

**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



TRABAJO FIN DE GRADO
Grado en Ingeniería Informática
**PLATAFORMA WEB PARA EL
SEGUIMIENTO DEL APRENDIZAJE DE
TEORÍA MUSICAL A TRAVÉS DE
JUEGOS ONLINE**

Manual Técnico

Autor: Javier Ramírez Quintero
Director: Antonio Aráuzo Azofra
María Luque Rodríguez

Córdoba, septiembre de 2019



UNIVERSIDAD DE CÓRDOBA

AGRADECIMIENTOS

A mis padres, por todo lo que me han dado ya que sin ellos no hubiese sido capaz de que cumpliera mi vocación.

A mi hermana, por todo el apoyo que siempre me ha ofrecido, hasta en los momentos más difíciles.

A mis tutores, por todo el apoyo recibido a lo largo de este proyecto y todos estos años de carrera, que con su dedicación han conseguido sacar lo mejor de mí.

A todos los magníficos compañeros que he tenido, que entre todos hemos conseguido superar las dificultades que encontrábamos en el camino.

Y por último, y no menos especial, a todos mis amigos, por aguantar todas las locuras que tiene un informático y darme apoyo en todos los momentos difíciles a lo largo de esta etapa.

Índice de contenido

1. INTRODUCCIÓN	1
2. DEFINICIÓN DEL PROBLEMA	3
2.1. Identificación del problema real	3
2.2. Identificación del problema técnico	3
2.2.1. Funcionamiento	3
2.2.2. Entorno.....	3
2.2.3. Vida Esperada	3
2.2.4. Ciclo de mantenimiento	4
2.2.5. Competencia	4
2.2.6. Aspecto externo	4
2.2.7. Estandarización	4
2.2.8. Calidad y Fiabilidad.....	4
2.2.9. Programa de tareas	5
3. OBJETIVOS.....	7
4. ANTECEDENTES.....	9
5. RESTRICCIONES	13
5.1. Factores DATO	13
5.2. Factores estratégicos.....	13
6. RECURSOS	15
6.1. Recursos Humanos	15
6.2. Recursos Hardware.....	15
6.3. Recursos Software	15
7. ESPECIFICACIÓN DE REQUISITOS	17
7.1. Requisitos de información.....	17
7.2. Requisitos funcionales.....	17
7.3. Requisitos no funcionales.....	18
8. ANÁLISIS DEL SISTEMA.....	19
8.1. Identificación de actores.....	19
8.2. Identificación de casos de uso	19
8.2.1. Diagrama de casos de uso: Plataforma Web.....	20
8.3. Validación de casos de uso.....	27
9. DISEÑO DEL SISTEMA	29
9.1. Arquitectura del sistema	29

9.2.	Estructura de ficheros	30
9.3.	Diseño Procedimental.....	31
9.4.	Estructura Base de Datos	33
9.5.	Interfaz de usuario	34
9.5.1.	Diseño de la pantalla inicial.....	35
9.5.2.	Diseño de las pantallas de usuarios no registrados/alumnos	36
9.5.3.	Diseño de las pantallas del profesorado.....	36
10.	PRUEBAS	37
10.1.	Pruebas realizadas.....	37
10.2.	Sugerencias externas.....	38
11.	CONCLUSIONES Y FUTURAS MEJORAS	39
11.1.	Trabajo futuro	39
	BIBLIOGRAFÍA	41

Índice de figuras

Figura 1. Aplicación - ¡Vamos a la Ópera!	9
Figura 2. Aplicación – Toc and Roll	10
Figura 3. Aplicación – Little Musician: Musical Instruments	10
Figura 4. Plataforma Web – Juegos.com	11
Figura 5. Plataforma Web – Minijuegos	12
Figura 6. Ejemplo de BD con Modelo Relacional.....	13
Figura 7. Esquema Modelo-Vista-Controlador	14
Figura 8. Diagrama de casos de uso: Plataforma Web	20
Figura 9. Estructura MVC	29
Figura 10. Estructura del proyecto Plataforma Web	30
Figura 11. Diagrama de clases – Plataforma Web.....	31
Figura 12. Diagrama de secuencia – Recibir Puntuaciones (API)	32
Figura 13. Base de Datos Relacional (SGBD) de nuestro proyecto – Plataforma Web.	33
Figura 14. Interfaz de Usuario – Pantalla Inicial – Normal y Responsive	35
Figura 15. Interfaz de usuario – Perfil Alumnado – Normal y Responsive	36
Figura 16. Interfaz de usuario – Gestión de grupo – Normal y Responsive.....	36

Índice de tablas

Tabla 1. Definición de actores	19
Tabla 2. Plantilla Descripción de Caso de Uso	19
Tabla 3. Especificación – Registrar Alumno	21
Tabla 4. Especificación – Registrar Profesor	21
Tabla 5. Especificación - Jugar	22
Tabla 6. Especificación – Guardar Puntuaciones	22
Tabla 7. Especificación – Elegir Nivel	23
Tabla 8. Especificación – Iniciar Sesión	23
Tabla 9. Especificación – Crear Grupo	23
Tabla 10. Especificación – Gestionar Grupo	24
Tabla 11. Especificación – Visualizar Resultados	24
Tabla 12. Especificación – Visualizar Logros	25
Tabla 13. Especificación – Modificar Alumno	25
Tabla 14. Especificación – Modificar Profesor	25
Tabla 15. Especificación – Registrar Resultados	26
Tabla 16.: Especificación – Eliminar Usuarios Inactivos	26
Tabla 17. Matriz de Trazabilidad	27
Tabla 18. Prueba 1	37
Tabla 19. Prueba 2	37
Tabla 20. Prueba 3	37
Tabla 21. Prueba 4	37
Tabla 22. Prueba 5	38
Tabla 23. Prueba 6	38

1. INTRODUCCIÓN

Existen múltiples definiciones del término *Educación*, se puede decir que es la acción que lleva implícitamente un proceso de socialización del ser humano, a través del cual desarrollamos todas las capacidades físicas e intelectuales de éste para enfrentar la realidad de manera consciente, equilibrada y eficiente.

Si consideramos que solo aquellos que muestran aptitudes sobre música desde la infancia son aptos para recibir *educación musical*, es una equivocación. Todo ser humano al nacer posee de manera innata una predisposición a desarrollar cualquier tipo de inteligencia, nadie está negado a ninguna de ellas, tan solo hay que potenciar estas capacidades.

Desde la antigüedad, se ha reconocido el importante valor formativo de la música, considerada como una de las materias fundamentales para la educación del ser humano. Es por esto, que, a lo largo del siglo XX, pedagogos y psicólogos convienen en constatar la inestimable contribución que la educación musical realiza al desarrollo integral de la persona, al igual que coinciden en que la misma fortalece una serie de capacidades basadas en cuatro áreas fundamentales (cognitiva, afectiva, psicomotriz y social). [1]

En la actualidad, existen numerosos juegos para las dos grandes plataformas móviles, Android e iOS. Sin embargo, en la mayoría de los casos, éstos son distintos, lo que hacen que muchos sean usados exclusivamente en clase. Ya sean porque sólo están disponibles para una plataforma exclusiva o que éstos sean de pago.

Muchos alumnos que necesitan o quieren aprender más sobre el lenguaje musical, se ven con la necesidad de contratar profesores particulares, los cuáles, los inician en la música. El problema principal, que, al ser particulares, no disponen de los recursos actuales y suficientes para esta enseñanza, por lo que muchos optan por ir a academias.

Sin embargo, hoy en día, no existen apenas academias que inicien desde una edad temprana al lenguaje musical ya que como se ha citado anteriormente, no disponen de las tecnologías necesarias para este aprendizaje, y si las tienen, suelen estar muy obsoletas, por lo que dificulta en gran medida el aprendizaje del alumnado.

Ante esta situación, se estima que la mejor idea, es desarrollar una plataforma web, donde el alumnado pueda realizar las tareas que mande el profesorado, desde cualquier sitio, excluyendo las academias y en cualquier dispositivo electrónico.

2. DEFINICIÓN DEL PROBLEMA

2.1. Identificación del problema real

Actualmente la mayoría de las academias, se encuentran envueltas en diversos problemas en cuanto a la enseñanza musical infantil. Ya que la mayoría de las técnicas utilizadas no son efectivas, o son muy aburridas.

Es por eso, que muchas optan por incorporar las nuevas tecnologías para que sus actividades sean más efectivas y amenas. Nos referimos a actividades como juegos o herramientas para hacer cuestionarios online. A raíz de esto, les surge la necesidad de llevar un registro de lo que hacen sus alumnos y poder ver la progresión de su aprendizaje con estos juegos o herramientas.

2.2. Identificación del problema técnico

Para poder realizar la identificación de este problema se va a utilizar la técnica llamada Product Design Specification (PDS) [2], la cual documenta todas las limitaciones y requisitos que debe de cumplir nuestro diseño.

2.2.1. Funcionamiento

Actualmente se tienen varios juegos, los cuales sólo sirven para jugar y aprender, pero no es posible tener un seguimiento de lo que se hace, es por eso, que nuestro software ayudará a cambiar todo esto.

Cuando un usuario no está registrado puede acceder al juego y jugar el juego completo, el problema viene que cuando cierra el juego y vuelve a comenzar, no tiene nada guardado, ni niveles ni puntuaciones.

Si el usuario está registrado, el problema anterior no sucede.

El problema en sí viene dado cuando el profesorado quiere llevar un seguimiento de sus alumnos con respecto a los distintos juegos utilizados, y no puede ya que estos juegos no disponen de ninguna plataforma la cual lleve las puntuaciones y/o niveles del alumnado.

2.2.2. Entorno

El principal subobjetivo de nuestro proyecto es la realización de una plataforma web la cual se adapte a la perfección a cualquier dispositivo electrónico.

La plataforma web va dedicada en su mayor parte a un público infantil, por lo que su interfaz gráfica será sencilla e intuitiva. Del mismo modo, hará que la experiencia de usuario sea amena y divertida, sin dejar de lado el aprendizaje musical.

2.2.3. Vida Esperada

Nuestro proyecto, como todo el software existente, se encuentra ante un continuo cambio, debido a las necesidades del mercado.

Cabe destacar, que las tecnologías usadas son las más actuales, por lo que no se espera ninguna actualización al menos en los próximos 4 años.

Sin embargo, se pueden dar alguna serie de mejoras para nuestro proyecto.

2.2.4. Ciclo de mantenimiento

Al tratarse de un trabajo de fin de grado, se considera que su autor no será responsable del mantenimiento futuro de la plataforma web.

Sin embargo, el desarrollo de la aplicación debe estar sujeto a las reglas básicas de programación, para que las futuras mejor realizadas por otros programadores sean ordenadas, sencillas y fácil de entender.

2.2.5. Competencia

En la actualidad, existe gran diversidad de plataformas web, las cuales registran a los usuarios para poder acceder a los distintos juegos que ofrecen.

Sin embargo, muchas de ellas no sirven para el sistema educativo musical; en cambio, las que sí, no incentivan a los alumnos en clase de ninguna forma. Y otras tantas, poseen una interfaz gráfica demasiado compleja, lo que conlleva al abandono del aprendizaje musical.

Este proyecto hará hincapié en las problemáticas citadas anteriormente, resolviendo en primera instancia, el problema con la interfaz gráfica, haciendo que esta sea sencilla e intuitiva. Así como también, creando unos logros que incentiven al alumno de cara al aprendizaje musical.

2.2.6. Aspecto externo

La plataforma web dispondrá de dos interfaces gráficas, una interfaz adaptada a un público infantil, principalmente niños/as de 3-10 años, y otra interfaz adaptada a un público profesional del sector educativo. Esta última será prácticamente igual a la primera, pero enfocada más a la educación.

2.2.7. Estandarización

En este proyecto será necesario seguir los estándares HTML (Hypertext Markup Language) [3] y CSS (Cascading Style Sheets) [4], los cuales ayudarán a la elaboración de la plataforma web, y su posterior validación de la W3C (World Wide Web Consortium) [5], garantizando un correcto funcionamiento en todos los dispositivos y navegadores.

2.2.8. Calidad y Fiabilidad

En este proyecto, no se pone en juego la salud de ninguna persona puesto que se usarán dispositivos electrónicos (tablets, ordenadores), a su vez, no requerirá de ningún coste y, por tanto, de ninguna pérdida económica.

Gracias al proyecto, los juegos tomarán mayor importancia a la hora de interactuar con el alumnado, ya que se le dará opciones para guardar sus datos, así como sus puntuaciones, las cuales tendrán sus rankings y/o logros. Esto dará mayor visibilidad a las academias que usen este software.

No se espera que el proyecto presente ningún fallo a grande escala.

2.2.9. Programa de tareas

El desarrollo de la plataforma vendrá dado por las siguientes etapas:

✓ Aprendizaje y familiarización de las tecnologías a utilizar en el proyecto

Antes de comenzar el proyecto, se debe realizar un estudio de las tecnologías que se van a utilizar para el desarrollo de la plataforma web; como se hace referencia más adelante, se usará el framework Flask, HTML, CSS, JavaScript, Python y SQL.

✓ Análisis y diseño

Una vez adquiridos todos los conocimientos para el desarrollo, se debe realizar un estudio de las funcionalidades que se van a incluir en la plataforma web, las cuales deben cumplir los objetivos propuestos.

Así como también el diseño de la interfaz gráfica.

✓ Desarrollo e implementación

Esta fase comprende todo lo relacionado con el código fuente.

✓ Pruebas

En esta fase, se diseñan todas las pruebas necesarias para detectar posibles fallos que pueda ocasionar la plataforma web, y así, poderlos identificar y dar soluciones.

También, en esta fase se verificará y validará la plataforma web.

✓ Documentación

Simultáneamente a las fases indicadas, se realizarán tres documentos que corresponden a un proyecto de ingeniería informática.

- Manual técnico: este documento contiene la descripción del problema y la solución a este que se ha desarrollado en este proyecto. Se tratan los siguientes apartados:
 1. Introducción
 2. Descripción del problema
 3. Objetivos
 4. Antecedentes
 5. Restricciones
 6. Recursos
 7. Especificación de requisitos
 8. Análisis del sistema
 9. Diseño del sistema
 10. Pruebas
 11. Conclusiones y futuras mejoras
 12. Bibliografía
- Manual de usuario: este documento abarca todos los pasos a seguir para poder acceder a la plataforma web, así, como una guía de uso sencilla y concisa, para que cualquier usuario pueda aprender a usarla.

- Manual de código: incluirá el código fuente del desarrollo de la plataforma web, de modo que cualquier usuario pueda entender el código, las variables y el funcionamiento de las funciones.

3. OBJETIVOS

El objetivo principal es implementar una plataforma web para juegos relacionados con el aprendizaje musical. En ella se deben recoger todos los datos relacionados con los usuarios (profesores y estudiantes), así como los juegos jugados y sus puntuaciones.

Para ellos se plantean los siguientes subobjetivos:

- Permitir la utilización de la plataforma web desde cualquier dispositivo electrónico mediante diseño *responsive* (adaptativo) [6].
- Permitir integrar diferentes juegos en la plataforma web con una forma de puntuar y lanzar niveles homogénea.
- Permitir diferenciar entre los distintos usuarios (alumnos, profesores).
- Adaptar la interfaz de usuario a público de diferentes edades.
- Permitir probar los juegos sin necesidad de registro previo
- Registro y acceso de usuarios.
- Premiar a los alumnos registrados con logros por las puntuaciones obtenidas

4. ANTECEDENTES

Hoy en día podemos encontrar en la web muchas aplicaciones o juegos a disposición del usuario, que permiten aprender música, descubrir instrumentos, identificar sonidos, aprender la teoría del lenguaje musical, etcétera.

La gran mayoría de estas aplicaciones, son especialmente dedicadas para sistemas operativos (SO) móviles, lo que hace que tengan que depender de un dispositivo móvil (smartphone/tablet). Y gran parte de éstas, son básicamente el juego en sí, sin ningún tipo de registro de usuarios, por lo no permiten llevar un seguimiento por parte de terceros en cuanto al tema de puntuaciones.

A continuación, podemos observar algunas de ellas:

✓ **¡Vamos a la Ópera!** [7]

Plantea la trayectoria y asistencia como una auténtica diversión “sembrada de secretos”. Durante su ejecución, el escolar acudirá a una visita guiada y descubrirá “los secretos más íntimos del funcionamiento del teatro”. En esta aplicación tienen cabida desde cantantes, bailarines, hasta luces y escenografía.

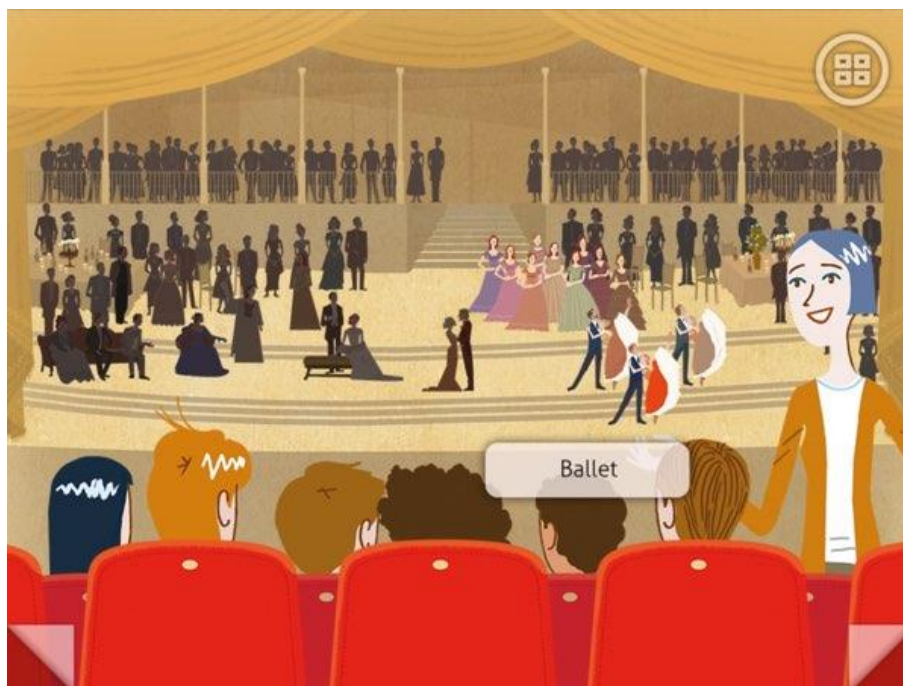


Figura 1. Aplicación - ¡Vamos a la Ópera!

✓ **Toc and Roll** [8]

Ideada para que los niños puedan componer sus primeras canciones y desarrollar su creatividad de una forma fácil, divertida e intuitiva. Ofrece la posibilidad de mezclar diferentes instrumentos en un multipistas virtual, añadir efectos, grabar la voz y compartir el resultado con el resto del mundo.



Figura 2. Aplicación – Toc and Roll

✓ **Little Musician - Musical Instruments [9]**

Enseña a los usuarios información básica sobre instrumentos musicales. En la aplicación, hay dos opciones principales. Primero, los usuarios pueden seleccionar la opción "Aprender", que les proporciona información básica sobre categorías de instrumentos musicales, como viento, metales, cuerdas, teclados e instrumentos de percusión. En segundo lugar, los usuarios pueden seleccionar la opción "Juego", que les proporciona "Juegos de dedos" y "Juegos de cartas" que pueden completar.



Figura 3. Aplicación – Little Musician: Musical Instruments

En cambio, existen otras, que sí están disponibles para cualquier dispositivo electrónico, que sí disponen de registro de usuarios y no disponen de seguimiento de terceros.

A continuación, podemos observar algunas de ellas:

✓ **Juegos.com [10]**

Plataforma web con gran cantidad de juegos musicales, además de otras categorías. Dispone de registro de usuarios, con vista de perfil puntuaciones y premios.

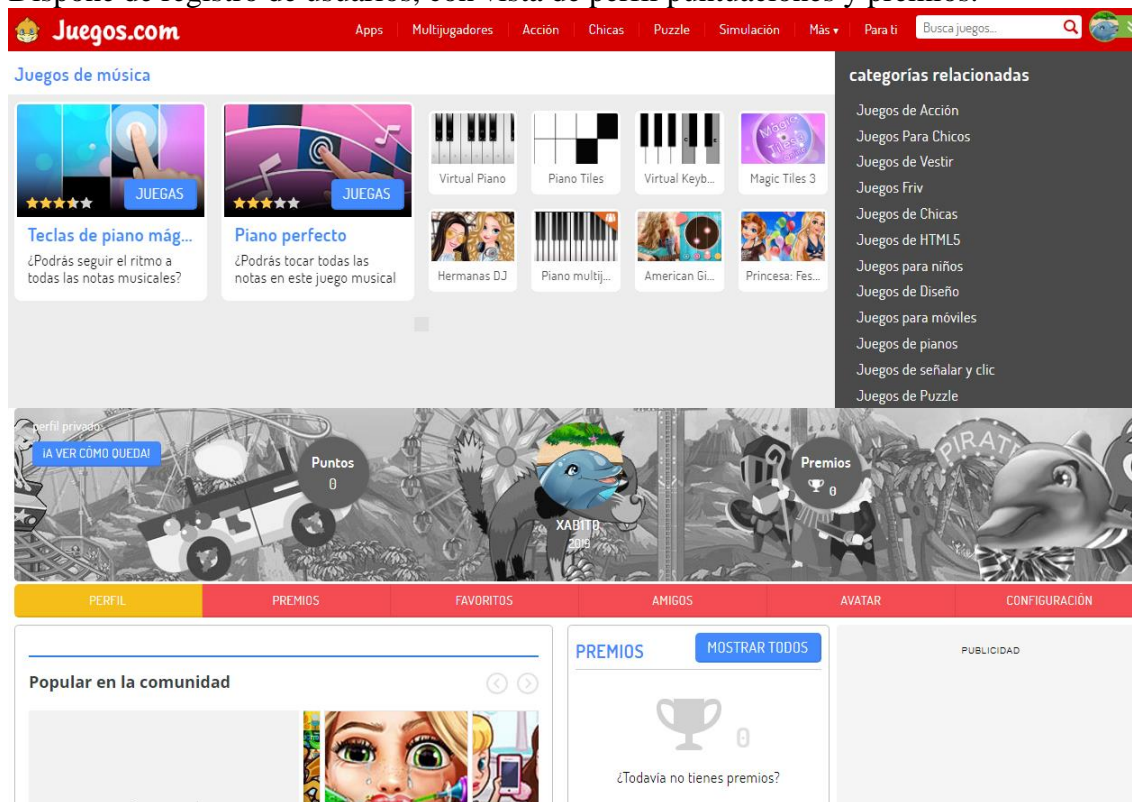
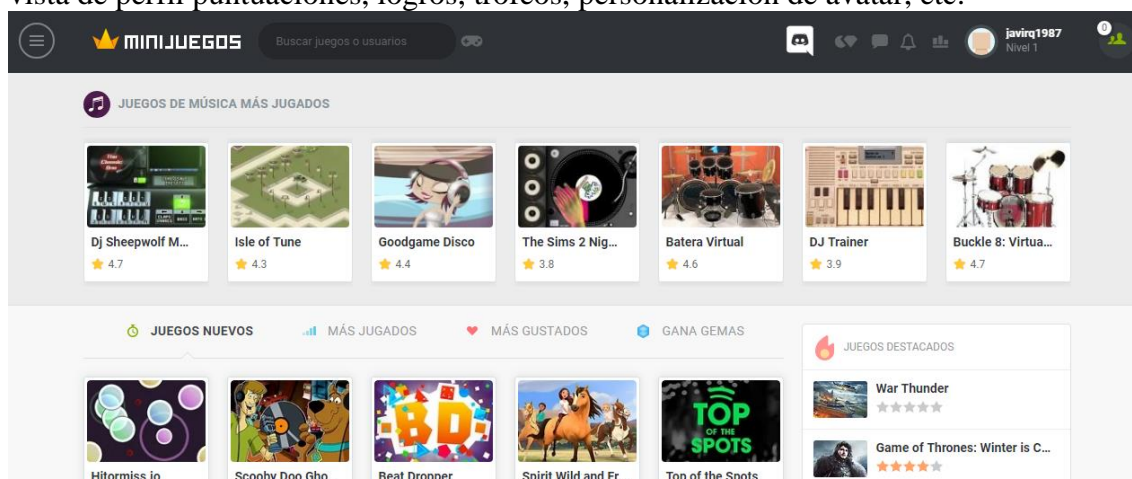


Figura 4. Plataforma Web – Juegos.com

✓ **Minijuegos [11]**

Plataforma web con gran cantidad de juegos musicales, aunque muchos no aptos para un público infantil, además de otras categorías. Dispone de registro de usuarios, con vista de perfil puntuaciones, logros, trofeos, personalización de avatar, etc.



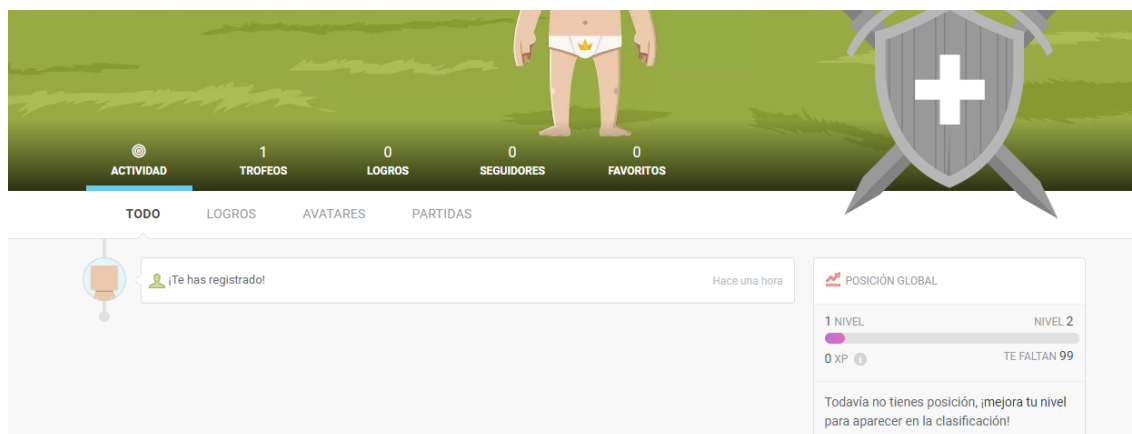


Figura 5. Plataforma Web – Minijuegos

La gran mayoría de estas aplicaciones que ya están en el mercado son exclusivas para SO móviles, tienen registro por Google/Facebook o manual. Este registro, es meramente para llevar un control de los niveles superados y/o puntuaciones de cada nivel.

En cambio, las plataformas web que existen actualmente en el mercado, disponen también de las características anteriores, y a parte, son compatibles con cualquier dispositivo electrónico. Además de todo esto, disponen de sistema de puntuaciones, logros, etc.

Estudiando cómo se encuentra hoy en día el mercado, y las pocas posibilidades de adaptar estas aplicaciones o plataformas web al sector educativo, más en concreto a academias. La plataforma web a desarrollar dará principalmente soporte a estas academias, escuelas, etc., con seguimiento de sus alumnos por parte de los profesores.

5. RESTRICCIONES

Llegados a este apartado, se estudiarán todos los factores o restricciones que limitan el ámbito del diseño y que son condicionales al desarrollo del producto. Estos factores se identifican como factores dato y estratégicos.

5.1. Factores DATO

Estos factores vienen impuestos por la naturaleza del producto o por el propio cliente, los cuales marcarán las pautas a seguir en la realización del software. En nuestro caso vendrán impuestos por los directores del proyecto.

La restricción principal impuesta es la de que la plataforma web debe tener un diseño responsive, la cual ofrezca la posibilidad de ser adaptada a cualquier dispositivo electrónico.

5.2. Factores estratégicos

Se tratan de decisiones que se han tomado para un mejor desarrollo del proyecto, lo que es lo mismo, éstas no estaban fijadas por naturaleza a nuestro proyecto.

Para la gestión de la base de datos se ha decidido usar MySQL.

MySQL [12] se trata de un sistema de gestión de base de datos relacional (SGBD), desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

La base de datos relacional (BDR) es un tipo de base de datos (BD) que cumple con el modelo relacional (el modelo más utilizado actualmente para implementar las BD ya planificadas).

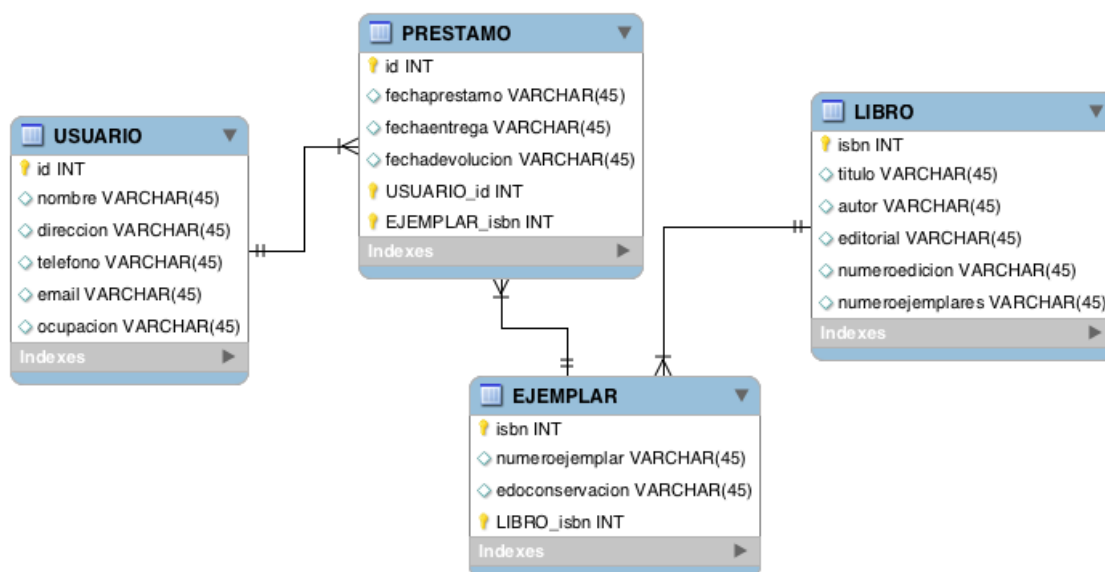


Figura 6. Ejemplo de BD con Modelo Relacional

Para el desarrollo de la plataforma web se ha decidido usar FLASK.

FLASK [13] es un “micro” framework escrito en Python basado en la especificación de Werkzeug y el motor de templates Jinja2. Viene concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC (Modelo Vista Controlador).

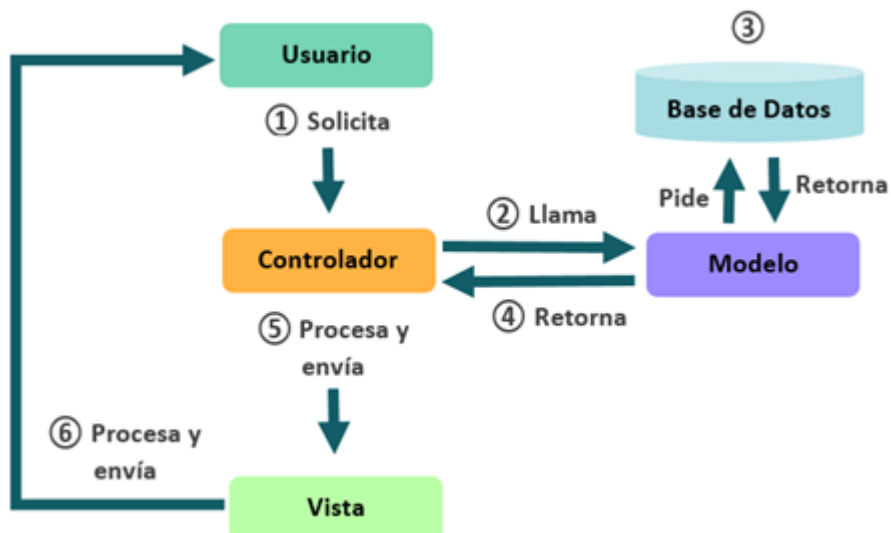


Figura 7. Esquema Modelo-Vista-Controlador

6. RECURSOS

6.1. Recursos Humanos

Este proyecto será llevado a cabo por:

- ✓ *Javier Ramírez Quintero*, alumno del Grado en Ingeniería Informática con mención en Software y dirigido por:
- ✓ *Antonio Araúzo Azofra*, profesor doctor perteneciente al área de Proyectos de Ingeniería,
- ✓ *María Luque Rodríguez*, profesora doctora perteneciente al departamento de Informática y Análisis Numérico.

6.2. Recursos Hardware

En recursos hardware, se dispone de un ordenador portátil Lenovo G505, con las especificaciones siguientes:

- ✓ **Procesador:** AMD - A4/5000 (Quad-core - 2MB - 1'5 GHz).
- ✓ **Memoria RAM:** 4 GB DDR3.
- ✓ **Almacenamiento:** 240 GB SSD.
- ✓ **Tarjeta de vídeo:** AMD Radeon HD 8570M (1GB).

6.3. Recursos Software

En recursos software, se disponen de los siguientes:

- ✓ **Sistema Operativo:** Linux Mint 19 Cinnamon (64 bits).
- ✓ **Software de Desarrollo:** Framework Flask.
- ✓ **Herramientas de Documentación:** Google Docs, Microsoft Office.
- ✓ **Editor de Texto:** Atom, Notepad++.
- ✓ **Lenguajes de Programación:** PHP, HTML, CSS, Python, JavaScript, jQuery, SQL
- ✓ **Sistema de Gestión de BBDD:** MySQL

7. ESPECIFICACIÓN DE REQUISITOS

7.1. Requisitos de información

Llegado a este punto, se recogerá toda la información que se estime necesaria acerca de los requisitos de información.

A continuación, se pasa a definir los requisitos funcionales que debe satisfacer nuestro sistema:

- ✓ **RI-01:** Nombre completo del alumno
- ✓ **RI-02:** Nombre de usuario del alumno
- ✓ **RI-03:** Contraseña de usuario del alumno
- ✓ **RI-04:** Año de nacimiento del alumno
- ✓ **RI-05:** Correo electrónico del alumno
- ✓ **RI-06:** Nombre completo del profesor
- ✓ **RI-07:** Nombre de usuario del profesor
- ✓ **RI-08:** Contraseña de usuario del profesor
- ✓ **RI-09:** DNI del profesor
- ✓ **RI-10:** Teléfono del profesor
- ✓ **RI-11:** Correo electrónico del profesor
- ✓ **RI-12:** Género del profesor
- ✓ **RI-13:** Nombre del grupo de clase
- ✓ **RI-14:** Aula del grupo de clase
- ✓ **RI-15:** Nombre del juego
- ✓ **RI-16:** Descripción del juego
- ✓ **RI-17:** Niveles del juego

7.2. Requisitos funcionales

La plataforma web nos da la posibilidad de crear un registro de usuarios que juegan a determinados juegos musicales y obtienen una serie de puntuaciones de forma más dinámica.

A continuación, se pasa a definir los requisitos funcionales que debe satisfacer nuestro sistema:

- ✓ **RF-01:** El sistema deberá registrar a un alumno
- ✓ **RF-02:** El sistema deberá registrar a un profesor
- ✓ **RF-03:** El sistema deberá distinguir entre alumno y profesor
- ✓ **RF-04:** El sistema deberá restringir el acceso a usuarios no registrados
- ✓ **RF-05:** El sistema deberá mostrar el perfil del alumno
- ✓ **RF-06:** El sistema deberá mostrar el perfil del profesor
- ✓ **RF-07:** El sistema deberá permitir modificar los datos del alumno
- ✓ **RF-08:** El sistema deberá permitir modificar los datos del profesor
- ✓ **RF-09:** El sistema deberá crear un nuevo grupo de clase
- ✓ **RF-10:** El sistema deberá permitir modificar un grupo de clase
- ✓ **RF-11:** El sistema deberá permitir eliminar un grupo de clase
- ✓ **RF-12:** El sistema deberá mostrar los resultados de cada alumno perteneciente cada grupo de clase, al profesor
- ✓ **RF-13:** El sistema debe comprobar que el alumno ha jugado a un juego
- ✓ **RF-14:** El sistema deberá mostrar los resultados del alumno de cada juego jugado

- ✓ **RF-15:** El sistema deberá mostrar los logros del alumno
- ✓ **RF-16:** El sistema deberá permitir elegir a que juego jugar
- ✓ **RF-17:** El sistema deberá permitir elegir los distintos niveles del juego
- ✓ **RF-18:** El sistema deberá guardar las puntuaciones de cada jugador
- ✓ **RF-19:** El sistema deberá comprobar el tiempo sin acceder (alumno, profesor) a la plataforma
- ✓ **RF-20:** El sistema deberá eliminar a los alumnos/profesores inactivos en seis meses o más
- ✓ **RF-21:** El sistema deberá permitir jugar al usuario no registrado un nivel

7.3. Requisitos no funcionales

Se tratan de aquellos recursos que no se refieren directamente a las funciones específicas del sistema, más bien a las características con las que se dota a este, tales como la arquitectura lógica, la infraestructura, la interfaz gráfica, el rendimiento, la disponibilidad, la seguridad, la portabilidad y la integración. Se pasa a detallar nuestros requisitos no funcionales:

- ✓ **RNF-01:** El sistema debe ser fiable y robusto.
- ✓ **RNF-02:** El sistema debe estar disponible 24/7.
- ✓ **RNF-03:** La plataforma web debe ser capaz de adaptarse a cualquier dispositivo electrónico, diseño responsive.
- ✓ **RNF-04:** La plataforma web debe ser fácil de usar e intuitiva.
- ✓ **RNF-05:** La interfaz gráfica de la plataforma web debe estar adaptada para un público infantil y para un público profesional del sector educativo

8. ANÁLISIS DEL SISTEMA

8.1. Identificación de actores

Se define actor como una entidad o entidades externas que guardan relación con el sistema y que demandan una funcionalidad de éste.

A continuación, se pasa a definir los actores que concurren nuestro sistema:

NOMBRE DEL ACTOR	BREVE DESCRIPCIÓN
Usuario no registrado	Persona que puede acceder a la plataforma y jugar el nivel 1, de cada juego
Alumno	Persona que puede jugar a cualquier juego de la plataforma, ver sus puntuaciones y logros
Profesor	Persona que puede ver los resultados de cada alumno, crear y gestionar sus grupos. Pero no puede jugar a ningún juego
Juego Musical	Juego con temática musical
Tiempo	Periodo determinado durante el que se realiza una acción o se desarrolla un acontecimiento

Tabla 1. Definición de actores

8.2. Identificación de casos de uso

A continuación, se realizará una descripción del comportamiento, funcionalidad y comunicación que debe de tener el sistema mediante sus interacciones con los actores.

Para realizar las descripciones asociadas a cada caso de uso detectado, se usará la siguiente plantilla:

Caso de uso: Título	
Id	Identificador del caso de uso
Descripción	Breve párrafo, resumiendo sin entrar en detalles, el objetivo del caso de uso
Actores principales	Actores que interactúan directamente con el sistema
Actores secundarios	Actores que interactúan indirectamente con el sistema
Precondiciones	Condiciones que se deben dar para comenzar el caso de uso
Flujo principal	Se describen los pasos que sucedería en el caso ideal, o lo que es lo mismo en el camino correcto
Postcondiciones	Condiciones que se deben cumplir para finalizar el caso de uso
Flujo Alternativo	Flujo independiente al principal, que se aplica a todo el caso de uso

Tabla 2. Plantilla Descripción de Caso de Uso

8.2.1. Diagrama de casos de uso: Plataforma Web

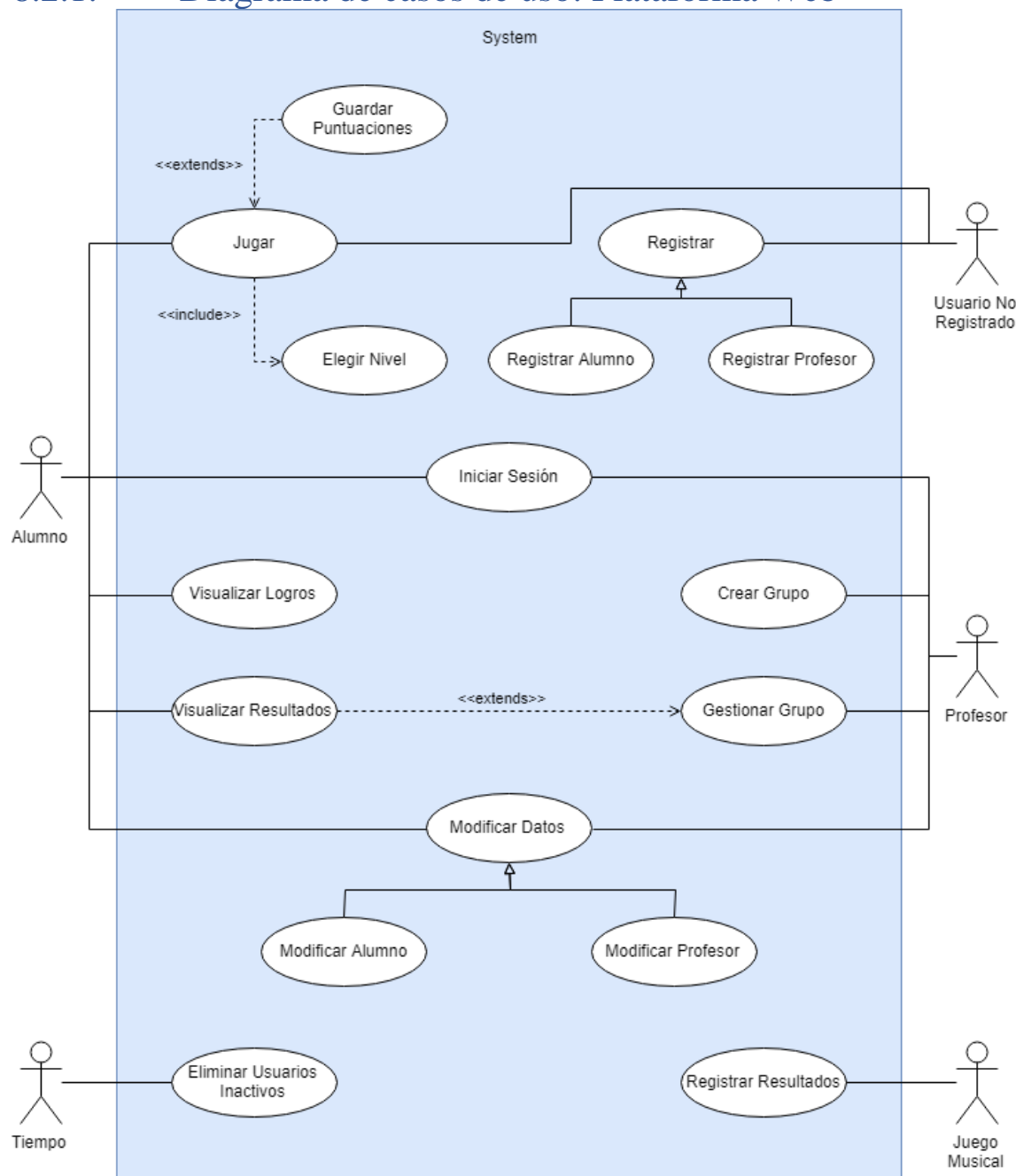


Figura 8. Diagrama de casos de uso: Plataforma Web

Caso de uso: Registrar Alumno	
Id	1
Descripción	Un usuario crea una cuenta en el sistema
Actores principales	Usuario no registrado
Actores secundarios	Alumno
Precondiciones	La cuenta no debe existir previamente
Flujo principal	<ol style="list-style-type: none"> 1. El usuario no registrado selecciona la opción “Usuarios No Registrados” 2. El usuario no registrado selecciona la opción “Registro” 3. El sistema le pide al usuario no registrado los datos de registro 4. El usuario no registrado introduce los datos requeridos 5. El sistema registra al alumno
Postcondiciones	El alumno debe haberse registrado
Flujo Alternativo	Ninguno

Tabla 3. Especificación – Registrar Alumno

Caso de uso: Registrar Profesor	
Id	2
Descripción	Un usuario crea una cuenta en el sistema
Actores principales	Usuario no registrado
Actores secundarios	Profesor
Precondiciones	La cuenta no debe existir previamente
Flujo principal	<ol style="list-style-type: none"> 1. El usuario no registrado selecciona la opción “Profesorado” 2. El sistema pide una clave de acceso <p><i>Mientras el sistema no valida los datos:</i></p> <ol style="list-style-type: none"> 2.1 El sistema valida la clave introducida 3. El usuario no registrado selecciona la opción “Registro” 4. El sistema le pide al usuario no registrado los datos de registro 5. El usuario no registrado introduce los datos requeridos 6. El sistema registra al profesor
Postcondiciones	El profesor debe haberse registrado
Flujo Alternativo	2.2. El sistema no valida la clave

Tabla 4. Especificación – Registrar Profesor

Caso de uso: Jugar	
Id	3
Descripción	El alumno/usuario no registrado juega a un juego musical
Actores principales	Alumno/Usuario no registrado
Actores secundarios	Ninguno
Precondiciones	No es necesario estar registrado
Flujo principal	<ol style="list-style-type: none"> 1. El alumno/usuario no registrado selecciona la opción “Jugar” 2. El alumno/usuario no registrado finaliza el primer nivel 3. El sistema pide al alumno/usuario no registrado los datos de acceso 4. El alumno/usuario no registrado introduce los datos requeridos <p><i>Mientras el sistema no valida los datos:</i></p> <ol style="list-style-type: none"> 4.1.El sistema valida los datos <p><i>include (Elegir Nivel)</i></p> <ol style="list-style-type: none"> 5. El sistema devuelve la puntuación del nivel <p><i>extension point (Guardar Puntuaciones)</i></p> <ol style="list-style-type: none"> 6. El alumno sigue jugando
Postcondiciones	El alumno debe de tener las puntuaciones guardadas
Flujo Alternativo	4.2. El sistema no valida los datos

Tabla 5. Especificación - Jugar

Caso de uso: Guardar Puntuaciones	
Id	4
Descripción	El sistema guarda las puntuaciones de cada juego
Actores principales	Ninguno
Actores secundarios	Alumno
Precondiciones	El alumno se ha registrado y ha jugado al menos un nivel del juego
Flujo principal	<ol style="list-style-type: none"> 1. El sistema comprueba que el alumno ha jugado al menos un nivel del juego 2. El sistema recibe las puntuaciones de cada nivel
Postcondiciones	El sistema guarda las puntuaciones del alumno
Flujo Alternativo	Ninguno

Tabla 6. Especificación – Guardar Puntuaciones

Caso de uso: Elegir Nivel	
Id	5
Descripción	El alumno elige un nivel del juego
Actores principales	Alumno
Actores secundarios	Ninguno
Precondiciones	El alumno ha iniciado sesión en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El sistema muestra los niveles a los que puede jugar 2. El alumno selecciona el nivel al que quiere jugar 3. El sistema registra el nivel
Postcondiciones	El sistema registra el nivel elegido
Flujo Alternativo	Ninguno

Tabla 7. Especificación – Elegir Nivel

Caso de uso: Iniciar Sesión	
Id	6
Descripción	El alumno/profesor se identifica en el sistema
Actores principales	Alumno/Profesor
Actores secundarios	Ninguno
Precondiciones	El alumno/profesor debe estar registrado
Flujo principal	<ol style="list-style-type: none"> 1. El alumno/profesor selecciona la opción “Acceso” 2. El sistema pide al alumno/profesor los datos de acceso 3. El alumno/profesor introduce los datos requeridos <p><i>Mientras el sistema no valida los datos:</i></p> <ol style="list-style-type: none"> 3.1.El sistema valida los datos introducidos 4. El alumno/profesor accede a su cuenta
Postcondiciones	El alumno/profesor queda logueado en el sistema
Flujo Alternativo	3.2.El sistema no valida los datos

Tabla 8. Especificación – Iniciar Sesión

Caso de uso: Crear Grupo	
Id	7
Descripción	El profesor crea un grupo de clase
Actores principales	Profesor
Actores secundarios	Ninguno
Precondiciones	El profesor debe haber iniciado sesión en el sistema
Flujo principal	<ol style="list-style-type: none"> 1. El profesor selecciona la opción “Crear” 2. El sistema pide al profesor el nombre y aula del grupo 3. El profesor introduce los datos requeridos <p><i>Mientras el sistema no valida los datos:</i></p> <ol style="list-style-type: none"> 3.1.El sistema valida los datos 4. El sistema registra el grupo
Postcondiciones	El grupo de clase se ha creado
Flujo Alternativo	3.2. El sistema no valida los datos

Tabla 9. Especificación – Crear Grupo

Caso de uso: Gestionar Grupo	
Id	8
Descripción	El profesor gestiona su grupo
Actores principales	Profesor
Actores secundarios	Ninguno
Precondiciones	El profesor debe haber iniciado sesión y grupo debe estar creado
Flujo principal	<ol style="list-style-type: none"> 1. El profesor selecciona la opción “Gestionar” 2. El sistema pide al profesor que elija el grupo 3. El profesor selecciona el grupo y accede a este 4. El sistema muestra los alumnos del grupo <i>extension point (Visualizar Resultados)</i>
Postcondiciones	El profesor gestiona su grupo de clase
Flujo Alternativo	Ninguno

Tabla 10. Especificación – Gestionar Grupo

Caso de uso: Visualizar Resultados	
Id	9
Descripción	El sistema muestra las puntuaciones pertenecientes a cada uno de los juegos jugados por el alumno
Actores principales	Alumno/Profesor
Actores secundarios	Ninguno
Precondiciones	El alumno/profesor ha iniciado sesión y el alumno ha jugado al menos un nivel del juego
Flujo principal	<ol style="list-style-type: none"> 1. El alumno/profesor selecciona la opción “Puntuaciones” 2. El sistema comprueba que el alumno ha jugado al menos un nivel del juego 3. El sistema muestra las puntuaciones
Postcondiciones	El sistema debe mostrar las puntuaciones del alumno de cada juego
Flujo Alternativo	Ninguno

Tabla 11. Especificación – Visualizar Resultados

Caso de uso: Visualizar Logros	
Id	10
Descripción	El sistema muestra los logros de cada alumno
Actores principales	Alumno
Actores secundarios	Ninguno
Precondiciones	El alumno ha iniciado sesión y haber jugado al menos una vez
Flujo principal	<ol style="list-style-type: none"> 1. El alumno selecciona la opción “Logros” 2. El sistema comprueba que el alumno ha jugado al menos una vez 3. El sistema muestra los logros obtenidos en base a sus puntuaciones
Postcondiciones	El alumno debe poder ver sus logros
Flujo Alternativo	Ninguno

Tabla 12. Especificación – Visualizar Logros

Caso de uso: Modificar Alumno	
Id	11
Descripción	Un alumno modifica sus datos
Actores principales	Alumno
Actores secundarios	Ninguno
Precondiciones	El alumno ha iniciado sesión
Flujo principal	<ol style="list-style-type: none"> 1. El alumno selecciona la opción “Modificar” 2. El sistema muestra sus datos 3. El alumno cambia los datos 4. El sistema valida los datos introducidos 5. El sistema muestra el perfil actualizado
Postcondiciones	El alumno debe haberse actualizado
Flujo Alternativo	Ninguno

Tabla 13. Especificación – Modificar Alumno

Caso de uso: Modificar Profesor	
Id	12
Descripción	Un profesor modifica sus datos
Actores principales	Profesor
Actores secundarios	Ninguno
Precondiciones	El profesor ha iniciado sesión
Flujo principal	<ol style="list-style-type: none"> 1. El profesor selecciona la opción “Modificar” 2. El sistema muestra sus datos 3. El profesor cambia los datos 4. El sistema valida los datos introducidos 5. El sistema muestra el perfil actualizado
Postcondiciones	El profesor debe haberse actualizado
Flujo Alternativo	Ninguno

Tabla 14. Especificación – Modificar Profesor

Caso de uso: Registrar Resultados	
Id	13
Descripción	Una vez finalizado el juego musical, se guardan las puntuaciones de cada alumno
Actores principales	Juego Musical
Actores secundarios	Ninguno
Precondiciones	Se debe haber completado al menos un nivel del juego
Flujo principal	<ol style="list-style-type: none"> 1. El juego musical finaliza la ejecución 2. El sistema pide al juego musical las puntuaciones obtenidas 3. El juego musical solicita al sistema el alumno que ha jugado 4. El juego musical envía al sistema las puntuaciones 5. El sistema carga las puntuaciones del alumno
Postcondiciones	Se obtiene las puntuaciones de cada alumno
Flujo Alternativo	Ninguno

Tabla 15. Especificación – Registrar Resultados

Caso de uso: Eliminar Usuarios Inactivos	
Id	14
Descripción	Se eliminarán aquellos alumnos/profesores que no entren en la plataforma por seis meses
Actores principales	Tiempo
Actores secundarios	Ninguno
Precondiciones	El alumno/profesor debe tener cuenta
Flujo principal	<ol style="list-style-type: none"> 1. El sistema comprueba el tiempo sin acceder a la plataforma del alumno/profesor <p><i>Mientras el sistema no valida el tiempo:</i></p> <ol style="list-style-type: none"> 1.1. El sistema confirma que el tiempo transcurrido es igual o superior a seis meses 2. El sistema elimina de la base de datos al alumno/profesor que ha excedido el tiempo
Postcondiciones	El alumno/profesor debe quedar eliminado
Flujo Alternativo	1.2. El sistema no valida el tiempo

Tabla 16.: Especificación – Eliminar Usuarios Inactivos

8.3. Validación de casos de uso

A continuación, se comprobará que el sistema obtiene la funcionalidad esperada. Para ello, utilizaremos la matriz de trazabilidad que se observa a continuación, dónde se relacionan los casos de uso especificados con los requisitos funcionales ya especificados anteriormente.

RF/CU	1	2	3	4	5	6	7	8	9	10	11	12	13	14
RF-01	X													
RF-02		X												
RF-03	X	X				X								
RF-04						X								
RF-05						X								
RF-06						X								
RF-07											X			
RF-08												X		
RF-09							X							
RF-10								X						
RF-11								X						
RF-12									X					
RF-13									X	X			X	
RF-14									X					
RF-15										X				
RF-16			X											
RF-17					X									
RF-18				X										
RF-19														X
RF-20														X
RF-21			X											

Tabla 17. Matriz de Trazabilidad

9. DISEÑO DEL SISTEMA

En este apartado, se definirá la arquitectura del sistema y la especificación de cada uno de sus componentes.

Esta fase, está basada en la información detallada del anterior punto (análisis del sistema), cuyo fin es cumplir todos los objetivos propuestos.

9.1. Arquitectura del sistema

La definición de arquitectura de un sistema nos indica que es el diseño e implementación de este, o lo que es lo mismo, es el resultado de enlazar algunos elementos arquitectónicos de una forma concreta para que de esta forma cumplan con la funcionalidad y los requisitos del sistema.

Nuestra plataforma web está desarrollada bajo el patrón de arquitectura llamado Modelo-Vista-Controlador (MVC) [14], citado en anteriores apartados.

También se ha citado con anterioridad el framework que usaremos para el desarrollo de la plataforma web, Flask. Este framework o “microframework” puesto que sólo instala las librerías que necesita para funcionar, está escrito en Python y basado en la especificación de Werkzeug y el motor de templates Jinja2.

Flask es compatible con todos los lenguajes de web, HTML5, CSS, JavaScript entre otros y sigue el patrón de arquitectura nombrado anteriormente. Dicho patrón, separa la lógica, las interfaces de usuario y los datos:

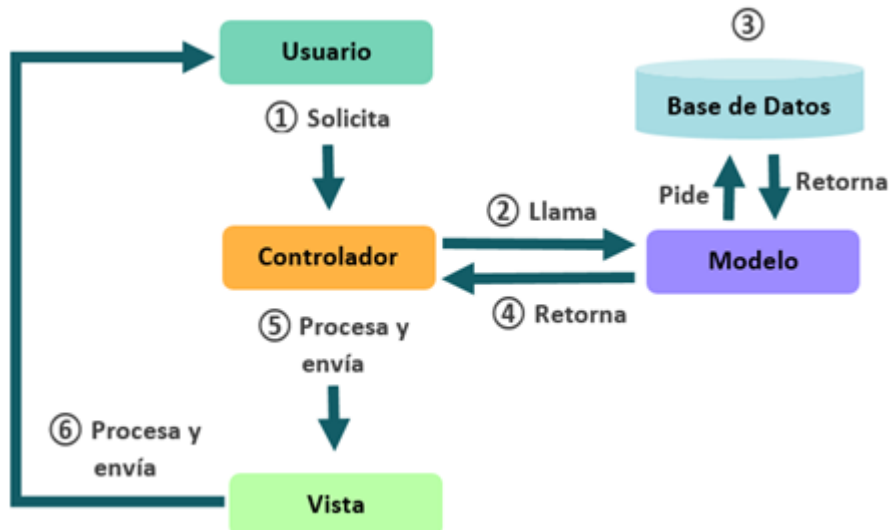


Figura 9. Estructura MVC

Como se observa aparecen los tres bloques definidos como:

- ✓ **Modelo**
Representa los datos con los que opera el sistema
- ✓ **Vista**
Representa los datos del sistema con los que trabaja el modelo, para interactuar con ellos
- ✓ **Controlador**
Controla y responde a las acciones de usuario, procesando y enviando datos a Vista

9.2. Estructura de ficheros

En este apartado se muestra y explica la estructura de archivos y ficheros de nuestro proyecto creado con Flask, dónde se ubican todo lo necesario para nuestro proyecto. A continuación, se describen todos los elementos de nuestra estructura de ficheros.

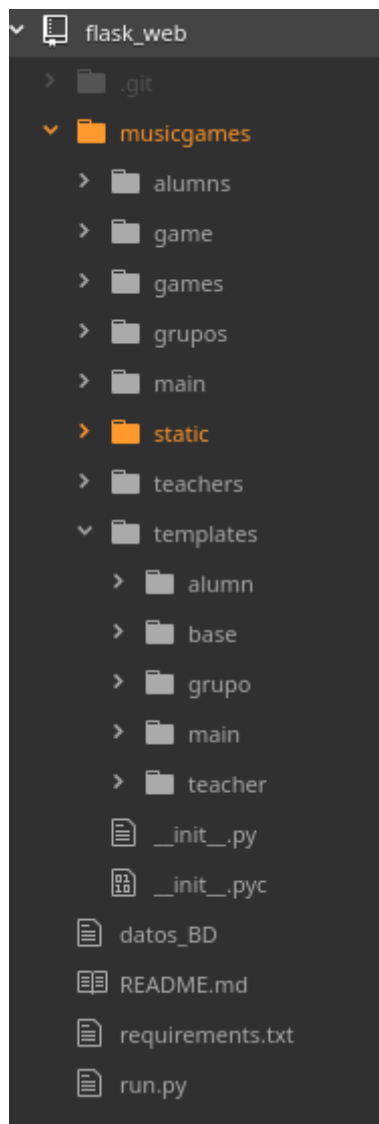


Figura 10. Estructura del proyecto Plataforma Web

En un proyecto Flask, el archivo “requirements.txt”, indica al desarrollador las dependencias Python que debe instalar para que el proyecto funcione correctamente. El archivo “run.py” es el encargado de ejecutar la aplicación Flask.

El fichero [musicgames] contiene la aplicación Flask en sí. Y dentro de éste:

- ✓ [templates]
Contiene las plantillas web de nuestra aplicación Flask, divididas en [alumn], [base], [grupo], [main], [teacher].
- ✓ [static]
Contiene los archivos y ficheros estáticos de nuestra aplicación:
 - [css]

- Contiene los archivos de estilos y de responsive.
- [img]
Contiene los archivos imagen usados.
- [js]
Contiene los archivos JavaScript usados
- ✓ [alumnos], [grupos], [games], [main], [teachers]
Distintos módulos de la aplicación Flask.
- “__init__.py”
Sirve para inicializar cada módulo como paquete de Python.
- “views.py”
Contiene todas las vistas accesibles.
- “__init__.py”
Conecta con la base de datos (MySQL), crea la aplicación Flask y registra los objetos alumnos, games, grupos, main y teachers.

9.3. Diseño Procedimental

En este apartado se especificará la estructura estática a través del diagrama de clases en el que se especifican distintas clases del sistema, operaciones y atributos.

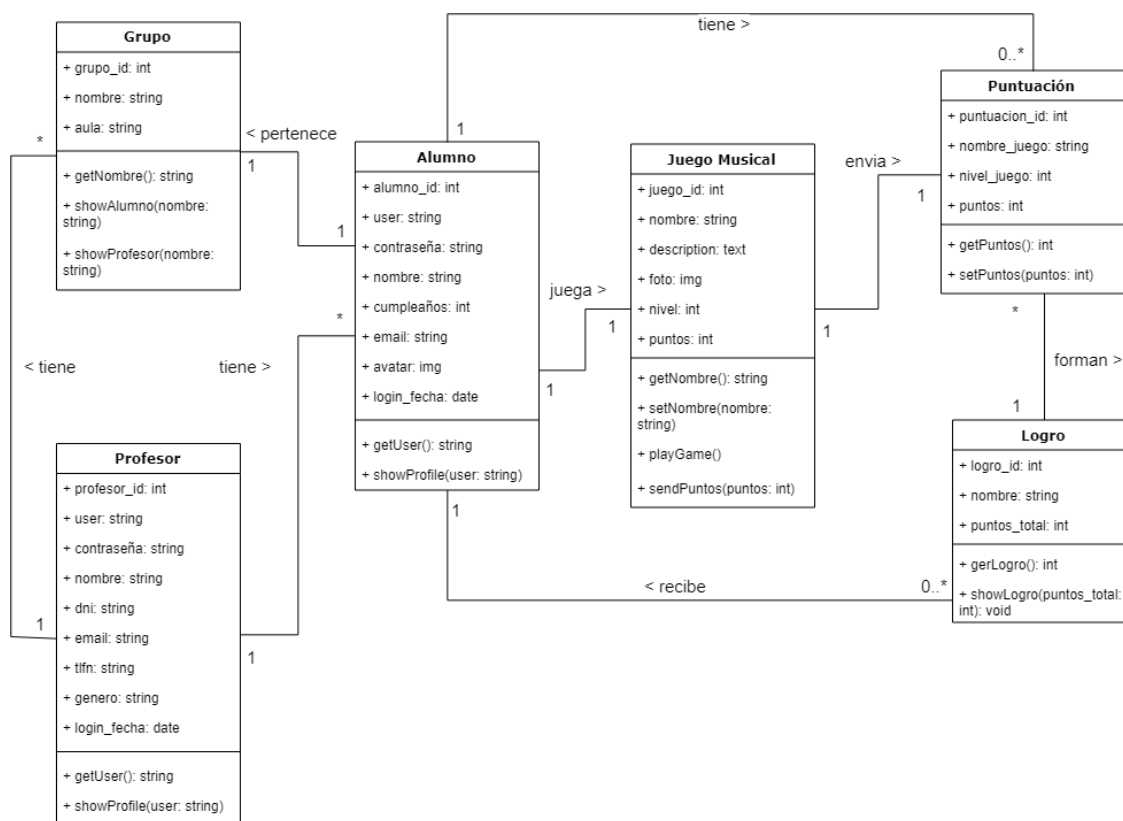


Figura 11. Diagrama de clases – Plataforma Web

Como podemos observar, la mayoría de las relaciones entre una tabla y otra son de 1 a muchos, salvo en tres casos que son de 1 a 1. Estos casos son por ej., un alumno pertenece a un grupo, se ha puesto esta restricción puesto que se trata de una academia, y queremos que el alumno solo pueda estar en un grupo.

En cambio, si decimos que un profesor puede tener varios grupos, y por tanto varios alumnos, es por eso que en estos casos usamos la restricción 1 a muchos.

Se muestra también, el diagrama de secuencia, detallando cómo se comunica la plataforma web con los juegos, a la hora de recibir las puntuaciones.

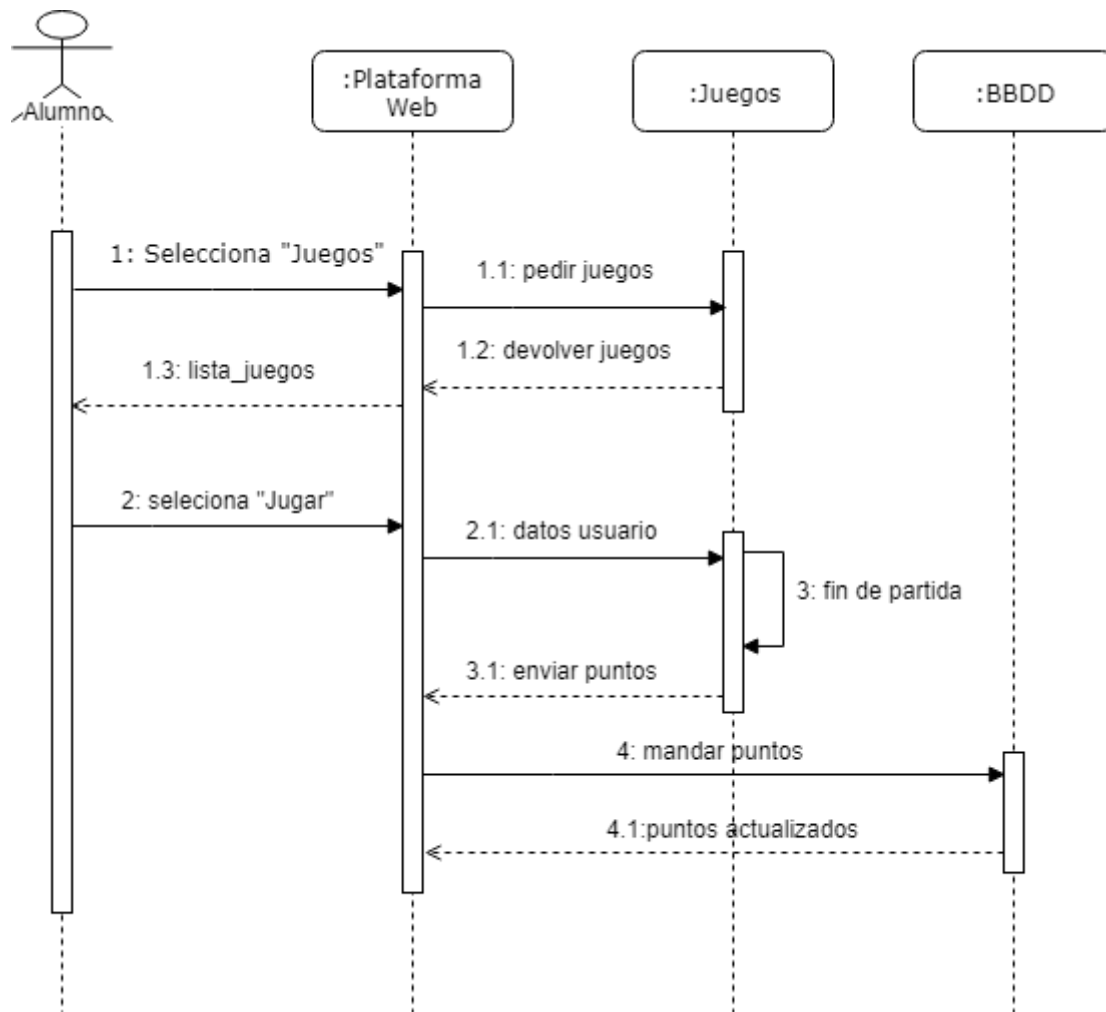


Figura 12. Diagrama de secuencia – Recibir Puntuaciones (API)

9.4. Estructura Base de Datos

En este apartado se mostrará la estructura de nuestra base de datos (BD), que como se ha citado anteriormente, trabajará con el modelo relacional y bajo el sistema de gestión, MySQL.

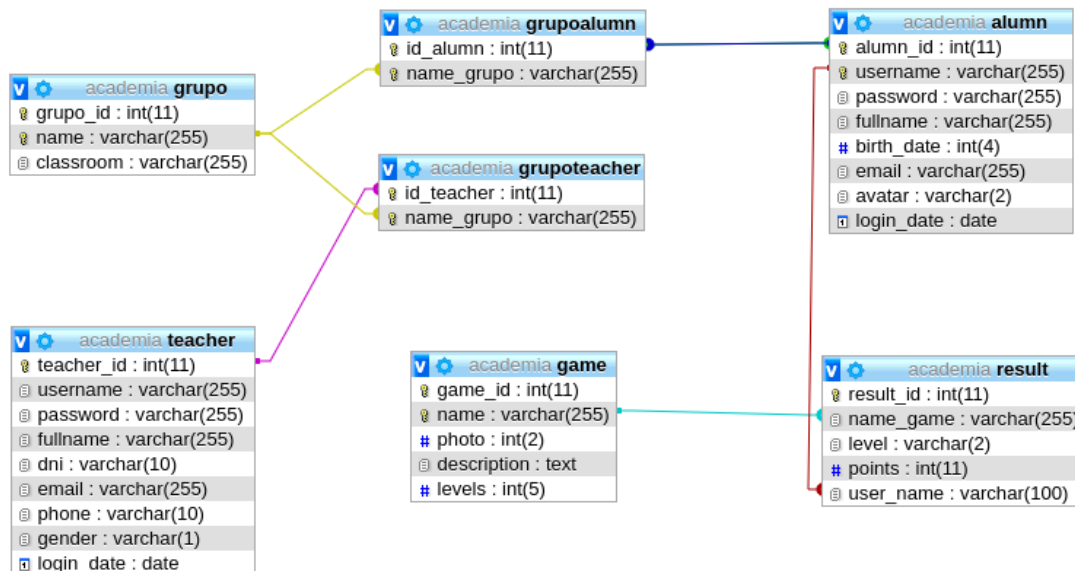


Figura 13. Base de Datos Relacional (SGBD) de nuestro proyecto – Plataforma Web

Como se observa en la Figura 13, disponemos de un SGBD, el cuál en nuestro proyecto es llamado “Academia”, con siete tablas, las cuáles recogen todos los requisitos de información mencionados anteriormente:

- ✓ Tabla alumn
Recoge toda la información del alumno, y es a partir de esta, de la cual podemos saber si un alumno pertenece a algún grupo y, por tanto, dispone de profesor; si ha jugado a un juego, a cuál ha sido y su puntuación.
- ✓ Tabla teacher
Recoge toda la información del profesor, y a través de esta, podemos saber que grupos gestiona y que alumnos están en cada uno de ellos, así como sus puntuaciones
- ✓ Tabla grupo
Recoge toda la información del grupo, y sirve como relación entre las Tablas alumn y teacher
- ✓ Tabla grupoalumn (relación muchos a muchos)
Relaciona alumno con grupo
- ✓ Tabla grupoteacher (relación muchos a muchos)
Relaciona profesor con grupo
- ✓ Tabla game
Recoge toda la información del juego, y se encuentra conectada indirectamente, con la Tabla result
- ✓ Tabla result
Recoge todas las puntuaciones en función del juego y el alumno que juega a dicho juego

Gracias a las Tablas grupoalumn y grupoteacher, podemos realizar consultas con mayor rendimiento para nuestras vistas y así poder cumplir con todos los objetivos y subobjetivos propuestos.

9.5. Interfaz de usuario

Se entiende por interfaz de usuario, el medio por el cual un usuario puede interactuar con el dispositivo, se describirá la interfaz de usuario de la plataforma web.

A continuación, se explica el diseño y funcionamiento de las principales pantallas de la plataforma web.

Puesto que la plataforma web, se le dará mayor uso en dispositivos móviles, la plataforma debe ser “responsive”, para ello se ha utilizado la propiedad `@media only screen` y la de `min/max-device` en el archivo “resp.css”. Estas propiedades hacen que nuestra pantalla de la plataforma web se adapte a la perfección del dispositivo móvil en concreto. Dando como soporte a todos los dispositivos móviles del mercado.

A continuación, se muestra un ejemplo, para hacer responsive la pantalla de un iPhone Plus:

```
@media only screen
and (min-device-width : 414px)
and (max-device-width : 736px) { /* iPhone Plus */
    #navegador {
        float: left;
        width: 25%;
    }
}
```

9.5.1. Diseño de la pantalla inicial

En esta pantalla el usuario tiene tres opciones posibles, “Usuarios No Registrados”, “Alumnado” y “Profesorado”.

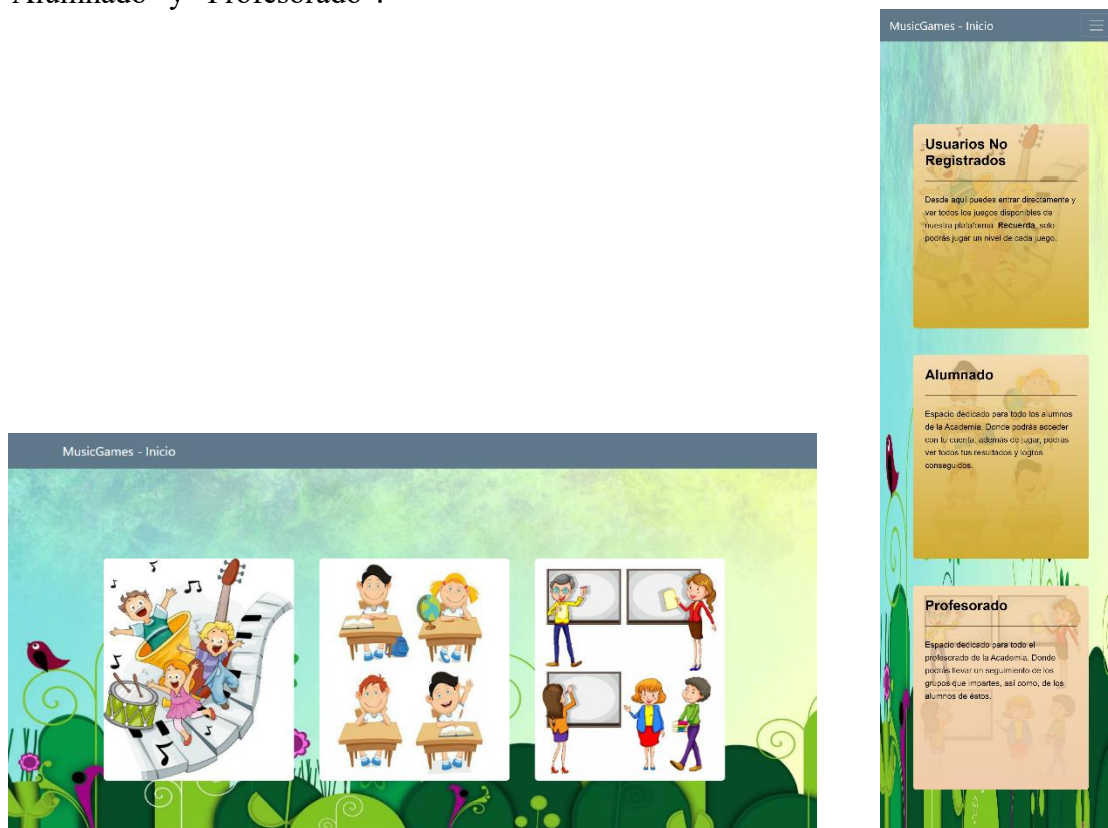


Figura 14. Interfaz de Usuario – Pantalla Inicial – Normal y Responsive

Cuando se está en dispositivos móviles las tres imágenes permanecen fijas por la parte de atrás, en cambio, cuando se está en un ordenador las imágenes voltean mostrando la información en la parte trasera. Cuando se cliquea en cada imagen, se redirige a cada sesión.

9.5.2. Diseño de las pantallas de usuarios no registrados/alumnos

Ambas sesiones siguen una interfaz clara y concisa, y específicas para un público infantil, con fondo animado, avatares en los perfiles y botones de colores, entre otros.

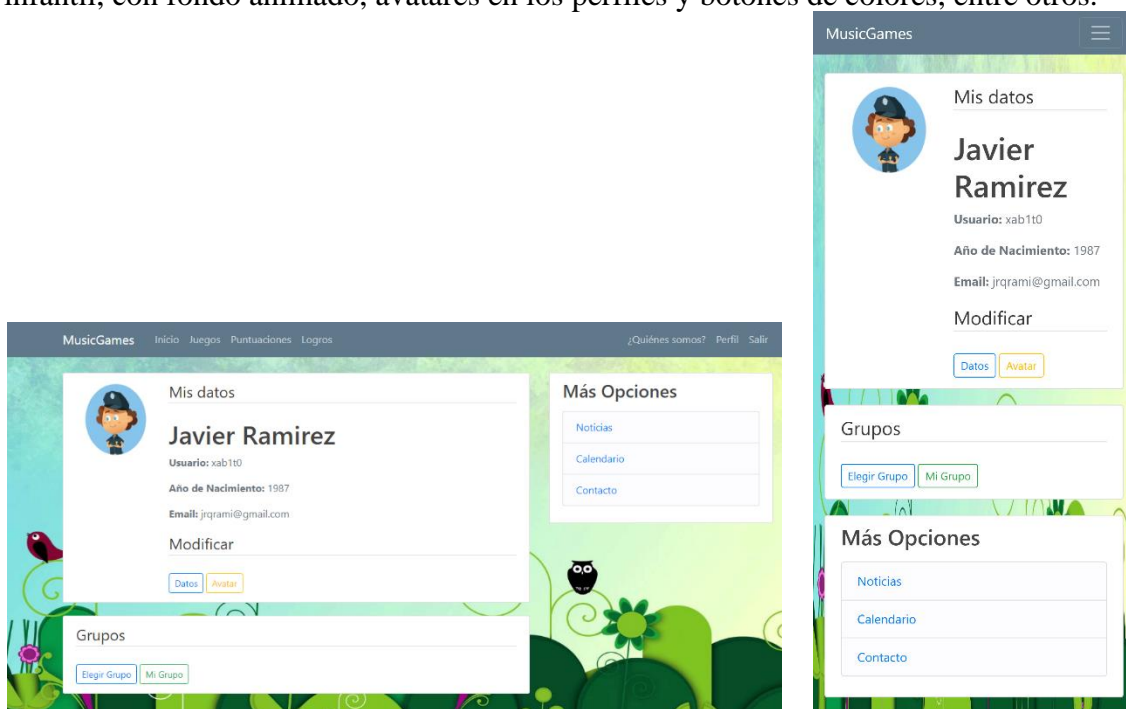


Figura 15. Interfaz de usuario – Perfil Alumnado – Normal y Responsive

9.5.3. Diseño de las pantallas del profesorado

El diseño de esta sesión es más limpio ya que no tiene fondo de pantalla, solo dos tipos de avatar para diferenciar entre profesor y profesora, entre otros.

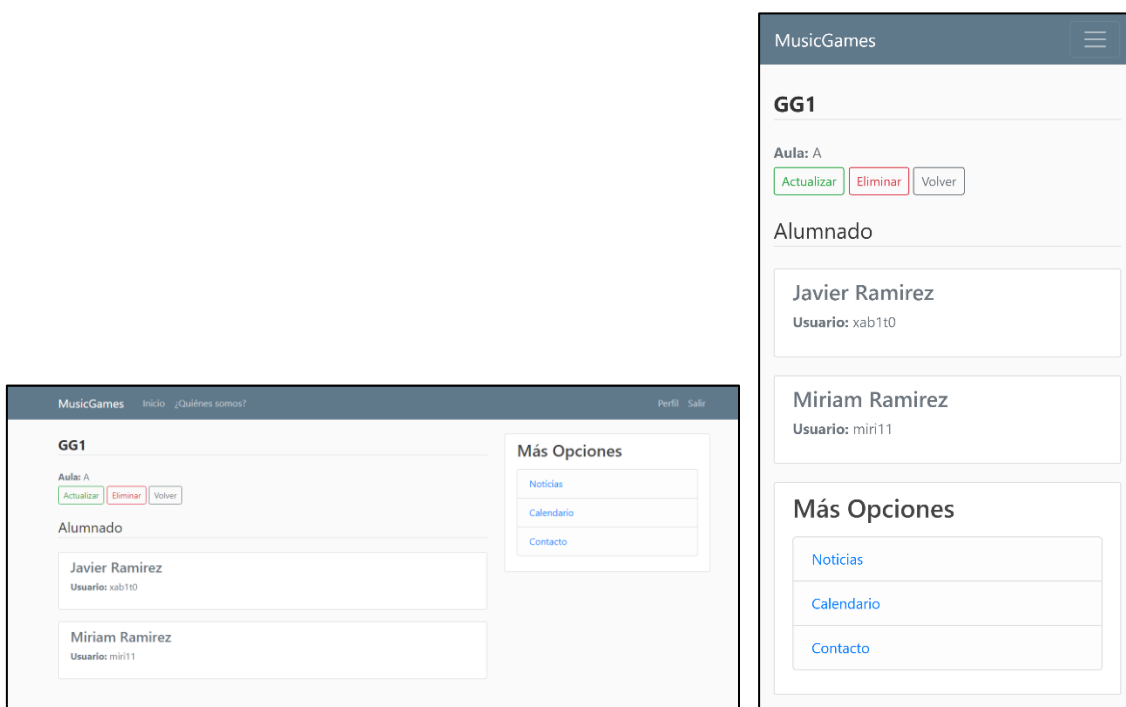


Figura 16. Interfaz de usuario – Gestión de grupo – Normal y Responsive

10. PRUEBAS

Una vez llegado a esta fase, se comprobará la calidad y el funcionamiento de la plataforma web, de esta forma, certificaremos que el software cumple con los requisitos propuesto en un inicio y cómo a través de recomendaciones sugeridas por compañeros de la materia, estos requisitos se han actualizado o incrementado.

La duración de esta fase es combinada con la fase de desarrollo, en la que es necesario realizar pruebas contantemente para garantizar la integridad del sistema desarrollado.

10.1. Pruebas realizadas

A continuación, se describen las pruebas realizadas sobre la plataforma web, describiendo los problemas encontrados, si los hubo y las soluciones adoptadas para solventar dichos problemas.

En la mayoría de los problemas encontrados, se volvió al proceso de desarrollo, sin embargo, en algunos casos fue necesario volver a fases anteriores.

Prueba 1	
Descripción	El usuario no introduce contraseña en acceso
Problemas encontrados	Permite al usuario acceder a su sesión
Soluciones adoptadas	La etiqueta password del form no estaba escrita correctamente, se modifica

Tabla 18. Prueba 1

Prueba 2	
Descripción	El usuario introduce mal los datos de acceso
Problemas encontrados	Ninguno
Soluciones adoptadas	Ninguno

Tabla 19. Prueba 2

Prueba 3	
Descripción	El usuario no registrado escribe la ruta de perfil de un usuario
Problemas encontrados	Permite al usuario no registrado acceder al perfil
Soluciones adoptadas	Se hace un control de “session” en todas las rutas que son privadas para usuarios registrados

Tabla 20. Prueba 3

Prueba 4	
Descripción	La API no recibe las puntuaciones correctamente
Problemas encontrados	Se encuentra una variable mal
Soluciones adoptadas	Se modifica dicha variable

Tabla 21. Prueba 4

Prueba 5	
Descripción	El profesor crea un grupo
Problemas encontrados	El grupo no aparece en gestionar
Soluciones adoptadas	Se crea nueva tabla relacional grupoteacher, la cual relaciona el teacher_id con el grupo_id, visualizando así el grupo

Tabla 22. Prueba 5

Prueba 6	
Descripción	El alumno elige un grupo
Problemas encontrados	El grupo no aparece en mi grupo
Soluciones adoptadas	Se crea nueva tabla relacional grupoalumn, la cual relaciona el alumn_id con el grupo_id, visualizando así el grupo

Tabla 23. Prueba 6

10.2. Sugerencias externas

En este apartado se han tenido en cuenta ciertas sugerencias recibidas por compañeros de clase, con el fin de mejorar y/o completar la plataforma web.

En un principio las interfaces de usuario de la plataforma web para las sesiones “Usuarios No Registrados” y “Alumnado”, tenían un aspecto más serio como la sesión del “Profesorado”, por lo que no iba acorde con el público dirigido. Por lo que se optó por dar colores vivos a los botones y poner fondo con temática infantil, permitiendo que la información se mostrará de manera correcta en todo instante.

Respecto a la elección de avatar (modo alumno), en un principio se hacía por medio de menú desplegable, lo cual daba menos visibilidad a la página y hacía que este proceso fuese más engorroso. Se propuso eliminar este menú y que el alumno cuando seleccionará un avatar, éste se marcará sobre los demás. Para implementar esta sugerencia, se añadieron marcos a todos los avatares, y se modificó el form para que al seleccionar un avatar se seleccionara y guardara como un valor concreto en el form.

Para que el profesorado pudiera acceder a un registro y acceso propio, y el resto de los usuarios no pudiera acceder, se pensó en poner una contraseña al final del form de registro, pero esto conllevaba problemas y que el acceso fuese más inseguro. Se optó por crear una página la cual solicitara esta contraseña y si era correcta redirigiese al registro o acceso del profesor.

11. CONCLUSIONES Y FUTURAS MEJORAS

En esta fase del proyecto se expondrán las conclusiones obtenidas tras finalizar el proceso de desarrollo de la plataforma web, así como posibles mejoras y ampliaciones.

Teniendo en cuenta los objetivos establecidos en el proyecto y detallados en el capítulo 3, se establecen las siguientes conclusiones respecto al resultado de dichos objetivos:

- ✓ El diseño de la plataforma web permite que sea visualizada en varios dispositivos con diferentes tamaños de pantalla o lo que es lo mismo, por diseño responsive
- ✓ Se ha conseguido integrar diferentes juegos con distintas formas de puntuar y lanzar niveles, en la plataforma web
- ✓ Se ha logrado diferenciar a los distintos usuarios, los más destacados alumnos y profesores
- ✓ Se ha conseguido una plataforma web con una interfaz intuitiva y adaptada tanto para un público infantil, así como para un público más profesional
- ✓ Se ha permitido poder jugar a los usuarios sin registro previo, pero sólo jugando el primer nivel de cada juego, si los tuviera
- ✓ Se ha conseguido un registro y login de usuarios, muy cómodo y sencillo
- ✓ Se ha logrado premiar a los alumnos con mayor puntuación en total, a través de logros

11.1. Trabajo futuro

Para finalizar, en este apartado se comentarán los aspectos mejorables de la plataforma y de futuras ampliaciones, a pesar de haber cumplido a la perfección los objetivos propuestos al inicio de este proyecto. A continuación, se explican una serie de mejoras:

- ✓ Desarrollar la posibilidad de compartir las puntuaciones y/o logros obtenidos, por redes sociales
- ✓ Desarrollar logros que vayan enfocados a cada juego, al número de veces que ha jugado, al tiempo que ha jugado, ...
- ✓ Ofrecer al usuario la posibilidad de disponer de foto propia en perfil, en vez de sólo un avatar

BIBLIOGRAFÍA

- [1] Educación Musical, web consultada el 2019-03-02:
<http://www.educamus.es/index.php/educacion-musical>
- [2] Apuntes de la asignatura “Proyectos”, “Tema 2 – Estructura Documental del Proyecto de Informática”, Escuela Politécnica Superior de Córdoba
- [3] W3C HTML, web consultada el 2019-05-02:
<https://www.w3.org/html/>
- [4] CSS Snapshot 2018, web consultada el 2019-05-10:
<https://www.w3.org/TR/CSS/>
- [5] The W3C Markup Validation Service, web consultada el 2019-07-29:
<https://validator.w3.org>
- [6] ¿Qué es el diseño responsive?, web consultada el 2019-07-29:
<https://www.40defiebre.com/que-es/disenio-responsive>
- [7] ¡Vamos a la Ópera! - AppTK, web consultada el 2019-07-31:
<https://apptk.es/apps/vamos-a-la-opera>
- [8] Toc and Roll – Minimúsica, música moderna para toda la familia, web consultada 2019-07-31:
<http://minimusica.tv/toc-and-roll/>
- [9] Little Musician: An App for Instrument, web consultada el 2019-07-31:
<https://appreview.com/app/little-musician-musical-instruments/>
- [10] Juegos de música | Juegos.com, web consultada el 2019-07-31:
<https://www.juegos.com/juegos/juegos-de-musica>
- [11] Juegos de música - Minijuegos.com, web consultada el 2019-07-31:
<https://www.minijuegos.com/juegos-de-musica>
- [12] MySQL- Wikipedia, la enciclopedia libre, web consultada el 2019-05-27:
<https://es.wikipedia.org/wiki/MySQL>
- [13] Welcome | Flask (A Python Microframework), web consultada el 2019-05-20:
<http://flask.pocoo.org>
- [14] Apuntes de la asignatura “Modelado y Diseño Avanzado de Software”, “Bloque I – Tema 2 – Patrones Arquitectónicos”, Escuela Politécnica Superior de Córdoba