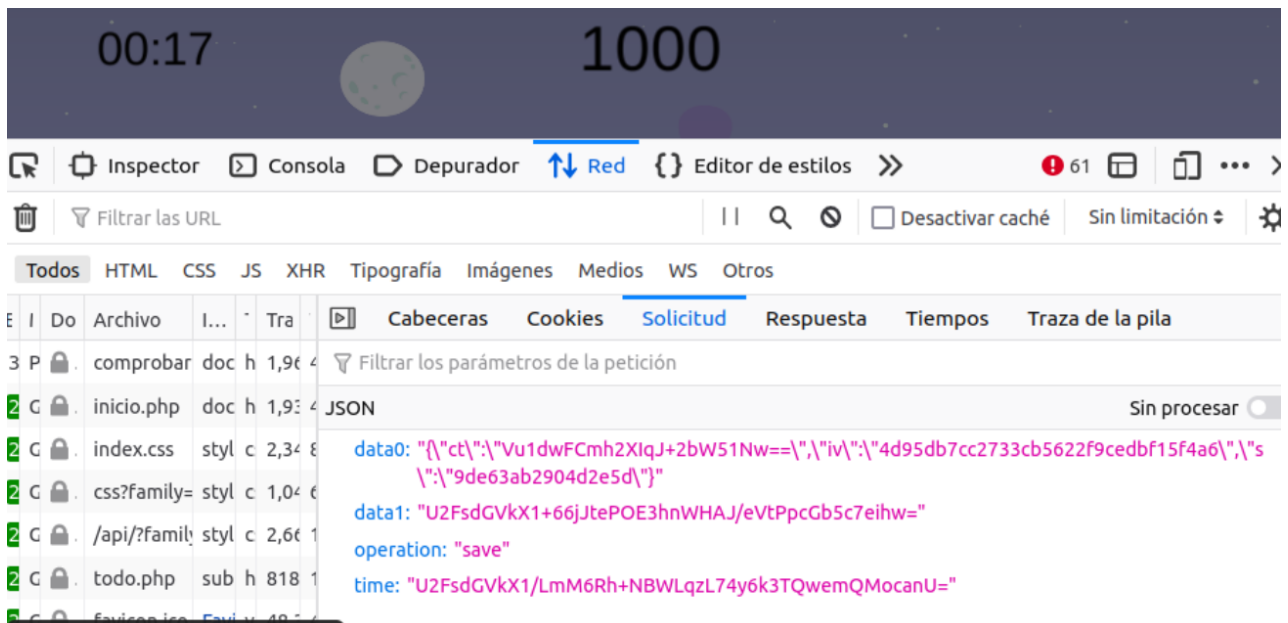


Figura 5.12: Cuerpo de la solicitud tras el encriptamiento de los datos enviados

La figura 5.12 muestra cómo aparecen los datos, tras aplicar dicha librería. Al revisar la petición dentro de la pestaña *Red*, se puede ver como ésta aparece encriptada, lo que no permite la edición directa del texto, y dificulta en gran medida la alteración de esta información.

### Modificaciones necesarias en los ficheros *index.html* de los minijuegos

Fue necesario importar en los ficheros *index.html* de los minijuegos existentes en la plataforma la librería *Javascript* utilizada para encriptar la información. Por ello, se añadió a la cabecera de cada uno de estos ficheros la siguiente orden:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.0.0/crypto-js.min.js"></script>
```

De esta forma, se importa la librería *crypto-js*, utilizada para encriptar y desencriptar en los ficheros *Javascript*.

### Modificaciones necesarias en el fichero *IntegrationApi.js*

Para adaptar el API a los cambios necesarios para la encriptación de la información, se utilizaron funciones pertenecientes a la librería *CryptoJS* de *Javascript*, para encriptar y desencriptar la información. De esta forma, las modificaciones en las funciones serán los que siguen.

- ***save\_update(data)***: Se encriptarán los valores recibidos en el parámetro *data*, de forma que estos se envíen encriptados en el *body* de la petición. El tratamiento de esta información será de vital importancia, ya que será el que posteriormente se introducirá en la base de datos.
- ***getOpenField()***: Se desencriptarán los valores recibidos de *executeGameRequest.php*, los cuales habrán sido previamente encriptados. En este caso, no se modifica la base de datos.

- ***setOpenField(openField)***: Al igual que en la primera función, los datos serán encriptados antes de ser enviados a *executeGameRequest.php*, ya que en este caso se envía información para modificar la base de datos.

Como medida de seguridad, la clave se genera de forma aleatoria en cada paso de mensajes, lo que se replica de forma análoga en el fichero *executeGameRequest.php* para establecer una comunicación coherente.

### Modificaciones necesarias en el fichero *executeGameRequest.php*

En este caso, se utilizó el kit de herramientas de OpenSSL como medida de seguridad de la comunicación en la red. Para ello, se usaron las funciones *openssl\_encrypt()* y *openssl\_decrypt()* para encriptar y desencriptar de forma correcta los mensajes enviados, y así poder comunicarse correctamente con la base de datos de la plataforma.

De forma análoga al fichero *IntegrationApi.js*, los datos serán encriptados y desencriptados según la opción indicada en cada caso.

## 5.3. Inclusión del apartado *Ránking por juego* en la plataforma

Una de las distintas mejoras que se solicitó que se implantaran en la plataforma fue la adición de una sección para visualizar la clasificación de los usuarios y los grupos, haciendo un filtrado previo por juego. Gracias a esta nueva funcionalidad, la investigación docente se vuelve más sencilla y eficaz.

### 5.3.1. *Ránking por juego* para los usuarios administradores

Aprovechando la estructura y las funcionalidades disponibles para los administradores de la plataforma, se añadió una nueva sección dentro del *Panel Administrador*, que se denominó *Ver clasificación* (figura 5.13).

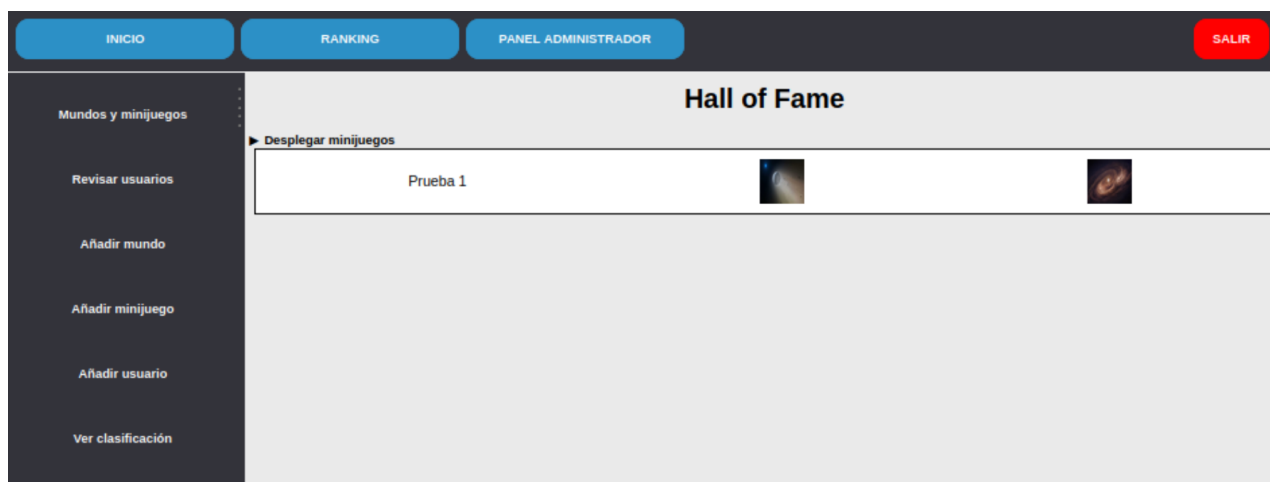


Figura 5.13: Nueva pestaña de *Ver clasificación* en el Panel Administrador

Esta pestaña contiene una lista de mundos, que a su vez contienen una lista con los minijuegos que lo componen, respetando el estilo usado en el resto de la plataforma hasta el momento. Al desplegar la lista de minijuegos, la interfaz está compuesta por los elementos de la figura 5.14.

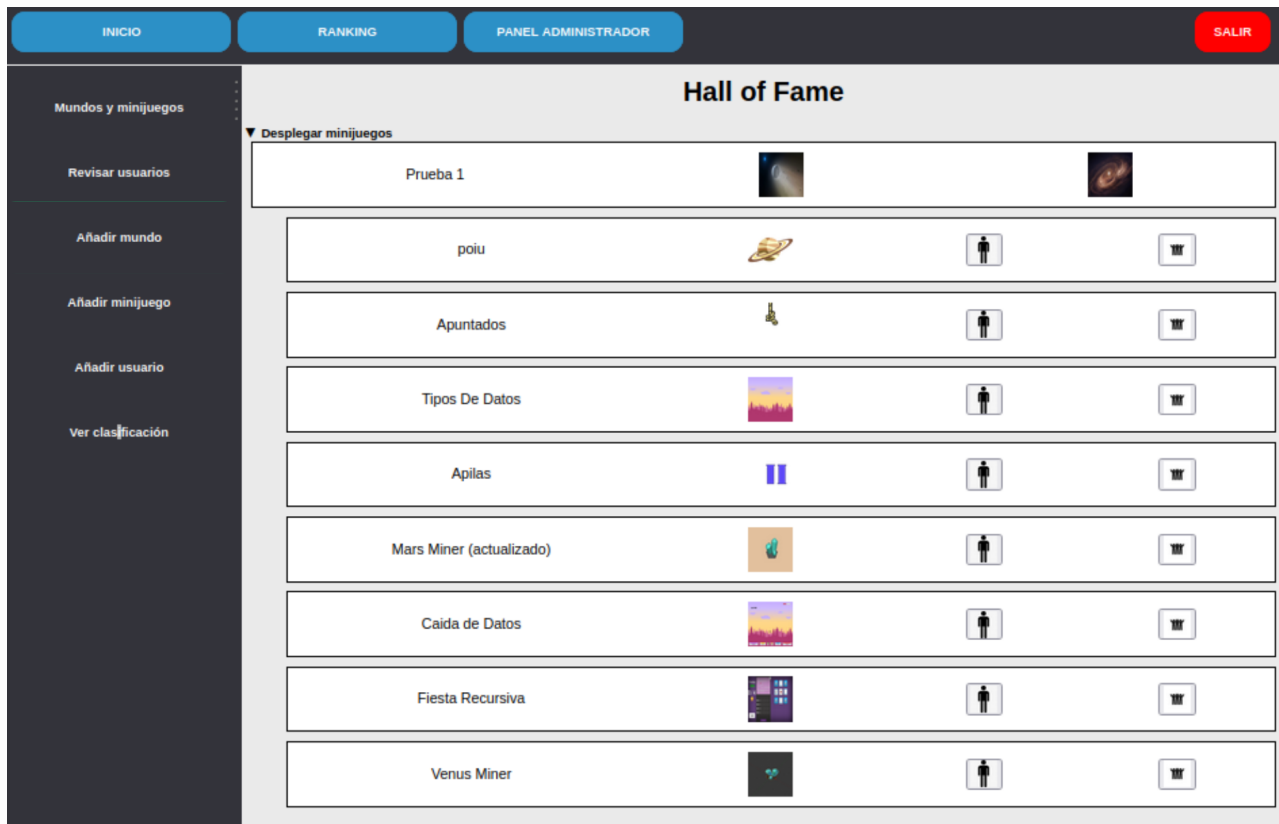


Figura 5.14: Nueva pestaña de *Ver clasificación* en el Panel Administrador

Para cada minijuego, se dispone de los siguientes campos en cada entrada de la lista:

- El nombre del minijuego en cuestión.
- La imagen con la que se identifica el minijuego.
- La opción de acceder al ránking individual, representada con el icono de una persona.
- La opción de acceder al ránking por grupos, que se representa con el icono de varias personas.

Cada uno de estos campos lleva asociado un evento de ratón en el que, al colocar el mismo sobre cada una de las opciones, muestra una etiqueta con el nombre de la opción. Pulsando sobre la opción *Ver ránking individual*, se llega a la pestaña *Ránking por juego individual*. La figura 5.15 representa un ejemplo de clasificación individual para un juego concreto, en este caso, el juego *Apuntados*.

Posición	Nombre	Grupo	★	⚡
1	asitorb	27	1	8000
2	sansimo	27	1	8000
3	davmore	21	1	8000
4	alecugn	21	1	8000
5	hugmata	21	1	8000

Figura 5.15: Ránking individual para el juego *Apuntados*

Esta interfaz muestra:

- En la parte superior izquierda, el nombre del minijuego seleccionado.
- A continuación, un selector desplegable para hacer un filtrado por grupo.
- La opción de descargar la clasificación en formato CSV. Aunque en la clasificación que se presenta en la plataforma únicamente aparecen las 10 primeras posiciones, este enlace permitirá la descarga de la lista completa.
- El resultado de la clasificación, incluyendo la posición, el nombre del usuario, su grupo, el número de estrellas obtenidas y su puntuación (respetando de nuevo la estructura usada para el ránking general).

Haciendo uso del filtrado por grupo, el resultado sería el reflejado en la figura 5.16.

Posición	Nombre	Grupo	★	⚡
1	andgome	Z10	1	8000
2	martgon	Z10	1	8000
3	aleguer	Z10	1	8000
4	hecgill	Z10	1	8000
5	alegonz	Z10	1	1500

Figura 5.16: Ránking individual para el juego *Apuntados* en el grupo *G10*

En este caso, solo aparecen los componentes del grupo seleccionado en la clasificación.

### 5.3.2. *Ránking por juego para el resto de usuarios*

Tal y como se comentó en la sección 4.2.2, aquellos usuarios de la plataforma que no tengan categoría de administrador (desarrolladores y jugadores) no disponen de un panel administrador. La solución que se propuso para permitir a estos usuarios disponer de esta posibilidad, fue añadir un acceso en la cabecera de la página, en la misma barra de herramientas en la que se encuentra el acceso al ránking global (figura 5.17)

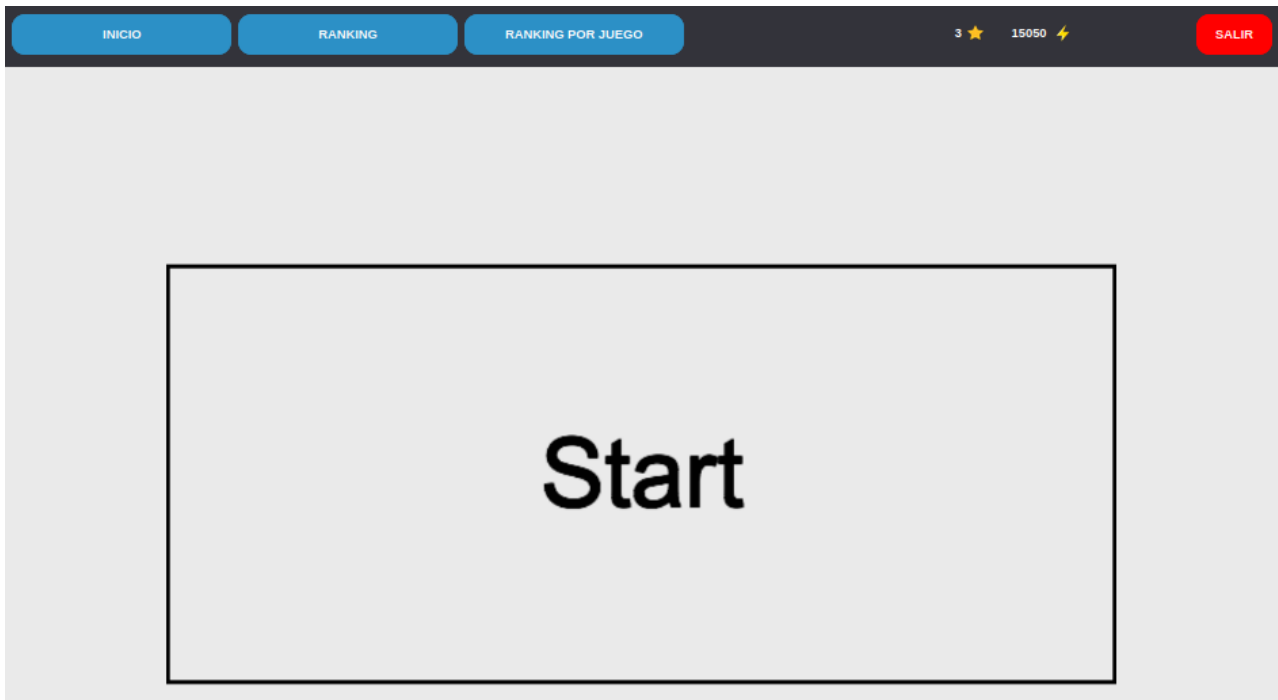


Figura 5.17: Nueva pestaña de *Ránking por juego* para usuarios no administradores

Pulsando en esta nueva opción, el usuario accede a la página *Hall of Fame* (figura 5.18), en la que puede seleccionar el juego para el que desee ver el ránking.

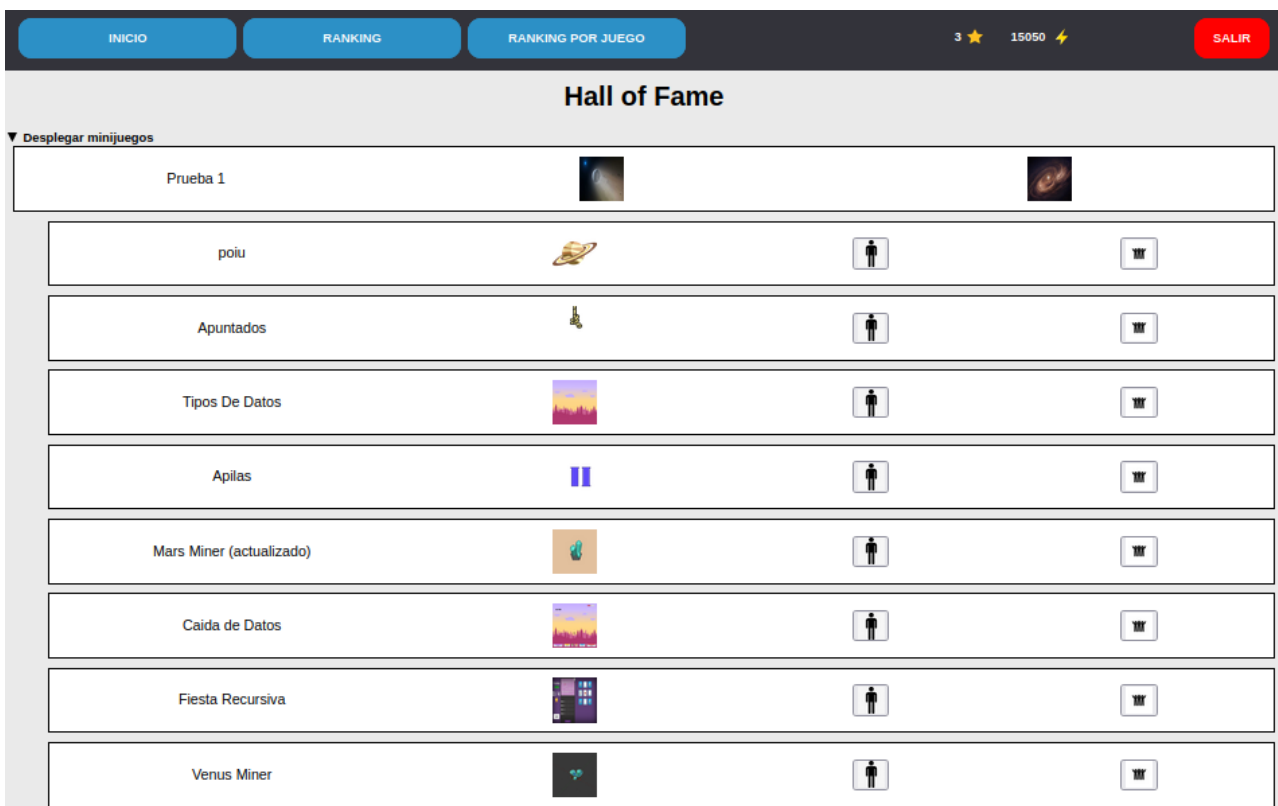
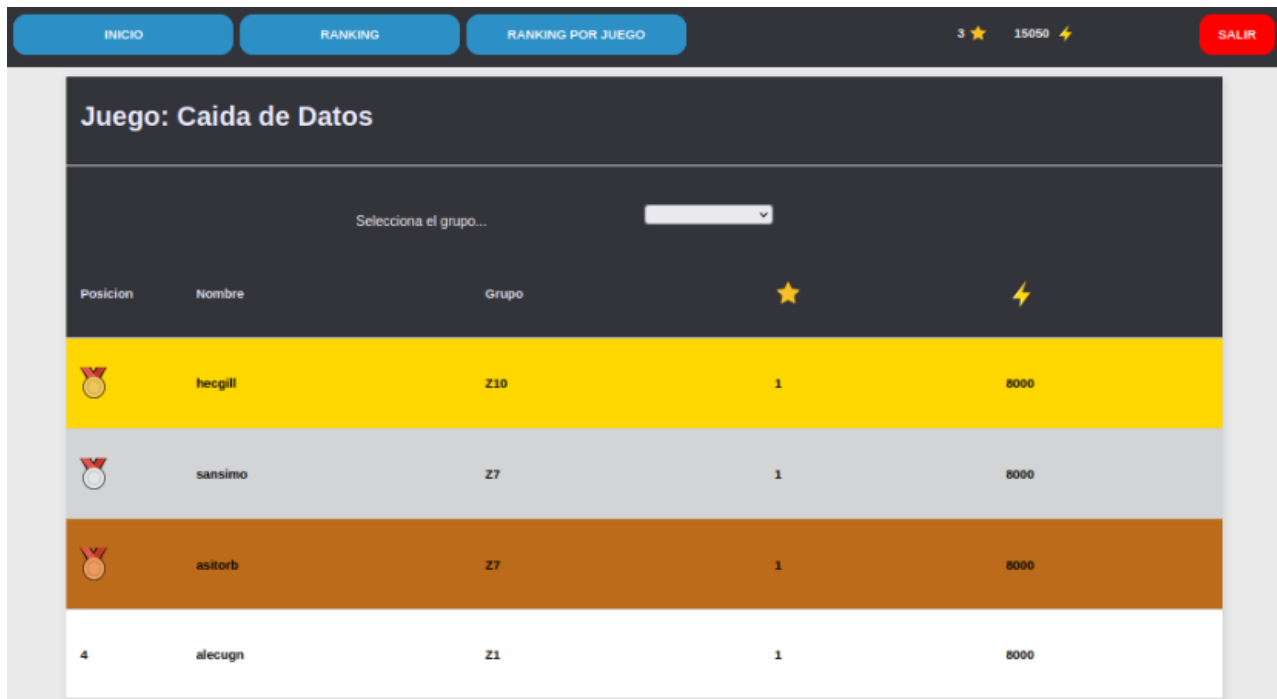


Figura 5.18: Página *Hall of Fame*

Para el caso del ránking individual, la vista es muy similar a la presentada para el caso de

los administradores, pero en este caso no está permitida la descarga del ránking por parte de estos jugadores (figura 5.19), por lo que esta opción no aparece en la interfaz.



The screenshot shows a web interface for a game called 'Caída de Datos'. At the top, there are navigation buttons: 'INICIO', 'RANKING', and 'RANKING POR JUEGO'. On the right, there is a user profile with 3 stars and 15050 points, and a red 'SALIR' button. Below the navigation is a header for the game 'Juego: Caída de Datos'. A dropdown menu labeled 'Selecciona el grupo...' is visible. The main content is a table with the following columns: 'Posicion', 'Nombre', 'Grupo', a star icon, and a lightning bolt icon. The table lists four players:

Posicion	Nombre	Grupo	★	⚡
1	hecgill	Z10	1	8000
2	sansimo	Z7	1	8000
3	asitorb	Z7	1	8000
4	alecugn	Z1	1	8000

Figura 5.19: Ránking individual para el juego *Caída de datos*

## 5.4. Diseño de un *Dashboard*

Con el fin de atender a los requerimientos de los docentes para mantener un mejor control y monitorización de la evolución del alumnado durante el uso de la plataforma para el desarrollo de la asignatura, fue requerido el diseño de un *dashboard* que sintetizase esta información.

Para ello, se utilizó la herramienta de prototipado *Balsamiq*, con la que se pudo hacer un diseño detallado de los diferentes aspectos que debía recoger este panel de información.

### 5.4.1. Ubicación del *Dashboard*

Puesto que este *Dashboard* sería únicamente accesible para aquellos usuarios dotados con el rol *Administrador*, se decidió ubicarlo en el apartado *Panel Administrador*. Y, siguiendo su estructura inicial, este se ubicaría dentro del panel de navegación lateral (figura 5.20, flecha 1).

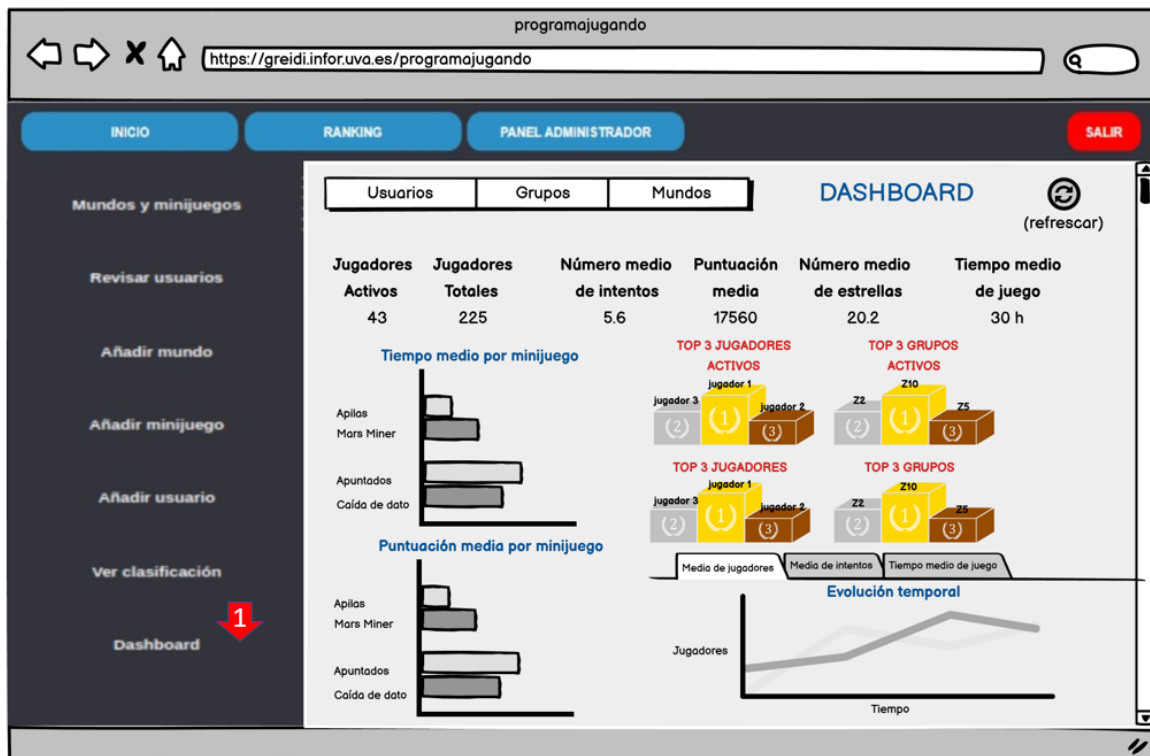


Figura 5.20: Ubicación y composición general del *dashboard*

### 5.4.2. Información general

La pantalla principal del dashboard (figura 5.20) aporta información de la situación en tiempo real de la plataforma. En la parte superior, se aportan una serie de *KPI (Key Performance Indicator)*, entre los que se encuentran:

- Número de jugadores activos en la plataforma.
- Número total de jugadores registrados.
- Número medio de intentos.
- Puntuación media global de los jugadores.
- Número medio de estrellas.
- Tiempo medio de juego global.



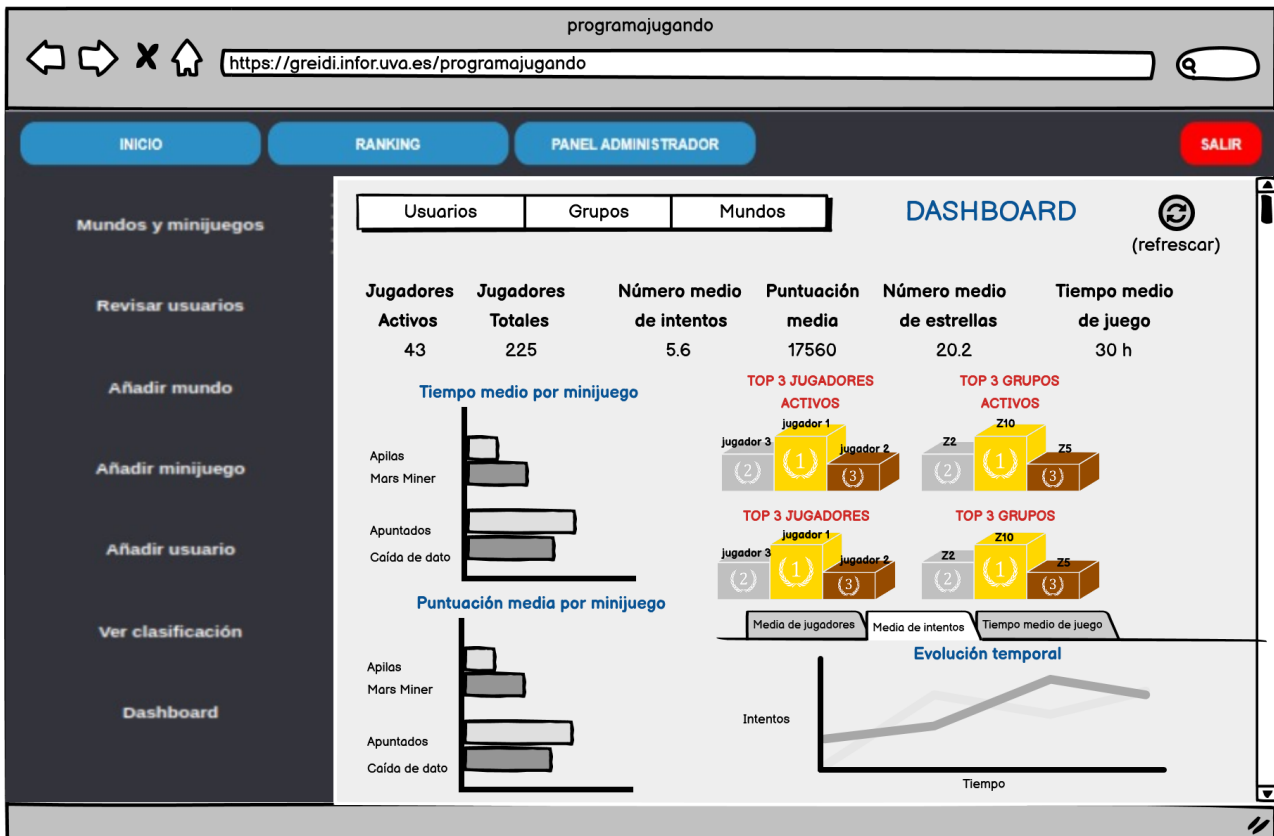


Figura 5.21: Vista general del dashboard con la subpestaña *Media de intentos*

Se incluye en forma de cuadrante de podios el top-3 de jugadores y grupos, tanto a nivel general, como para aquellos jugadores que estén activos en ese momento. Inmediatamente a continuación, en la parte inferior, se presentan la evolución temporal, de forma que se representa:

- La media de jugadores a lo largo del tiempo. (figura 5.20)
- La media de intentos a nivel global a lo largo del tiempo. (figura 5.21).
- La evolución del tiempo medio de juego. (figura 5.22).

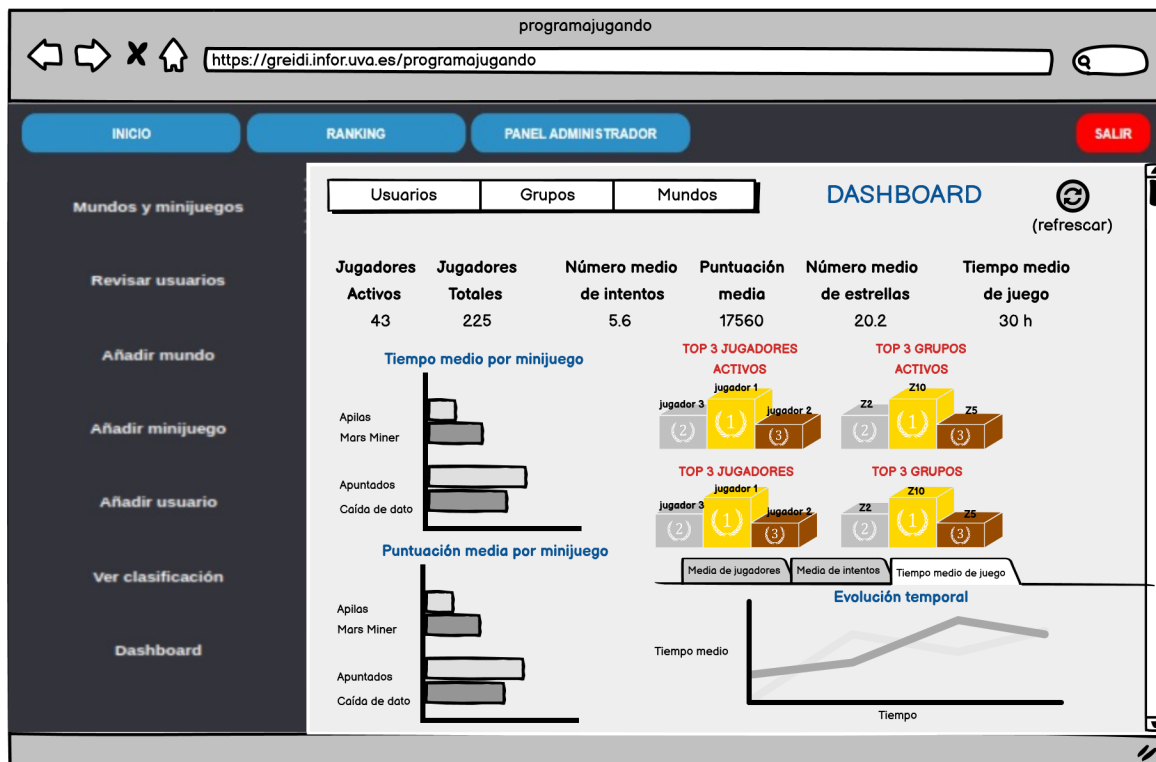


Figura 5.22: Vista general del dashboard con la subpestaña *Tiempo medio de juego*

Después, se presenta los gráficos de tiempo medio y puntuación media por minijuego, en el lateral izquierdo del dashboard. Y, para actualizar la información, se ubica el botón *refrescar* en la parte superior derecha.

Para navegar entre los diferentes elementos específicos del dashboard, se utiliza una barra de navegación ubicada en la parte superior izquierda, donde es posible acceder a los diferentes apartados: *Usuarios*, *Grupos* y *Mundos*.

### 5.4.3. Pestaña *Usuarios*

En esta primera pestaña, se visualiza la información relativa a los usuarios del sistema, y es la que aparece por defecto al entrar en el *dashboard*. Consta de su barra de navegación, y un selector de usuarios (figuras 5.23 y 5.30)

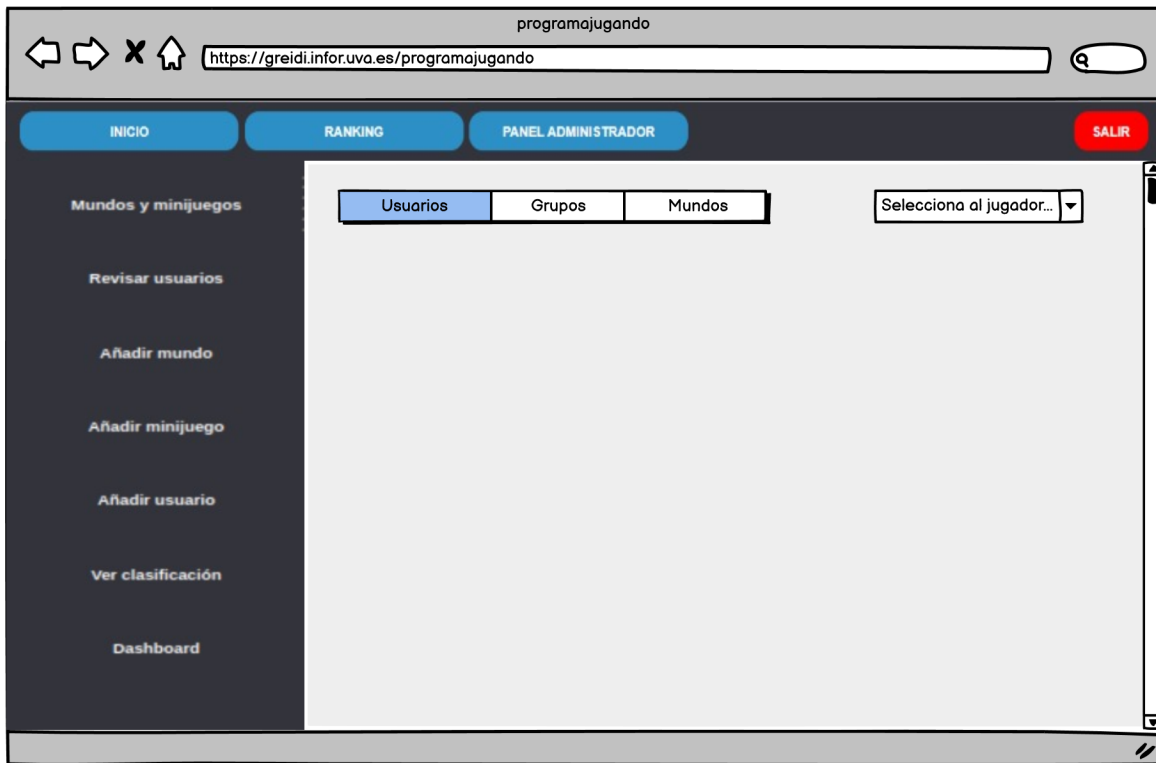


Figura 5.23: Pestaña *Usuarios* del *Dashboard*

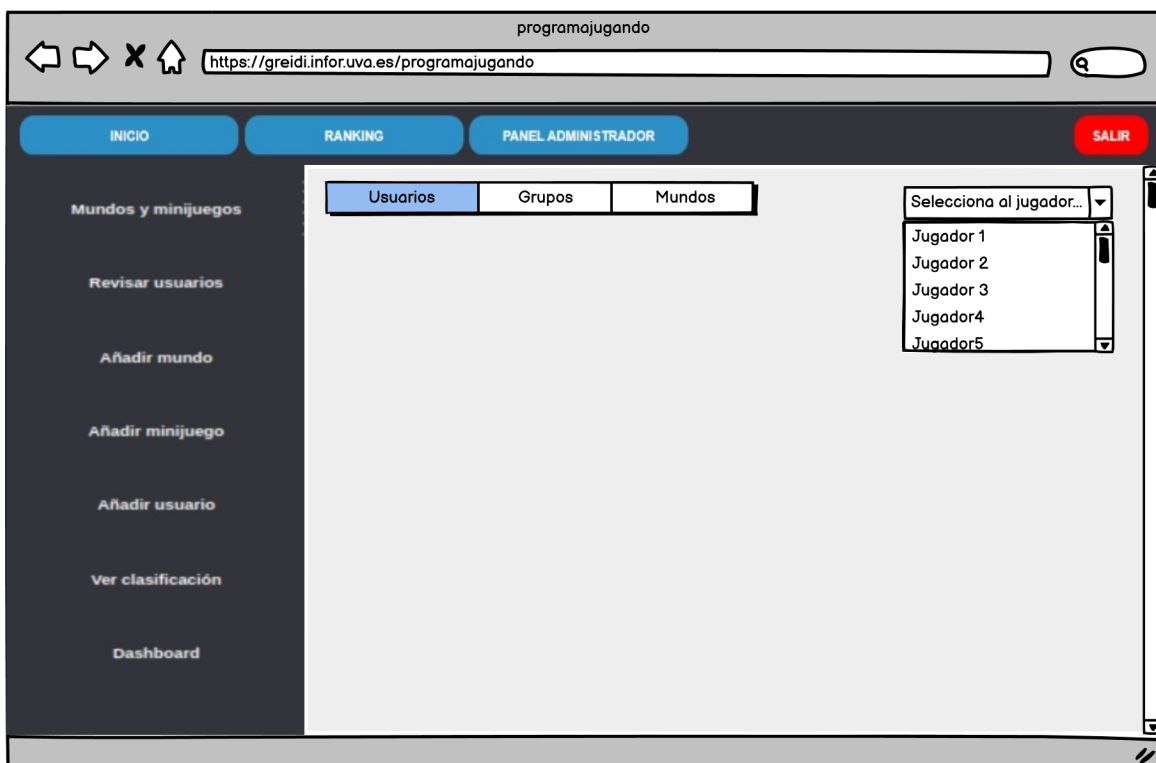


Figura 5.24: Pestaña *Usuarios* del *Dashboard* con los usuarios desplegados

Al seleccionar un usuario concreto, se muestra su información correspondiente (figura 5.25). Entre la información que se muestra, se encuentran:

- El grupo al que pertenece el jugador o jugadora.
- La suma total de puntos obtenida en todos los juegos.
- El total de estrellas obtenidas en todos los juegos.
- El tiempo total (en horas) que el usuario ha dedicado a jugar a los minijuegos de la plataforma.
- La fecha de la última partida jugada.
- Su posición en el ránking general de la plataforma.
- Su posición en el ránking grupal.

Además, cuenta con un podio que refleja los 3 minijuegos para los cuales el usuario en cuestión ha obtenido una mayor puntuación.



Figura 5.25: Pestaña *Jugadores* del dashboard con evolución de puntos por minijuego

Por otra parte, cuenta con un gráfico de líneas, que representa, para el mundo seleccionado (figura 5.27), los puntos obtenidos (figura 5.25) y las estrellas obtenidas (figura 5.26) para cada minijuego.



Figura 5.26: Pestaña *Jugadores* del dashboard con evolución de estrellas por minijuego



Figura 5.27: Pestaña *Jugadores* del dashboard con el selector de mundos desplegado



Figura 5.28: Pestaña *Jugadores* del dashboard con evolución de puntos por niveles para un minijuego concreto



Figura 5.29: Pestaña *Jugadores* del dashboard con evolución de estrellas por niveles para un minijuego concreto

También cuenta con la opción de filtrar por minijuego pulsando sobre el nombre de cada

minijuego, con la que se puede acceder a la evolución de los puntos (figura 5.28) y de las estrellas (figura 5.29) en cada nivel del minijuego seleccionado.

#### 5.4.4. Pestaña *Grupos*

En esta segunda pestaña, se muestra la información relativa a los grupos (figura 5.30).

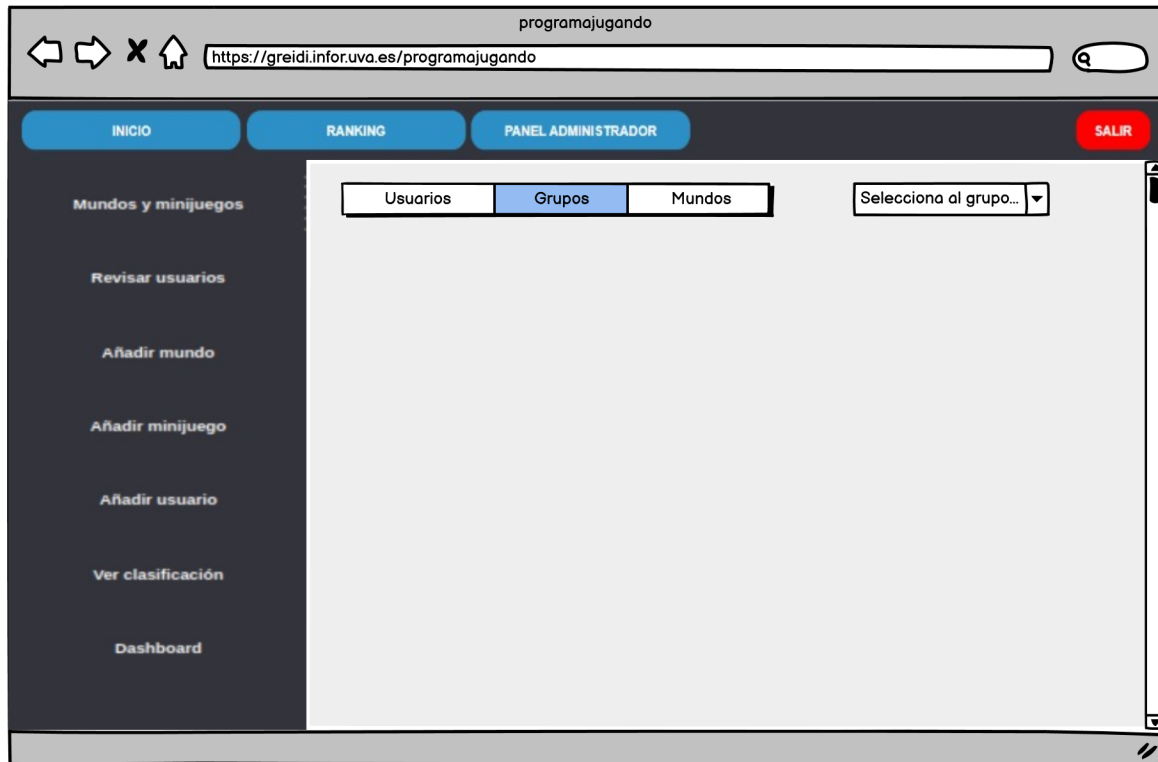


Figura 5.30: Pestaña *Grupos* inicial

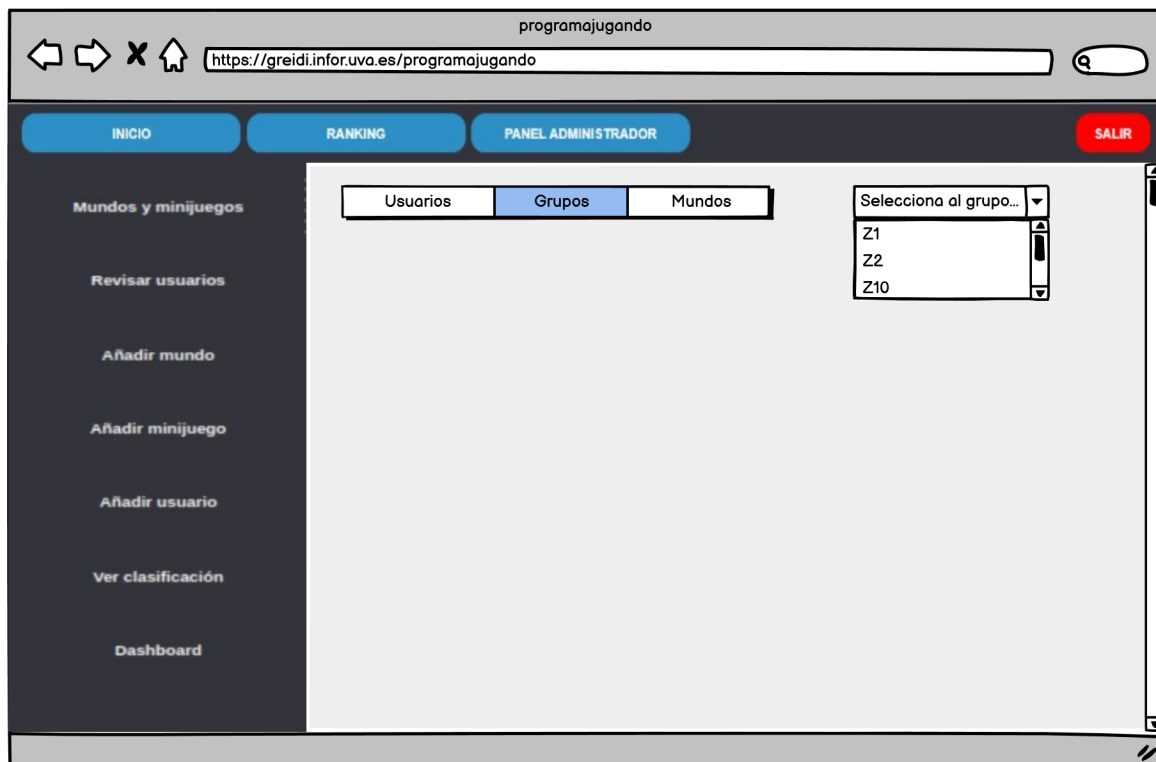


Figura 5.31: Pestaña *Grupos* con el selector de grupos desplegado

También contiene un selector en la parte superior izquierda, en el que se selecciona el grupo para el que se quiere obtener la información (figura 5.31). Una vez seleccionado el grupo, la información que se muestra es la que sigue (figura 5.32):

- El nombre del grupo.
- La posición general del grupo en el ránking.
- La suma total de puntos de todos los miembros del grupo, en todos los minijuegos.
- La suma total de estrellas de todos los miembros del grupo, en todos los minijuegos.
- La suma total (en horas) de tiempo jugado por los componentes del grupo.

Se muestra también una lista con los componentes del grupo, de forma que, seleccionando el nombre de cualquier jugador, se accede a la pestaña *Usuarios* del miembro correspondiente. Cuenta con un podio con los 3 juegos para los cuales se ha obtenido la mayor puntuación total.

Al igual que en el caso de la ventana *Usuarios*, se visualiza un gráfico de líneas con los puntos (figura 5.32) y estrellas (figura 5.33) por minijuego a lo largo del tiempo. Además, también se contará con una pestaña para la visualización de un gráfico de barras (figura 5.34) por juego.



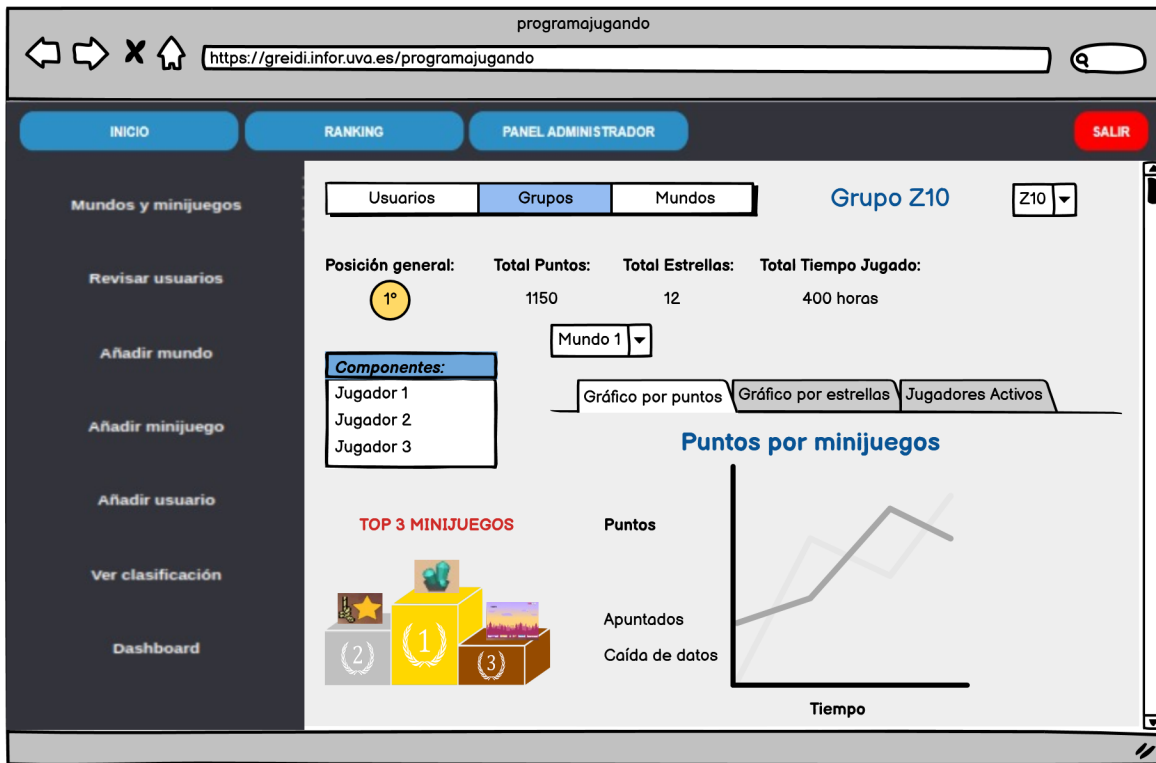


Figura 5.32: Pestaña *Grupos* con la evolución de puntos por minijuego



Figura 5.33: Pestaña *Grupos* con la evolución de estrellas por minijuego

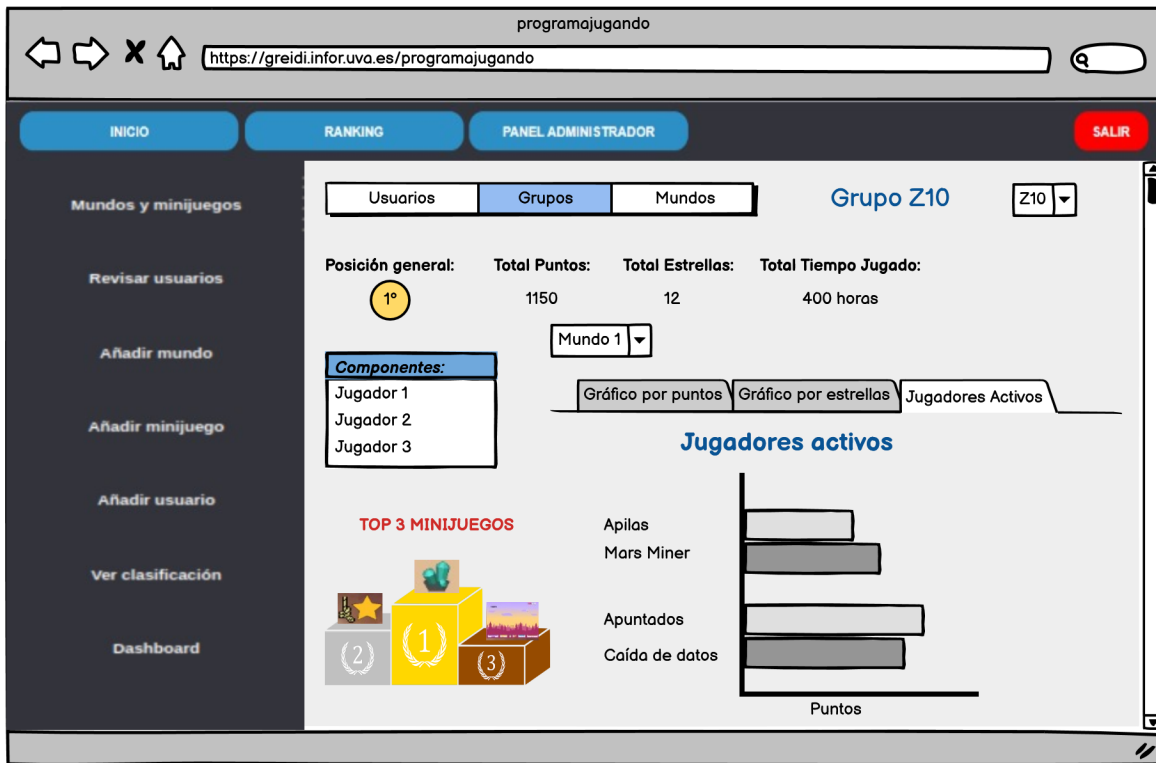


Figura 5.34: Pestaña *Grupos* con los jugadores activos por minijuego

Filtrando por minijuego (seleccionando en cualquiera de los gráficos el nombre del minijuego en cuestión), se obtendrán los puntos (figura 5.35) y estrellas ((figura 5.36) para cada nivel del minijuego.

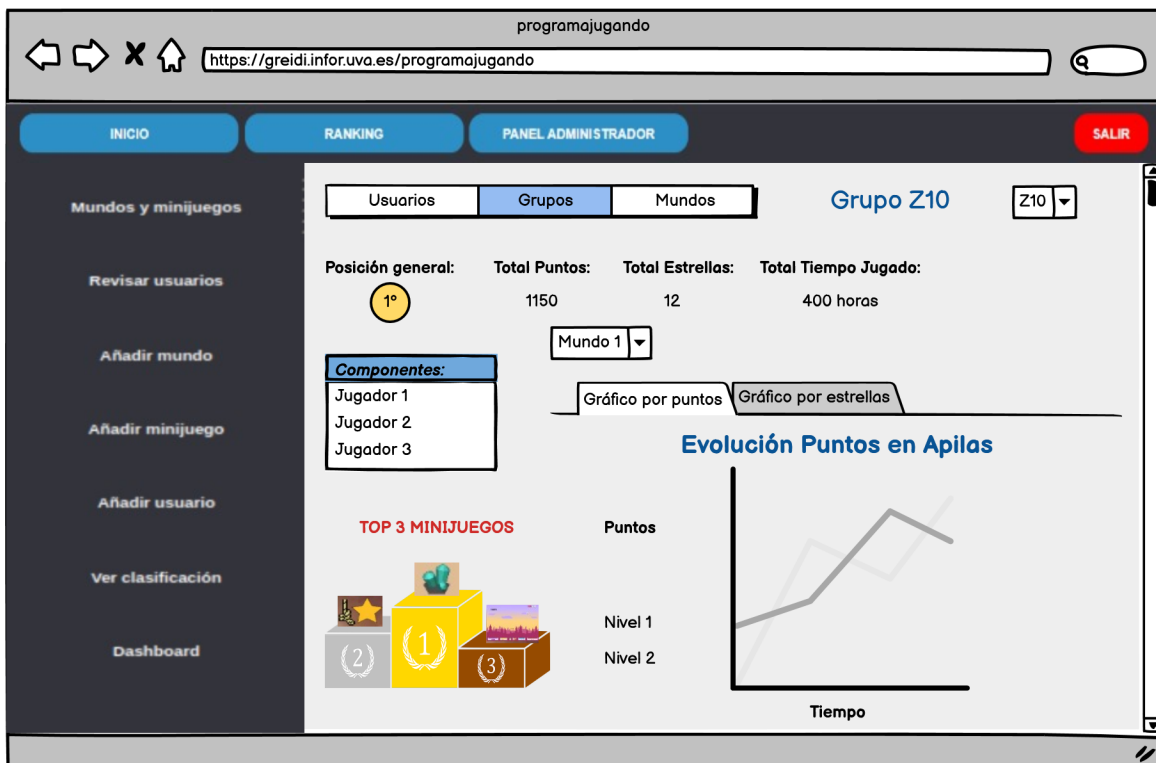


Figura 5.35: Pestaña *Grupos* con la evolución de puntos por niveles para un minijuego concreto



Figura 5.36: Pestaña *Grupos* con la evolución de estrellas por niveles para un minijuego concreto

#### 5.4.5. Pestaña *Mundos*

Esta última pestaña resume la información relativa a los mundos de la plataforma (figura 5.37). Siguiendo la estructura propuesta para las dos pestañas anteriores, dispone de un selector de mundos (figura 5.38) en la parte superior derecha.

Seleccionando un mundo concreto, la información se organiza en subpestañas, que presentan:

- La media de horas jugadas por minijuego (figura 5.39).
- La media de estrellas obtenidas por minijuego (figura 5.40).
- La media de puntos obtenidos por minijuego (figura 5.41).
- Los usuarios totales que han jugado a cada minijuego (figura 5.42).

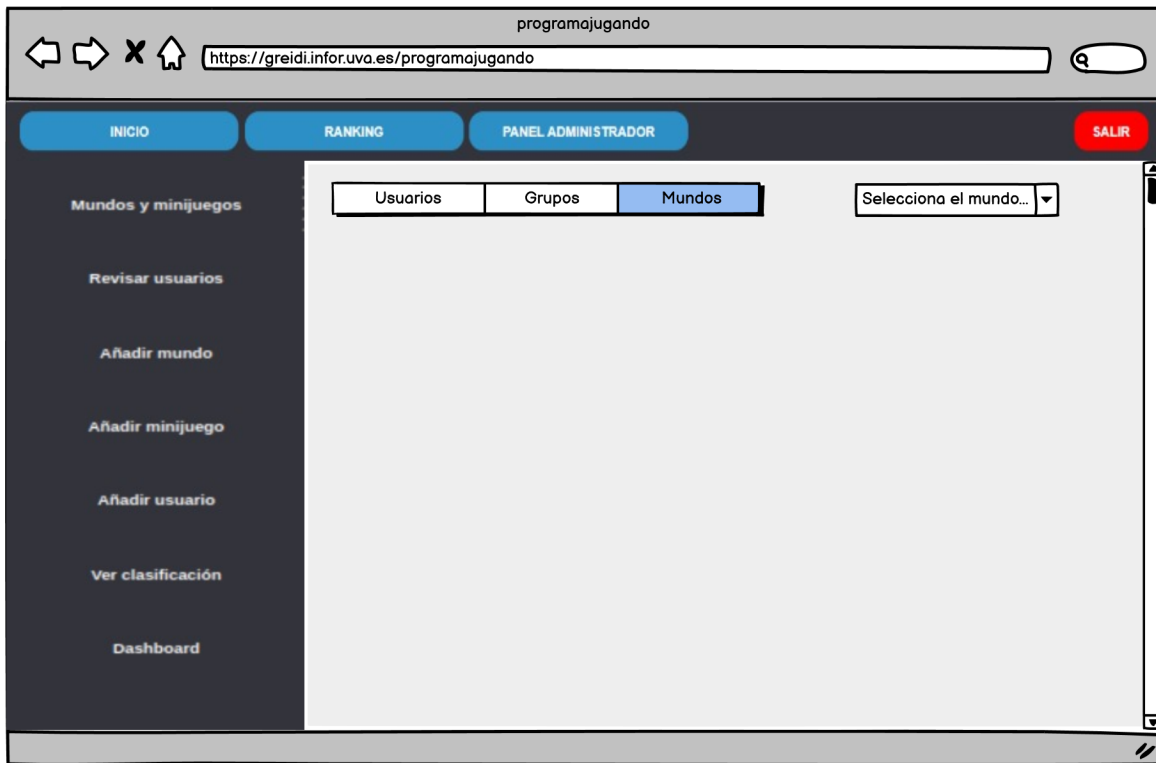


Figura 5.37: Pestaña *Mundos* inicial

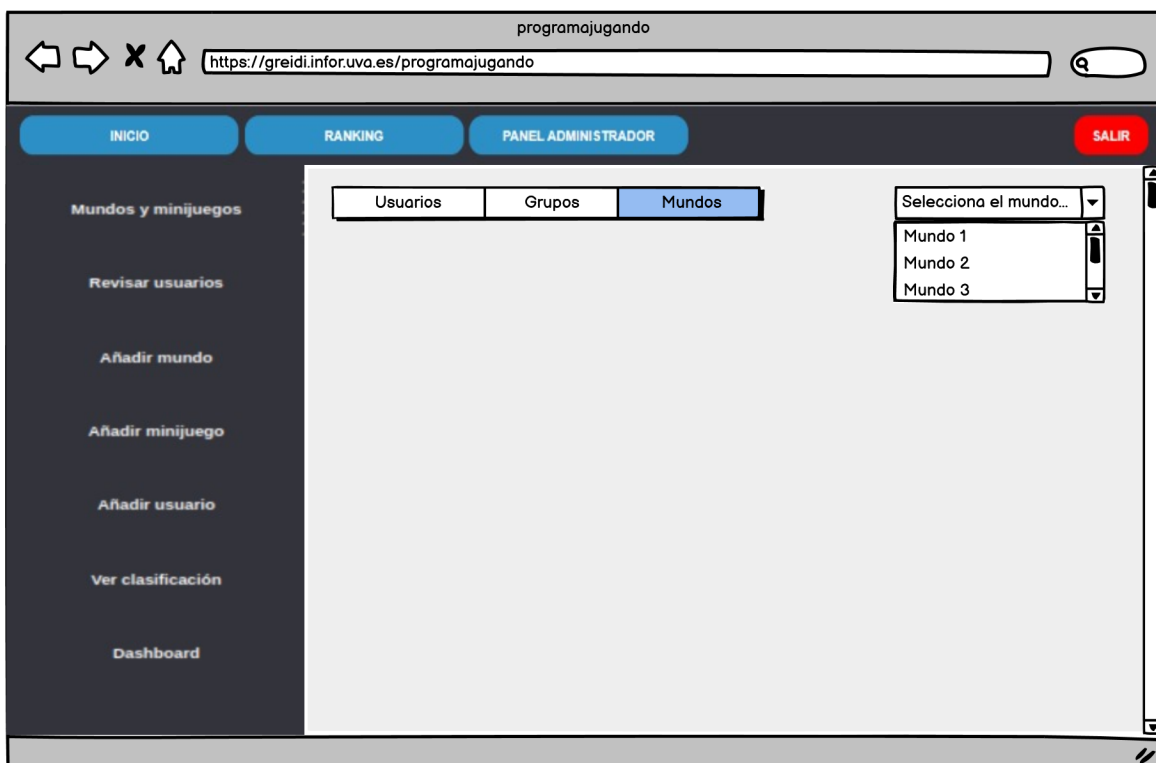


Figura 5.38: Pestaña *Mundos* con el selector de mundos desplegado



Figura 5.39: Pestaña *Mundos* con la media de horas por minijuego



Figura 5.40: Pestaña *Mundos* con la media de estrellas por minijuego



Figura 5.41: Pestaña *Mundos* con la media de puntos por minijuego



Figura 5.42: Pestaña *Mundos* con los usuarios totales por minijuego

En cada pestaña, se incluye también un podio, en el que se ubican los 3 minijuegos con mayor cantidad de horas, estrellas, puntos y usuarios.

# Capítulo 6

## Conclusiones

1. Tras el análisis de seguridad de la plataforma y la localización de las brechas de seguridad se necesitó ajustar el alcance de los objetivos a las capacidades máximas (300 horas) y esfuerzos estimados para la consecución de cada objetivo.
2. Se han conseguido los objetivos marcados en un 100 % tras la replanificación.
  - a) Se han eliminado las dependencias con *Unity* relacionadas con el acceso a los mini-juegos desde la plataforma, consiguiendo unificar la metodología de trabajo con la utilizada habitualmente en el desarrollo web.
  - b) Se ha conseguido identificar y solventar aquellas vulnerabilidades detectadas con la aplicación del patrón fachada para ocultar el código con las llamadas sensibles a la base de datos de la plataforma, y la encriptación y desencriptación de los datos en tránsito para evitar la falsificación de peticiones que pusieran en riesgo la integridad de los datos del sistema.
  - c) Se añadió la funcionalidad *Ránking por minijuegos* satisfactoriamente, otorgando la posibilidad de descarga en el caso de los usuarios *administradores*, y la visualización en el navegador para el resto de usuarios.
  - d) Se diseñó la interfaz del *dashboard* requerido por los docentes de la asignatura, sintetizando información relativa a los usuarios del sistema, los grupos de trabajo y los mundos disponibles en la plataforma. Sin embargo, no se consiguió implementar esta funcionalidad en el sistema, como consecuencia del aumento significativo del tiempo dedicado al análisis de seguridad del sistema.
3. El presente trabajo deja abierta la posibilidad de continuidad con la implementación del diseño realizado para el *Dashboard* de la plataforma.

# Bibliografía

- [1] Adcock, R. A., Thangavel, A., Whitfield-Gabrieli, S., Knutson, B., Gabrieli, J. D. (2006). *Reward-motivated learning: mesolimbic activation precedes memory formation*. *Neuron*, 50(3), 507–517. Recuperado de <https://doi.org/10.1016/j.neuron.2006.03.036>
- [2] Real Academia Española. (s.f.). Videojuego. En *Diccionario de la lengua española*. Recuperado en 07 de junio de 2022, de <https://dle.rae.es/videojuego?m=form>
- [3] Corvos, César Augusto, Peroni, Bruno Bizzozero, Pintos-Toledo, Enrique, Fernández-Giménez, Sofía, Brazo-Sayavera, Javier. (2020). Beneficios de los videojuegos activos sobre parámetros de aptitud física relacionada con la salud: un comentario en tiempos de cuarentena. *Revista Médica del Uruguay*, 36(4), 234-248. Epub 01 de diciembre de 2020. Recuperado de <https://doi.org/10.29193/rmu.36.4.11>
- [4] Grande De Prado, M. (2018). Beneficios educativos y videojuegos: revisión de la literatura española. *Education in the Knowledge Society (EKS)*, 19(3), 15–35. <https://doi.org/10.14201/eks20181933751> Recuperado de <https://revistas.usal.es/index.php/eks/article/view/eks20181931535/19379>
- [5] Kapp, Karl M. (2012) *The gamification of learning and instruction: game-based methods and strategies for training and education*
- [6] Rodríguez Sanz, Juan (2020) *Creación de una plataforma para la gamificación de una asignatura*. Universidad de Valladolid, Grado en Ingeniería Informática. Recuperado de <http://uvadoc.uva.es/handle/10324/44410>
- [7] Redacción KeepCoding (2022). *¿Cuánto gana un Desarrollador en España? [Salarios del 2022]* Recuperado de <https://keepcoding.io/blog/cuanto-gana-un-desarrollador-en-espana/>. Última consulta realizada el 10 de junio de 2022.
- [8] Fowler, Martin (2010). *Richardson Maturity Model*. Recuperado de <https://martinfowler.com/articles/richardsonMaturityModel.html>
- [9] *CryptoJS* Daemen, Joan, Rijmen, Vincent (1999) *AES Proposal: Rijndael* [https://www.cs.miami.edu/home/burt/learning/Csc688.012/rijndael/rijndael\\_doc\\_V2.pdf](https://www.cs.miami.edu/home/burt/learning/Csc688.012/rijndael/rijndael_doc_V2.pdf)