

IES Laguna de Joatzel Grado Superior de Desarrollo de
Aplicaciones Web

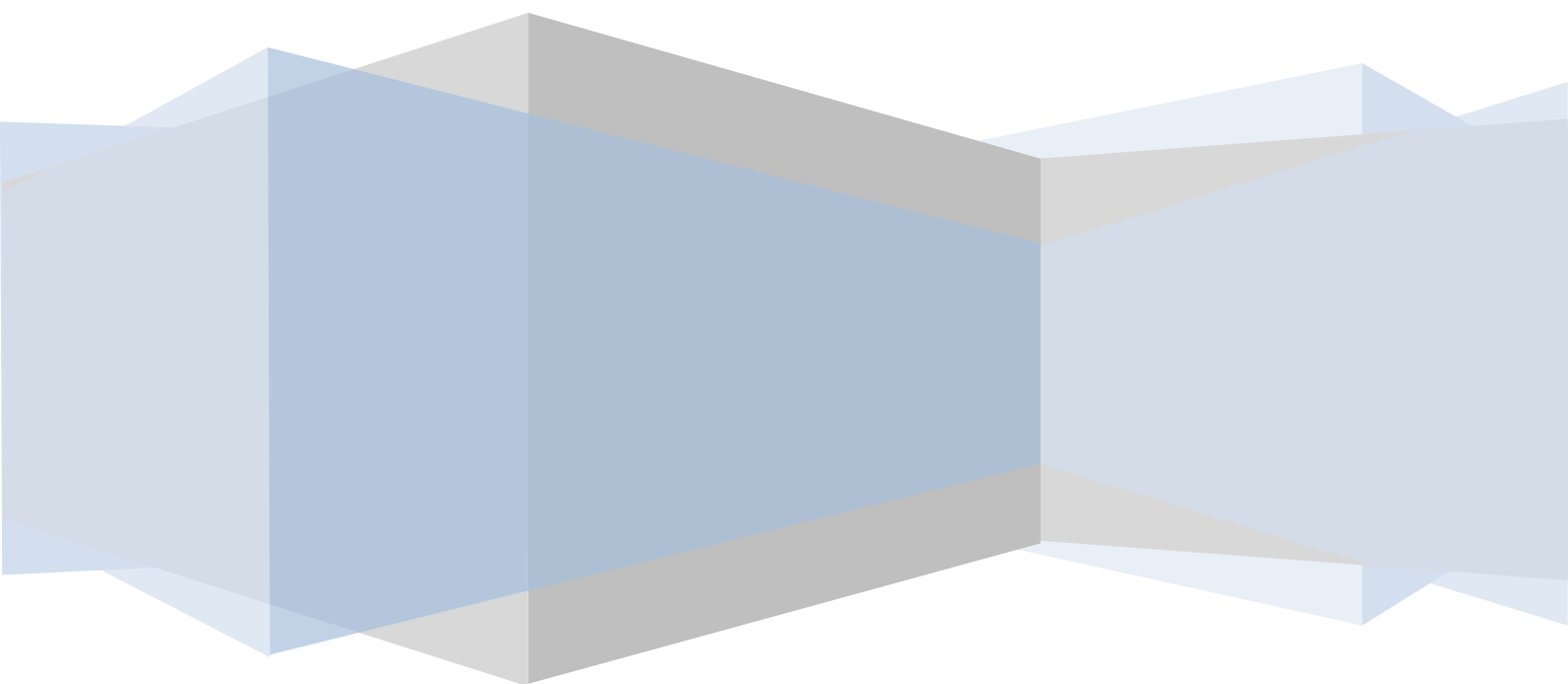
Trabajo Fin de Grado

Tienda de Videojuegos Online

Iván De Las Heras Martín | José Galea Torrecilla

Tutor: Guillermo Montaner Andrés

junio de 2022



Índice

Contenido

ABSTRACT	4
PALABRAS CLAVE.....	5
GLOSARIO	6
1 INTRODUCCIÓN	7
1.1 OBJETIVOS DEL PROYECTO	7
1.2 JUSTIFICACIÓN.....	8
2 ANÁLISIS.....	9
2.1 CATÁLOGO DE REQUISITOS.....	9
2.1.1 REQUISITOS FUNCIONALES	9
2.1.2 REQUISITOS NO FUNCIONALES	10
2.2 PLANIFICACIÓN PREVIA.....	11
2.3 FASES DE REALIZACIÓN	12
3 DISEÑO	13
3.1 TECNOLOGÍAS UTILIZADAS.....	13
3.1.1 HTML	13
3.1.2 CSS	14
3.1.3 PHP	15
3.1.4 JavaScript.....	15
3.1.5 PHPStorm	16
3.1.6 XAMPP	17
3.1.7 GitHub	17
3.2 DISEÑO DEL PROYECTO	18
3.2.1 Diagramas de Casos de Uso	18
3.2.1.1 Inicio de Sesión.....	18
3.2.1.2Menú de Juegos	19
3.2.1.3Mi Cuenta	19
3.2.1.4Mis Juegos	20
3.2.1.5Ficha de Juego	20
3.2.2 Diagrama de Flujo del proyecto	21
3.2.3 Modelo de datos	22

3.3 DISEÑO	23
3.3.1 Menú de Inicio de Sesión	23
3.3.1 Contraseña Olvidada	24
3.3.2 Catálogo de Videojuegos.....	24
3.3.3 Biblioteca de Mis Juegos	26
3.3.4 Ficha de Videojuegos.....	27
3.3.5 Mi Cuenta	28
3.3.6 Acerca De	29
3.3.7 Comprar Puntos	29
4 IMPLEMENTACIÓN	30
- Inicio de sesión y creación de nuevas cuentas.....	30
- Cerrar sesión	34
- Nueva Contraseña (PHP)	35
- Generar Nueva Contraseña (PHP)	35
- Enviar Correo Contraseña (PHP)	36
- Correo Contraseña (PHP)	38
- Validación Formulario Contraseña (JS)	39
- Validación del Código de usuario (JS).....	41
- Confirmación Contraseña (JS)	41
- Gestión Menú Principal (JS)	42
- Gestión de los videojuegos del usuario: compra	46
- Gestión de los videojuegos del usuario: eliminación	48
- Ajustes de usuario	51
- Página Oculta	55
- Modo Claro/Oscuro.....	57
5 PRUEBAS	59
6 TRABAJO FUTURO	61
7 CONCLUSIONES	62
8 ANEXO.....	63
8.1 CONFIGURACIÓN Y DESPLIEGUE	63
9 BIBLIOGRAFÍA.....	65

ABSTRACT

Este proyecto consiste en el análisis, diseño y desarrollo de una aplicación web que realice las funciones de una tienda de videojuegos. El objetivo es crear una tienda totalmente funcional en la cual los usuarios puedan obtener y jugar los videojuegos que deseen.

La aplicación está dirigida a todo tipo de usuarios, los únicos requisitos son que el usuario posea conexión a internet, un ordenador y un correo electrónico.

El sistema se divide en varios subsistemas:

Gestión de usuarios y sus cuentas, gestión de videojuegos, gestión y la comunicación con la base de datos.

Nuestra aplicación intenta abarcar casi todas las funcionalidades necesarias para que sea una tienda totalmente operativa.

Al estar en auge las tiendas online, la elección del equipo ha sido desarrollar una tienda web, debido al bajo costo de mantenimiento, la disponibilidad, la posibilidad de poder conocer mejor al cliente y el control y gestión de la tienda.

El Backend de la aplicación web está desarrollado en PHP, AJAX y MySQL.

El frontend de la aplicación web está desarrollado en HTML, CSS y JavaScript.

PALABRAS CLAVE

La siguiente lista contiene palabras descriptivas que podrán utilizarse para catalogar el proyecto.

- Php
- JavaScript
- HTML
- CSS
- Aplicación Web
- MySQL
- AJAX
- JQuery
- Cookies
- Sesiones

GLOSARIO

- **Hosting:** (Anglicismo), es un servicio de alojamiento en línea que permite publicar un sitio o aplicación web en Internet.
- **AJAX:** es una técnica de desarrollo web para crear aplicaciones web asíncronas. Estas aplicaciones se ejecutan en el cliente mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.
- **Comunicación Asíncrona:** Es una transmisión de datos la cual proviene de un intercambio intermitente de datos y no de un flujo constante.
- **Cookies:** (Anglicismo), es una pequeña información enviada por un sitio web y almacenado en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del navegador.
- **Script:** (Anglicismo), es un término informal para un documento que contiene instrucciones escritas en códigos de programación.
- **Sesión:** Es un intercambio de información semipermanente entre dos o más dispositivos de comunicación o entre un ordenador y un usuario.
- **Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
- **DAO: (Data Access Object),** es una clase que interactúa con la base de datos en la que se implementan métodos **CRUD**, un acrónimo en inglés de “Crear, Leer Actualizar y Borrar” y suministra una interfaz común entre la aplicación y dispositivos de almacenamiento.

1 | INTRODUCCIÓN

1.1 OBJETIVOS DEL PROYECTO

Los objetivos de un proyecto son el conjunto de resultados que se espera obtener tras concluir con el desarrollo activo del mismo.

El objetivo principal de este proyecto es el de desarrollar una plataforma web en la que los usuarios puedan acceder a un conjunto de videojuegos. La plataforma debe presentar una interfaz gráfica sencilla e intuitiva que invite al uso por parte del usuario; así mismo, cada usuario podrá acceder a los videojuegos que haya adquirido siempre que quiera, y podrá adquirir nuevos y eliminar los ya presente siempre que estime oportuno. Se implementará una serie de opciones para facilitar la búsqueda de videojuegos por parte del usuario (búsqueda por categoría, por nombre, etc).

Además, la plataforma dispondrá de un menú de navegación cómodo y efectivo mediante el cual los usuarios podrán navegar por los distintos apartados de la misma sin desubicarse.

Para garantizar la seguridad de los datos de los usuarios y el correcto uso de los mismos, la plataforma presentará un sistema de inicio de sesión que solicitará unas credenciales (usuario y contraseña) a todo aquel que intente acceder al contenido de la plataforma. Así mismo, se facilitarán distintas herramientas para facilitar la gestión de las credenciales por parte de los usuarios: la interfaz del inicio de sesión presentará un apartado para la creación de nuevas cuentas así como la posibilidad de recuperar la contraseña en caso de que se haya extraviado.

En conclusión, la finalidad de este proyecto es desarrollar una completa plataforma web en la que los usuarios puedan interactuar con los distintos elementos de la misma permitiéndoles realizar acciones como iniciar sesión, ver los distintos videojuegos, adquirir y eliminar los que se consideren y, en última instancia, jugar a todos aquellos videojuegos que se hayan adquirido de forma cómoda y sin necesidad de realizar ningún tipo de instalación.

1.2 JUSTIFICACIÓN

La finalidad de este apartado consiste en explicar, de forma argumentada, las razones que motivan a la realización de este proyecto; intentando responder a las preguntas “¿por qué?” y “¿para qué?” se lleva a cabo este proyecto.

A continuación, se procede a detallar dichas razones:

El uso de plataformas web ha crecido exponencialmente en los últimos años, cada vez son más los usuarios que prescinden de realizar tediosas instalaciones para poder disfrutar de los servicios de una aplicación. En su lugar, el mercado se está decantando por el uso de espacios en Internet que permiten la ejecución de diversas aplicaciones o programas en un mismo lugar. De esta forma, se aspira a posicionar a la web como principal canal para adquirir contenidos.

De hecho, las plataformas web proporcionan una mayor libertad de uso a los usuarios ya que pueden acceder a su contenido desde múltiples dispositivos. Si bien es cierto que la experiencia del usuario puede estar algo ligada a la calidad de la conexión a la red que presente, el poder conectarse desde cualquier lugar del mundo sitúa a las plataformas web por encima de las aplicaciones de escritorio en lo que a usabilidad y accesibilidad se refiere.

Nuestra intención es la de implementar un software que permita a los usuarios disfrutar de un conjunto de videojuegos en un entorno cómodo y accesible; proporcionándoles un amplio abanico de opciones.

Por tanto, hemos decidido que la mejor manera de llevar a cabo dicho propósito es desarrollando una plataforma web de videojuegos como proyecto.

2 | ANÁLISIS

En este capítulo se presenta la fase de análisis, parte inicial de todo proyecto software, se definen los requisitos y se muestra una vista global de la arquitectura pensada para el sistema. En este capítulo se tendrá en cuenta este catálogo de requisitos como base para el diseño de todos los aspectos de la plataforma web, así como la planificación y sus fases.

Es por ello por lo que la fase de análisis es de suma importancia para las siguientes fases de todo el producto software. Es en este capítulo donde se deben asentar las bases, a modo de cimientos, del proyecto, y, a partir de las cuales se construirá todo lo demás. El proceso a seguir se basa en primer lugar en una buena definición de requisitos y en elegir una metodología de desarrollo acorde con el proyecto. La arquitectura de la plataforma web estará basada en un conjunto de páginas web a las que el usuario podrá acceder a través de un menú de navegación presente en todas ellas.

A continuación, se desglosan la funcionalidad y las características a modo de catálogo de requisitos, teniendo en cuenta tanto requisitos funcionales como no funcionales.

2.1 CATÁLOGO DE REQUISITOS

El catálogo de requisitos es la especificación del comportamiento que se espera de cualquier proyecto software. Estudiando otras aplicaciones similares, se ha predefinido una serie de requisitos que se consideran indispensables para el proyecto. A continuación, se muestra una enumeración y breve descripción de los requisitos establecidos para el diseño y desarrollo de la plataforma web.

2.1.1 REQUISITOS FUNCIONALES

En este apartado se especifica cuáles son las tareas principales que debe realizar nuestro proyecto software.

En esta tabla se detallan los siguientes requisitos funcionales:

ID	Requisito	Descripción
RF-01	Inicio de sesión	Inicio de sesión del usuario al introducir el su identificador y su contraseña.
RF-02	Menú desplegable	Barra lateral izquierda con menú desplegable para la navegación del usuario por la plataforma web.
RF-03	Cerrar Sesión	La plataforma web presentará la opción de cerrar sesión por parte del usuario.
RF-04	Fichas de videojuegos	Cada videojuego de la plataforma web tendrá su respectiva ficha técnica a la que los usuarios autorizados podrán acceder.
RF-05	Adquirir videojuegos	La plataforma web permitirá a los usuarios autorizados adquirir videojuegos.
RF-06	Eliminar videojuegos	La plataforma web permitirá a los usuarios autorizados eliminar de su biblioteca los videojuegos adquiridos que consideren oportunos.
RF-07	Búsqueda de videojuegos	La plataforma web presentará diversos métodos de búsqueda de videojuegos (por categoría, nombre, etc).
RF-08	Envío correo electrónico	Se enviará un correo electrónico a cada usuario registrado dándole la bienvenida.
RF-09	Creación nuevas cuentas	La plataforma web gestionará la creación de nuevas cuentas realizando las validaciones pertinentes.
RF-10	Ajustes de usuario	Los usuarios podrán modificar sus datos personales (nombre, apellidos, email, foto de perfil, etc) una vez creada la cuenta en la plataforma.
RF-11	Recordar contraseña usuario	La plataforma web permitirá modificar la contraseña y recuperar la cuenta en caso de olvidarla llevando a cabo una serie de validaciones de seguridad.
RF-12	Modo Claro/Oscuro	Los usuarios podrán configurar la apariencia de la plataforma web pudiendo elegir entre un modo claro y otro oscuro.

2.1.2 REQUISITOS NO FUNCIONALES

En este apartado se especifica cuáles son los requisitos complementarios o atributos de calidad de nuestro proyecto software.

Estos especifican criterios que juzgan operaciones del sistema en lugar de su comportamiento (requisitos funcionales).

En esta tabla se detallan los siguientes requisitos no funcionales:

ID	Requisito	Descripción
RNF-01	Interfaz y usabilidad	La plataforma web debe constar de interfaces intuitivas y sencillas, que resulten atractivas a los usuarios.
RNF-02	Accesibilidad	La plataforma web debe ser accesible para distintos usuarios. El usuario debe saber en todo momento en que apartado de la plataforma se encuentra.

RNF-03	Páginas web responsivas	La plataforma web debe presentar páginas web que se adapten a los diferentes tamaños de pantalla que los usuarios puedan presentar.
RNF-04	Rendimiento	La plataforma web debe presentar mínimos tiempos de carga entre páginas y un acceso a BBDD lo más optimizado posible.
RNF-05	Escalabilidad y desarrollo ágil	El proyecto software debe estar diseñado para poder aumentar exponencialmente el número de usuarios, videojuegos, apartados, etc sin comprometer el sistema.
RNF-06	Pruebas	Deben realizarse pruebas de todo tipo para garantizar la calidad del producto y eliminar bugs y errores del sistema.
RNF-07	Refactorización	Reestructuración del código para una mayor legibilidad de los métodos.

2.2 PLANIFICACIÓN PREVIA

La planificación previa del proyecto se puede definir como la estructuración que el equipo ha hecho por periodos de tiempo de dicho proyecto hasta llegar al producto final.

Dicha planificación ha sido la siguiente:

Semana 1: Elaboración de la estructura del proyecto, software a utilizar y herramientas necesarias.

Semana 2: Análisis y Diseño de la estructura básica del proyecto.

Semana 3: Diseño del modelo de datos del proyecto.

Semana 4: Documentación del trabajo realizado en las semanas anteriores.

Semana 5: Implementación de la estructura básica del proyecto.

Semana 6: Implementación del modelo de datos.

Semana 7: Diseño del login y del menú principal de la plataforma web.

Semana 8: Importación de los videojuegos y diseño de las fichas.

Semana 9: Mejora del GUI de la plataforma web y documentación trabajo realizado en las semanas anteriores.

Semana 10: Optimización de código e implementaciones finales.

Semana 11: Resolución de errores y documentación restante.

Semana 12: Preparación de la presentación y retoques finales del proyecto.

2.3 FASES DE REALIZACIÓN

En este apartado se da una visión general de las fases seguidas a lo largo de la realización del proyecto, que son las correspondientes a la mayoría de un desarrollo software.

- Estudio:** Investigación sobre las posibles tecnologías a utilizar, posible estudio de mercado y de viabilidad. Indagación y pruebas sobre aplicaciones relacionadas, con el objetivo de extraer lo que se suele utilizar en proyectos similares.
- Análisis:** Definición de requisitos con el objetivo de dilucidar exactamente qué es lo que se desea construir.
- Diseño:** Tras finalizar las fases anteriores, se procede a diseñar la arquitectura de la aplicación web. Esto incluye diversos diagramas que definirán el diseño las páginas, la interfaz de usuario, las herramientas a utilizar, etc.
- Implementación:** Programación
- Pruebas:** Una vez finalizadas todas las fases anteriores, se hace necesaria la realización de una fase de pruebas generales para comprobar la calidad y efectividad del software desarrollado.
- Documentación:** La documentación es algo indispensable para el correcto mantenimiento de cualquier producto software. Tanto a nivel de código como de documento, con el fin de dejar constancia por escrito de las fases llevadas a cabo en el ciclo de vida del producto software.
- Mantenimiento:** Una vez finalizadas las fases anteriores, quedaría realizar un mantenimiento periódico para evitar errores o si surgen solventarlos con actualizaciones en la plataforma web.

3 | DISEÑO

3.1 TECNOLOGÍAS UTILIZADAS

En este apartado se enumeran las diferentes herramientas y tecnologías utilizadas durante el desarrollo del proyecto.

3.1.1 HTML

HTML (HyperText Markup Language), hace referencia al lenguaje de marcado para la elaboración de páginas web. Propone una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

HTML se considera el lenguaje web más importante ya que es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.



Logo de HTML5

3.1.2 CSS

CSS (Cascading Style Sheets) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML.

Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C); que proporciona una herramienta de validación de CSS gratuita para los documentos CSS.



Logo de CSS3

3.1.3 PHP

PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web.

Puede ser desplegado en la mayoría de los servidores web y en todos los sistemas operativos y plataformas. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores.

El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un “daemon” o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado (que puede ser cualquier tipo de datos, como HTML generado o datos de imágenes binarias) formaría la totalidad o parte de una respuesta HTTP.



Logo de PHP

3.1.4 JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado que se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. No obstante, el uso de JavaScript en el servidor en aplicaciones externas a la web (documentos PDF, aplicaciones de escritorio, etc.) es también significativo.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Javascript es el único lenguaje de programación que entienden de forma nativa los navegadores.



Logo de JavaScript

3.1.5 PHPStorm

PhpStorm es un IDE (entorno de desarrollo integrado) multiplataforma para PHP, desarrollado por la empresa JetBrains.

PhpStorm proporciona un editor para PHP, HTML y JavaScript con análisis de código en tiempo real, prevención de errores y refactorizaciones automatizadas para código PHP y JavaScript.

El autocompleta de código de PhpStorm es compatible con PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2, 7.3, 7.4, 8.0 y 8.1. Incluye un editor de SQL completo con resultados de consultas editables.

PhpStorm está escrito en Java. Los usuarios pueden ampliar el IDE instalando plugings creados para PhpStorm o escribiendo los suyos propios. El software también se comunica con fuentes externas como XDebug.



Logo de PhpStorm

3.1.6 XAMPP

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), A (Apache), M (MariaDB/MySQL), P (PHP), P (Perl).

A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, una bifurcación del desarrollo de MySQL con licencia GPL.

El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A día de hoy, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.



Logo de XAMPP

3.1.7 GitHub

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos que utiliza el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

El código de los proyectos alojados en GitHub se almacena generalmente de forma pública.

GitHub es famoso por el uso de los repositorios (lugares virtuales alojados en la nube en donde los usuarios pueden almacenar cualquier tipo de archivo). Suelen usarse para guardar archivos de texto que representan código en distintos lenguajes de programación, los cuales a su vez conforman un programa o aplicación. Este es el servicio principal que ofrece GitHub: repositorios en donde los usuarios pueden almacenar el código de sus aplicaciones, ya sea de forma pública, privada o a través de una organización.

Actualmente, GitHub continúa siendo la plataforma más importante de colaboración para proyectos de código abierto.



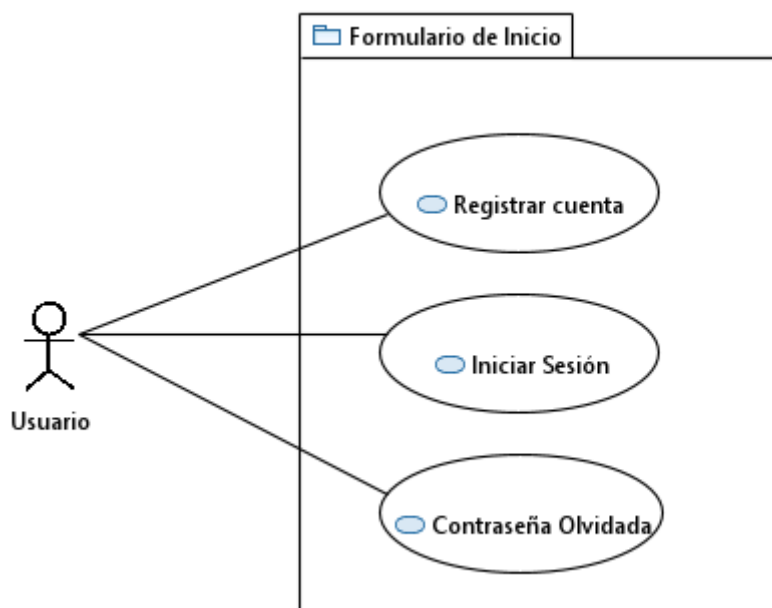
Logo de GitHub

3.2 DISEÑO DEL PROYECTO

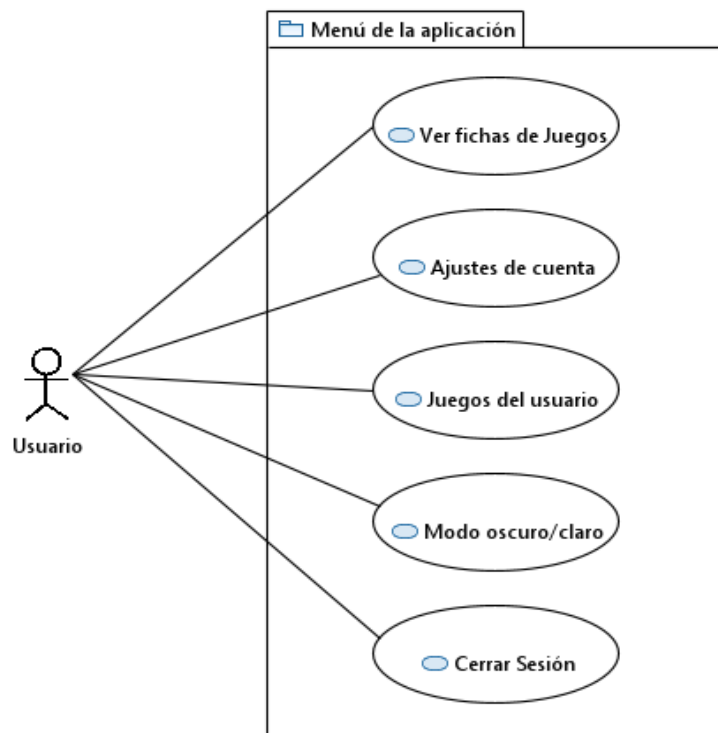
3.2.1 Diagramas de Casos de Uso

Los casos de uso que se presentan en el proyecto son los siguientes:

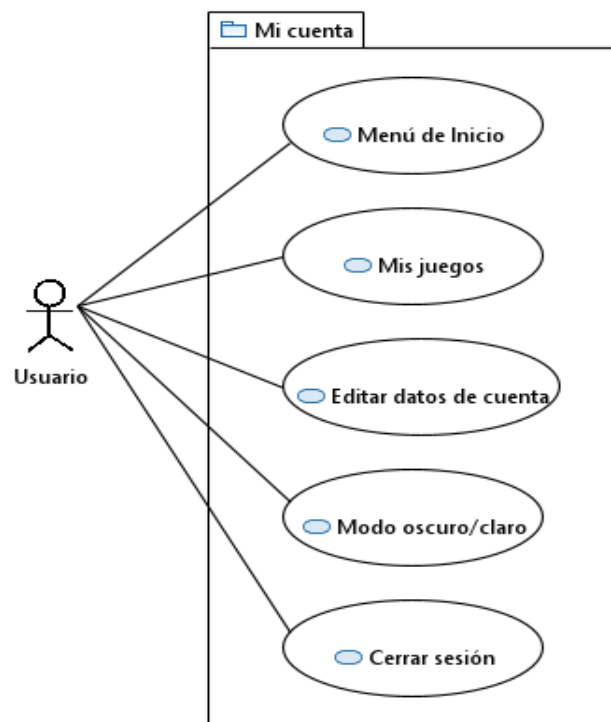
3.2.1.1 Inicio de Sesión



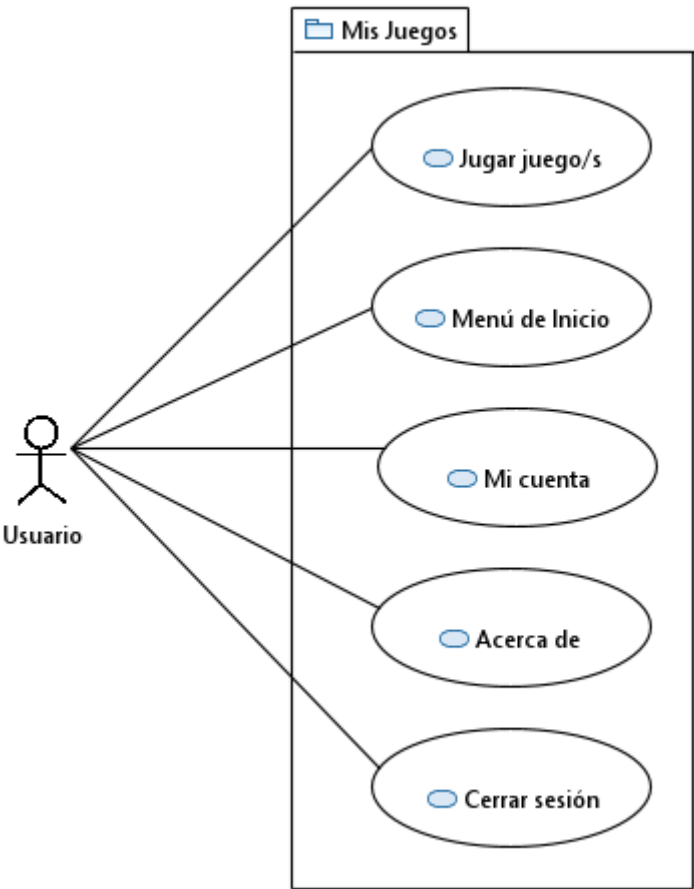
3.2.1.2 Menú de Juegos



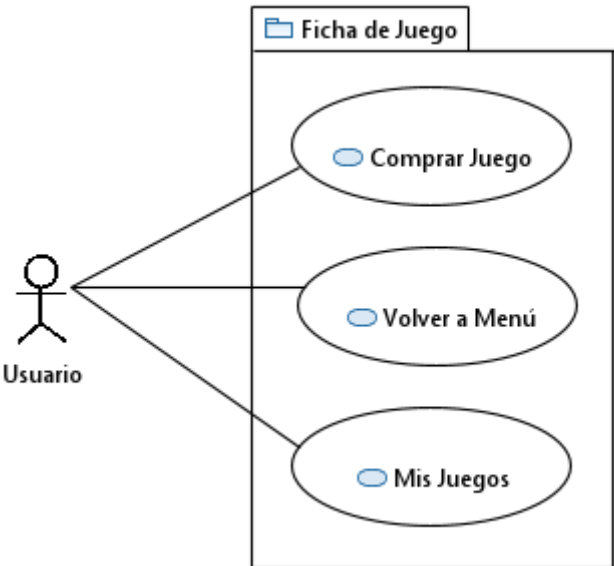
3.2.1.3 Mi Cuenta



3.2.1.4Mis Juegos

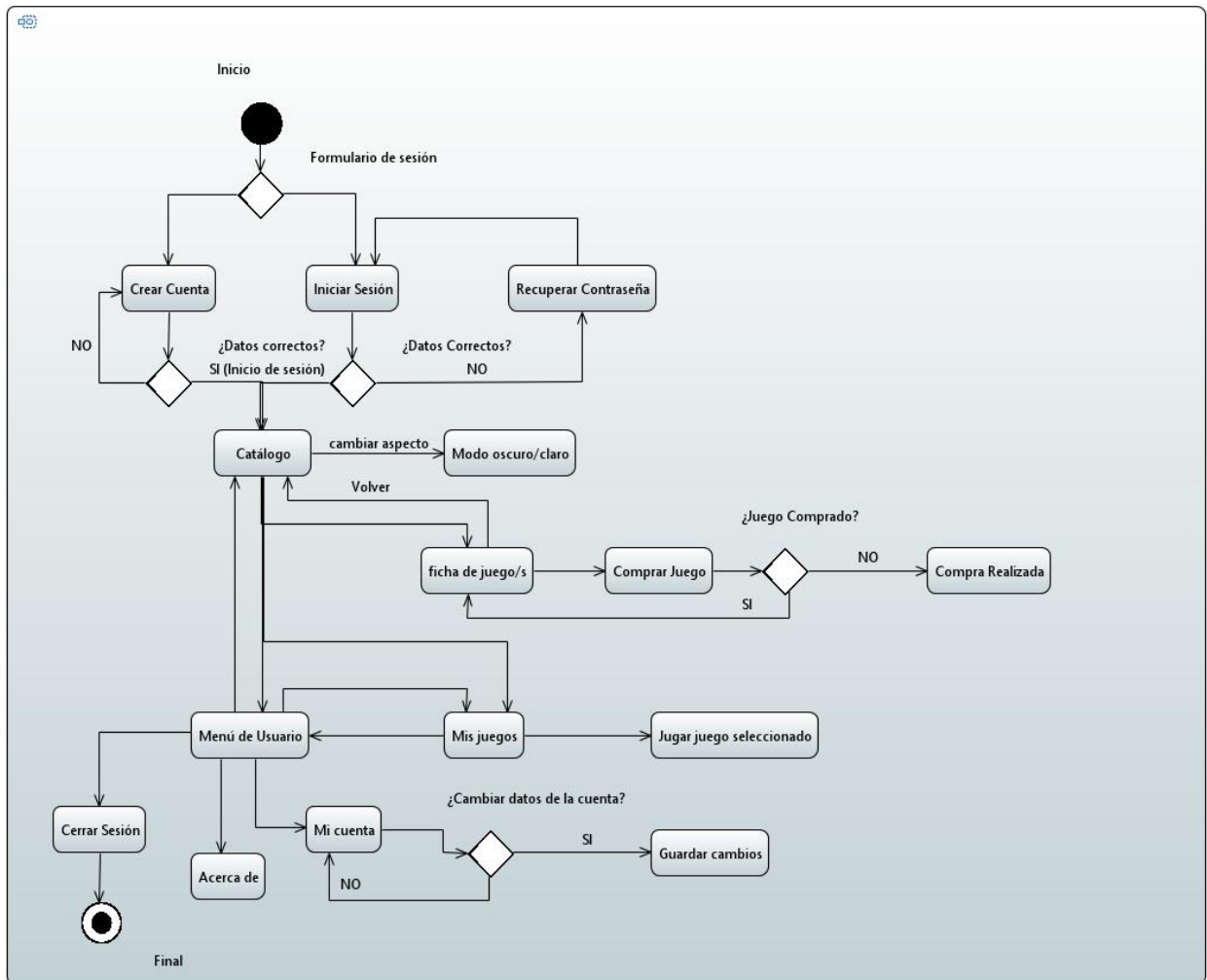


3.2.1.5Ficha de Juego



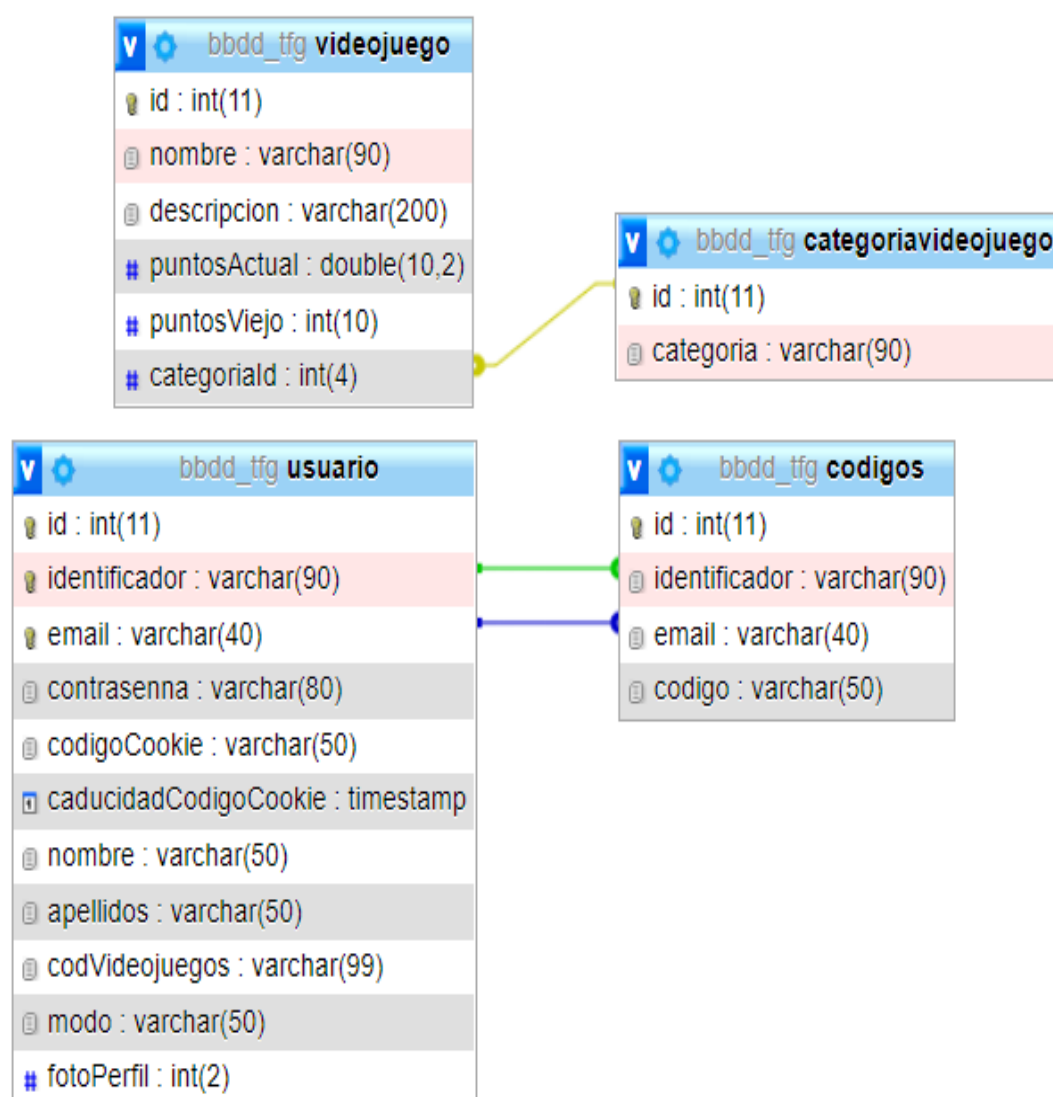
3.2.2 Diagrama de Flujo del proyecto

Imagen 8. Diagrama de Flujo del proyecto



3.2.3 Modelo de datos

Imagen Modelo de datos



3.3 DISEÑO

3.3.1 Menú de Inicio de Sesión

The image displays two side-by-side mockups of the 'Fandom Game Library' login and registration interface. Both mockups feature a dark blue header with the text 'Fandom Game Library' in white. The left mockup shows the registration form, which includes a 'Registrarse' button (highlighted in green) and an 'Iniciar sesión' button (greyed out). Below these buttons is the text 'Regístrate gratis'. The form fields are 'Nombre*', 'Apellidos*', 'Nickname*', 'Email*', and 'Set A Password*'. A large green button labeled 'EMPEZAR' is at the bottom. The right mockup shows the login form, which includes a 'Registrarse' button (greyed out) and an 'Iniciar sesión' button (highlighted in green). Below these buttons is the text '¡Bienvenido de nuevo!'. The form fields are 'Nickname*' and 'Password*'. A link '¿Contraseña Olvidada?' is located below the password field. A large green button labeled 'ACCEDER' is at the bottom.

El menú de inicio de sesión está estructurado en un contenedor principal el cual contiene dos contenedores, uno con el formulario de inicio de sesión y otro con el de creación de una cuenta y dos botones los cuales permiten cambiar entre formularios.

Cada formulario presenta un botón distinto de confirmación de los datos, además en el formulario de sesión presenta un enlace hacia la página de contraseña olvidada.

Encima del contenedor principal está presente la cabecera con el nombre de la aplicación.

Se decidió para el inicio de sesión, la página de contraseña olvidada y el modo oscuro de la aplicación el color #13232F el cual es un azul-cian oscuro para añadirle un aspecto profesional y personalizado, siendo una mejor opción que un color más oscuro ya que dejaba un contraste muy alto entre el fondo y los elementos de la página, además es un color oscuro pero suave el cual es agradable para la lectura.

3.3.1 Contraseña Olvidada

The image displays three sequential screenshots of a web application's password reset process. The first screenshot shows a form titled 'Fandom Game Library' with fields for 'Nickname' and 'Email', and a prominent green 'REESTABLECER' button. The second screenshot shows a confirmation screen with a green message 'Un paso más cerca.', a code '6275452408f45', and an 'Enviar' button. The third screenshot shows the final form with fields for 'Nickname', 'Email', 'Contraseña Nueva', and 'Confirmar Contraseña', and a 'REESTABLECER' button.

El diseño de restablecer la contraseña presenta una estructura muy simple, al tratarse de formularios, presentan los mismos colores y diseños de entradas de texto que el menú de inicio de sesión para una homogeneidad de colores y estructura.

3.3.2 Catálogo de Videojuegos

El catálogo de videojuegos presenta un diseño en el cual contiene una cabecera y un pie de página los cuales son fijos, la cabecera a su vez posee dos botones, uno el cual despliega un menú de usuario oculto con accesos para las páginas de la aplicación y otro el cual permite al usuario cambiar el modo de la aplicación (claro u oscuro).

En la parte superior, debajo de la cabecera se encuentra un seleccionador el cual filtra los videojuegos y debajo de este elemento se encuentra un contenedor el cual contiene todas las tarjetas creadas dinámicamente de los videojuegos.

A su vez, todas las tarjetas poseen el mismo estilo y dimensiones para un mayor aspecto de profesionalidad y homogeneidad en la página.

Catálogo

Filtrar por Tipo

Todos



COLORON
Cambia de color todas las plataformas para que la pelota logre avanzar.
GRATIS
100 Puntos
[VER FICHA](#)



COPYCAT
Consigue pasar los niveles con dos personajes.
GRATIS
200 Puntos
[VER FICHA](#)



EL AHORCADO
Adivina la palabra antes de que sea demasiado tarde.
GRATIS
200 Puntos
[VER FICHA](#)

© 2022 The FGL Association
Todos los derechos reservados a sus respectivos dueños.

Catálogo

Filtrar por Tipo

Todos



COLORON
Cambia de color todas las plataformas para que la pelota logre avanzar.
GRATIS
100 Puntos
[VER FICHA](#)









COPYCAT
Consigue pasar los niveles con dos personajes.
GRATIS
200 Puntos
[VER FICHA](#)



EL AHORCADO
Adivina la palabra antes de que sea demasiado tarde.
GRATIS
200 Puntos
[VER FICHA](#)

© 2022 The FGL Association
Todos los derechos reservados a sus respectivos dueños.

-  Inicio
-  Mi Cuenta
-  Mis Juegos
-  Acerca de
-  Comprar Puntos
-  Cerrar Sesión

Todos



COLORON
Cambia de color todas las plataformas para que la pelota logre avanzar.
GRATIS
100 Puntos
[VER FICHA](#)



COPYCAT
Consigue pasar los niveles con dos personajes.
GRATIS
200 Puntos
[VER FICHA](#)



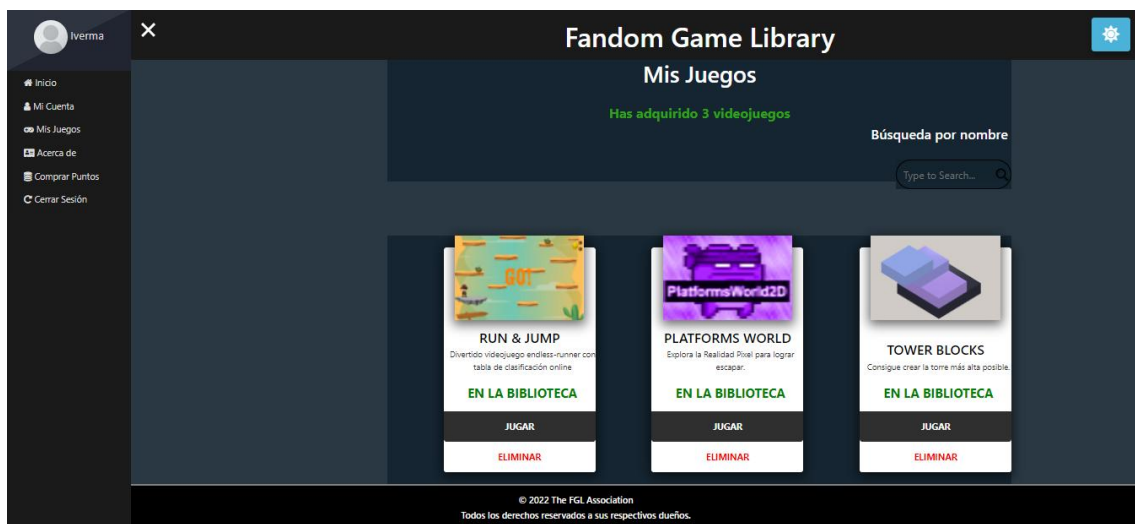
EL AHORCADO
Adivina la palabra antes de que sea demasiado tarde.
GRATIS
200 Puntos
[VER FICHA](#)

© 2022 The FGL Association
Todos los derechos reservados a sus respectivos dueños.

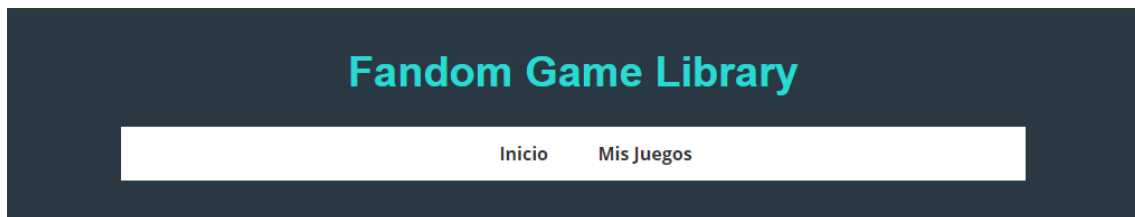
3.3.3 Biblioteca de Mis Juegos

La biblioteca de juegos del usuario presenta un diseño similar al catálogo de videojuegos, en la parte superior una cabecera fija con los mismos elementos (título, botón de menú desplegable y botón de modo claro/oscuro) y un buscador de juegos en vez de un seleccionador.

En la parte inferior se encuentra un contenedor principal donde dinámicamente aparecen todos los videojuegos obtenidos por el usuario y el pie de página fijo.



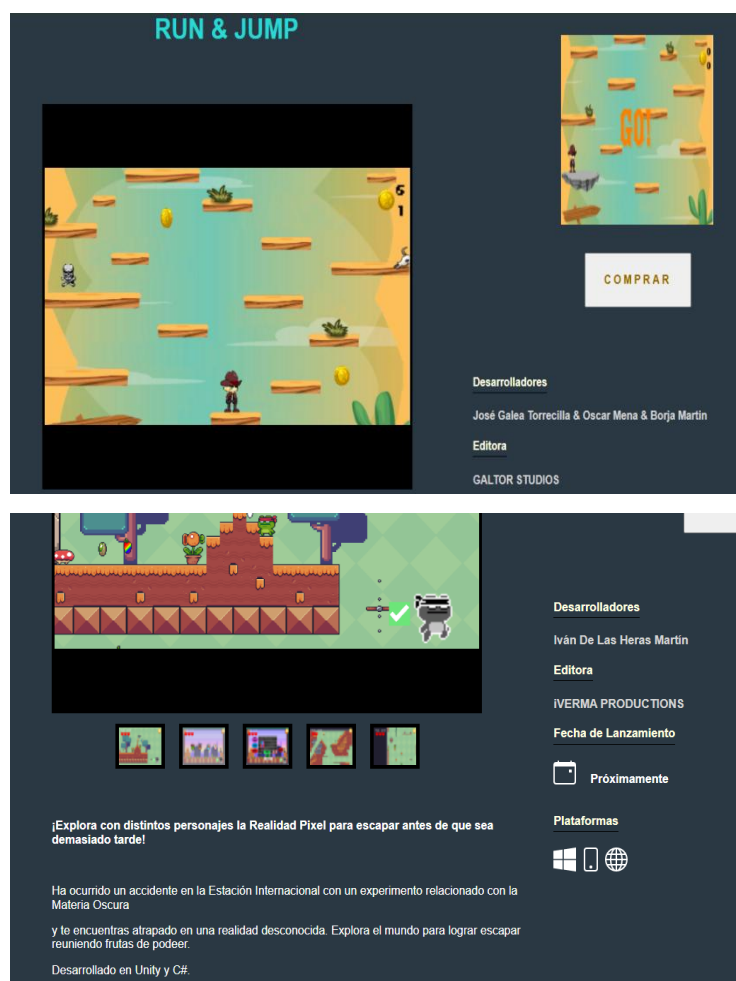
3.3.4 Ficha de Videojuegos



Las fichas de cada videojuego presentan una estructura similar, todas tienen en la parte superior una cabecera con un menú estático, logrando así una mayor accesibilidad, a partir de dicho menú se puede acceder a las páginas de “Inicio” (Catálogo) y “Mis Juegos”.

Después, está situado en la parte izquierda un contenedor con el título del videojuego, un carrusel de imágenes con botones de cambio de imagen, por último, estará situada una breve descripción con información relevante del videojuego debajo de dicho carrusel.

En la parte derecha se sitúa otro contenedor el cual contiene el logo del videojuego, un botón personalizado el cual permite adquirir el videojuego, información sobre las empresa y desarrolladores dedicados a la creación de dicho videojuego, su fecha de lanzamiento y las plataformas en las que está disponible.



3.3.5 Mi Cuenta

La página de “Mi cuenta” posee un formulario con todos los datos del usuario donde puede cambiar sus datos si desea. En la parte superior está situada la imagen de perfil del usuario la cual contiene una entrada para subir/cambiar la imagen y un botón de confirmación.

En la parte inferior presenta entradas de texto con el nickname, nombre, apellidos, correo electrónico del usuario y un botón de confirmación.

La cabecera presenta los mismos elementos que el catálogo para un fácil acceso del usuario al menú desplegable y todas las páginas de la aplicación o al cambio de modo claro/oscuro.

Fandom Game Library

Mi Cuenta

Seleccionar archivo

Subir imagen

Nickname
Iverma

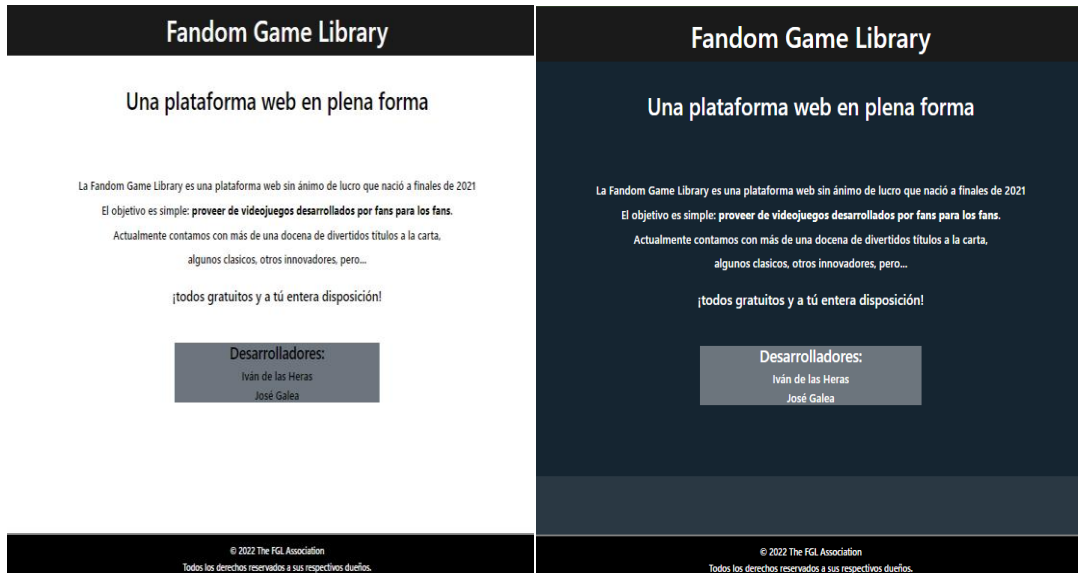
Nombre
Ivan

Apellidos
Delas

Email
dbdtodoeldia@gmail.com

3.3.6 Acerca De

En la página de “Acerca de” se muestra el objetivo de nuestra aplicación y quienes somos los desarrolladores.



3.3.7 Comprar Puntos

El diseño de la página de compra de puntos, al estar en desarrollo para futuras implementaciones, presenta la cabecera y el pie de página predeterminados de todas las páginas y tres botones con los cuales se podrán comprar distintas cantidades de puntos (al entrar, se le advierte al usuario de que esta función está en desarrollo).

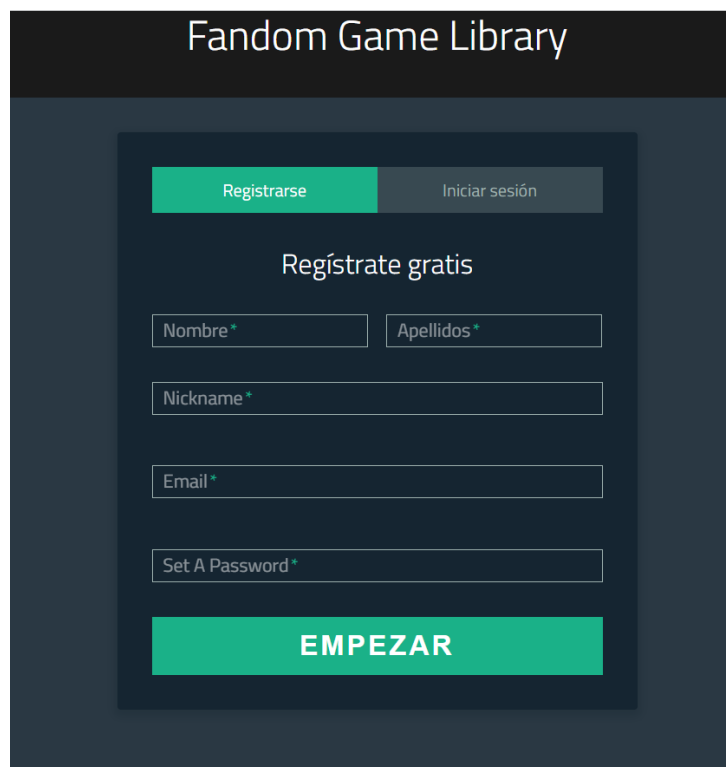


4 | IMPLEMENTACIÓN

En el siguiente punto se detallará la parte de programación e implementación de dicho código en el proyecto.

- Inicio de sesión y creación de nuevas cuentas

A continuación se muestra una captura de pantalla de la página que se encarga del logging por parte del usuario.

The image shows a web interface for 'Fandom Game Library'. At the top, there's a dark header with the title 'Fandom Game Library' in white. Below the header, there's a registration form. The form has two buttons at the top: 'Registrarse' (highlighted in green) and 'Iniciar sesión' (grey). The main heading of the form is 'Regístrate gratis'. Below this, there are five input fields: 'Nombre*' and 'Apellidos*' (side-by-side), 'Nickname*', 'Email*', and 'Set A Password*'. At the bottom of the form is a large green button labeled 'EMPEZAR'.

Registro de nuevos usuarios

Como se puede ver en el apartado “Registrarse” se solicitan una serie de datos al usuario para generar una nueva cuenta en el sistema. Una vez enviado, los datos del formulario son debidamente tratados y si cumplen los valores esperados se procederá a crear la nueva cuenta al usuario.

```

function crearNuevoUsuario($identificador, $email, $contrasenna, $nombre, $apellidos): int
{
    /*SIGNIFICADOS RETURNS:
    -3 -> Error al crear la cuenta (se ha realizado el "INSERT" pero algo ha salido mal) ERROR
    1 -> Operación realizada con ÉXITO
    */

    $conexion = obtenerPdoConexionBD();

    //En caso de que no esten repetidos. Creo el nuevo usuario en la BBDD
    $sql = "INSERT INTO usuario (identificador, email, contrasenna, codigoCookie, caducidadCodigoCookie, nombre, apellidos) VALUES (?, ?, ?, ?, ?, ?, ?)";
    $select = $conexion->prepare($sql);
    $select->execute([$identificador, $email, $contrasenna, null, null, $nombre, $apellidos]);
    $numRegistros = $select->rowCount();
    if ($numRegistros == 1) {
        //Devuelvo 1 que quiere decir que la operación ha salido bien
        return 1;
    } else {
        //Devuelvo -3 que quiere decir que algo ha salido mal en el insert de la BBDD
        return -3;
    }
}

```

Método utilizado para la creación de la nueva cuenta y su inserción en el sistema

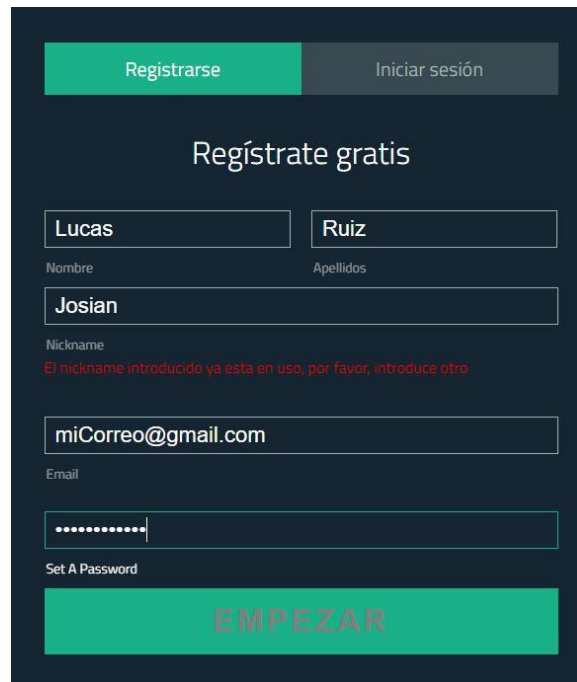
Si el proceso se ha realizado correctamente, el usuario podrá acceder a la plataforma, en caso contrario se le indicará el error en la misma página de registro ya que esta no necesita ser refrescada en ningún caso (pues el sistema actualiza el estado de la página mediante interacciones AJAX).

```

function validarCampos() {
    document.getElementById("pErrorNickName").innerHTML = "";
    document.getElementById("pErrorEmail").innerHTML = "";
    document.getElementById("btnCrearCuenta").removeAttribute("disabled");
    document.getElementById("btnCrearCuenta").style.color = "white";
    //console.log("Se salió del nickname o del email");
    nickname = document.getElementById("identificador").value;
    email = document.getElementById("email").value;
    //console.log("Los parametros dados son --> nickname: " + nickname + " y email: " + email);
    llamadaAjax("ValidacionFormulario.php", nickname, email, "",
        function(texto) {
            // debugger
            var numOperacion = JSON.parse(texto);
            // console.log("numOperacion es: " + numOperacion);
            if (numOperacion == -1) {
                document.getElementById("pErrorNickName").innerHTML = "El nickname introducido ya esta en uso, por favor, introduce otro";
                document.getElementById("btnCrearCuenta").setAttribute("disabled", "true");
                document.getElementById("btnCrearCuenta").style.color = "grey";
            } else if (numOperacion == -2) {
                document.getElementById("pErrorEmail").innerHTML = "El email introducido ya esta en uso, por favor, introduce otro";
                document.getElementById("btnCrearCuenta").setAttribute("disabled", "true");
                document.getElementById("btnCrearCuenta").style.color = "grey";
            }
        },
        function(texto) {
            notificarUsuario("Error Ajax al cargar personas al inicializar: " + texto);
        }
    );
}

```

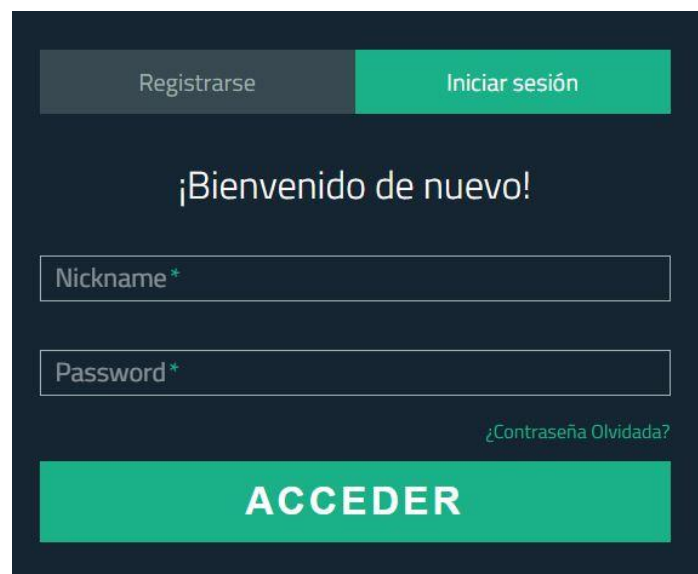
Método utilizado para la validación de los datos de registro



Registration form interface with a dark blue background. At the top, there are two buttons: "Registrarse" (highlighted in green) and "Iniciar sesión". Below these is the heading "Regístrate gratis". The form contains several input fields: "Nombre" with the value "Lucas", "Apellidos" with the value "Ruiz", "Nickname" with the value "Josian", and "Email" with the value "miCorreo@gmail.com". The password field is masked with dots. A red error message below the nickname field reads: "El nickname introducido ya está en uso, por favor, introduce otro". At the bottom is a large green button labeled "EMPEZAR".

El sistema notifica al usuario que el nickname que ha introducido ya está en uso mediante llamadas AJAX sin necesidad de actualizar la página.

En caso de que el usuario ya esté registrado en el sistema procederá a iniciar sesión, para ello entrará en el apartado de "Iniciar sesión" que se encuentra en la misma página de inicio.



Login form interface with a dark blue background. At the top, there are two buttons: "Registrarse" and "Iniciar sesión" (highlighted in green). Below these is the heading "¡Bienvenido de nuevo!". The form contains two input fields: "Nickname*" and "Password*". A link "¿Contraseña Olvidada?" is located below the password field. At the bottom is a large green button labeled "ACCEDER".

Inicio de sesión

Si el proceso se ha realizado correctamente, el usuario podrá acceder a la plataforma con su usuario. En caso de que los datos del logging (nickname y password) introducidos por el usuario sean incorrectos, el sistema le informará de ello.

The screenshot shows the login interface for 'Fandom Game Library'. At the top, a red banner displays the error message 'Error al introducir los datos'. Below this, there are two buttons: 'Registrarse' (grey) and 'Iniciar sesión' (green). The main heading is '¡Bienvenido de nuevo!'. The login form contains a 'Nickname' field with the text 'asdsd' and a 'Password' field with masked characters. A link for '¿Contraseña Olvidada?' is visible next to the password field. A large green 'ACCEDER' button is at the bottom of the form. Below the form, the text 'Datos incorrectos' is displayed.

Datos incorrectos

Para mayor seguridad, la plataforma almacena cifrados los datos sensibles de los usuarios (en este caso, las contraseñas) en la BBDD del sistema. Cuando un usuario se crea una cuenta, la contraseña que establece se almacena cifrada.

De esta forma, cuando el usuario inicia sesión el sistema cifra la contraseña introducida y la compara con el texto cifrado que tiene almacenado en la BBDD y, en caso de que coincida, se procede al inicio de sesión.

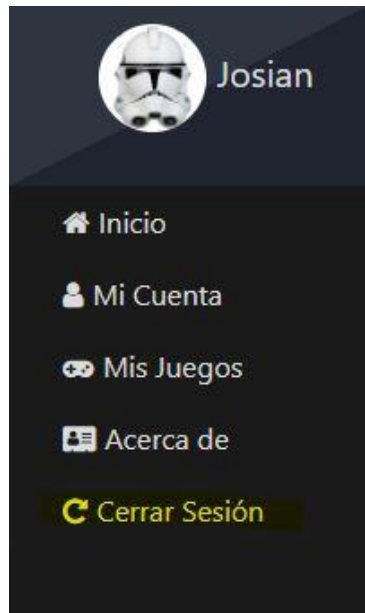
id	identificador	email	contrasenna	nombre	apellidos	codVideojuegos	modo	fotoPerfil
2	Josian	████████@gmail.com	8a89e5e5072e7ceb40c02b3c3d0fd977	Josian	Galtor	2,5	oscuro	0
3	ManCa	████████@gmail.com	d6fd0924e324f50669ae0295adf59567	Manuela	Carrasco	10,11	claro	0
4	Iverma	████████@gmail.com	f0b53b2da041fca49ef0b9839060b345	Ivan	Delas	1,2,13	claro	0
5	Novo	██████@gmail.com	3e6c7d141e32189c917761138b026b74	Nuevo	Nuevisimo	NULL	oscuro	0

Ejemplo de usuarios registrados en la tabla de "Usuarios"

Tanto para el registro de nuevos usuarios como para el inicio de sesión de los que están ya registrados el sistema consulta la tabla "Usuarios" de la BBDD del proyecto.

- Cerrar sesión

Una vez que el usuario se encuentra en el sistema podrá salir del mismo sin riesgo de perder datos siempre que cierre sesión. Para ello, se le facilita en un menú desplegable una serie de opciones entre las que se encuentra el cierre de sesión.



Menú desplegable del usuario "Josian"

Cuando el usuario hace clic en esta opción se llama a la misma página que muestra las opciones para el logging del usuario sólo que esta vez el sistema le envía un parámetro indicando que se trata de un cierre de sesión por parte del usuario.

Una captura de pantalla de la página 'Fandom Game Library'. En la parte superior, hay un mensaje de estado azul que dice 'Se ha cerrado la sesión.'. Debajo, hay dos botones: 'Registrarse' (verde) y 'Iniciar sesión' (gris). El formulario principal se titula 'Regístrate gratis' y contiene campos de entrada para 'Nombre *', 'Apellidos *', 'Nickname *', 'Email *' y 'Set A Password *'. Al final del formulario, hay un botón verde grande con el texto 'EMPEZAR'.

Sesión cerrada con éxito

```

<section class="bloqueInfoSesion">
  <?php if (isset($_REQUEST["sesionCerrada"])) {?>
    <p style="...">Se ha cerrado la sesión.</p>
  <?php } if (isset($_REQUEST["error"])) {?>
    <p style='...'>Error al introducir los datos<p>
  <?php } if (isset($_REQUEST["errorCreacionCuenta"])) {?>
    <p style='...'>Se ha producido un error inesperado. Por favor, vuelve a introducir los datos<p>
  <?php }?>
</section>

```

Código necesario para mostrar el mensaje correspondiente al usuario

- Nueva Contraseña (PHP)

- Generar Nueva Contraseña (PHP)

```

<?php
require_once '../_Sesion.php';
require_once '../_Varios.php';

$filasNuevas = crearNuevaContrasenna($_REQUEST["identificador"],
$_REQUEST["email"], $_REQUEST["contrasenna"]);

?>

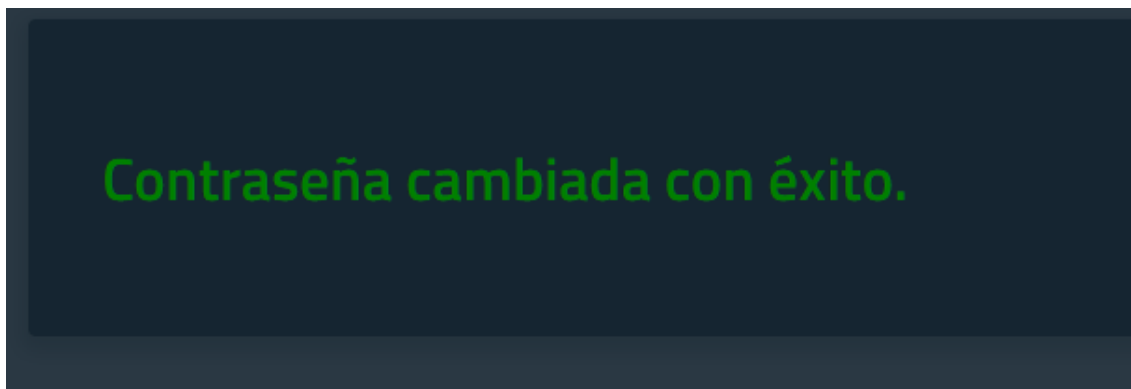
```

Este script notifica al Usuario si el proceso de actualización de la contraseña ha sido realizado de manera exitosa o no.

Primero, recibe como parámetros el nickname, correo electrónico y la contraseña nueva, llamando a un método externo para actualizar en la base de datos la

contraseña nueva al usuario con el nickname y el correo electrónico enviado, el cual, si uno de los dos datos no es válido, se cancelará la actualización.

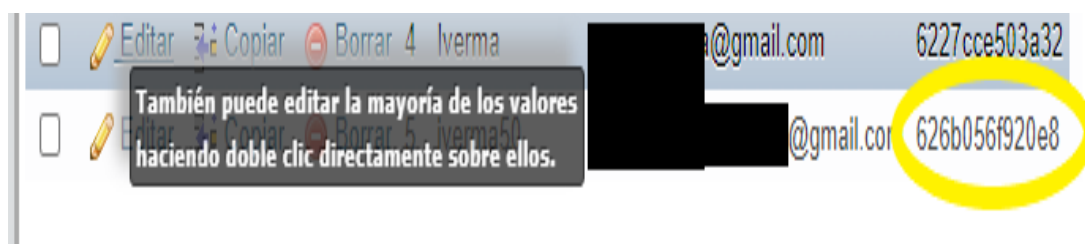
Si el método devuelve el número uno, la operación ha resultado exitosa, notificando con un mensaje al usuario, en caso contrario, se le notificará un error al usuario y se le dirigirá al formulario de inicio de sesión.



- Enviar Correo Contraseña (PHP)

Este script recibe el nickname y correo electrónico del usuario para después enviar un código de verificación a dicho correo, este código será utilizado para validar que el usuario que ha solicitado el cambio de contraseña es el propietario de dicha cuenta.

Todos los usuarios, al crear una cuenta se les asignan un código de verificación aleatorio que se guarda en una tabla diferente en la base de datos.

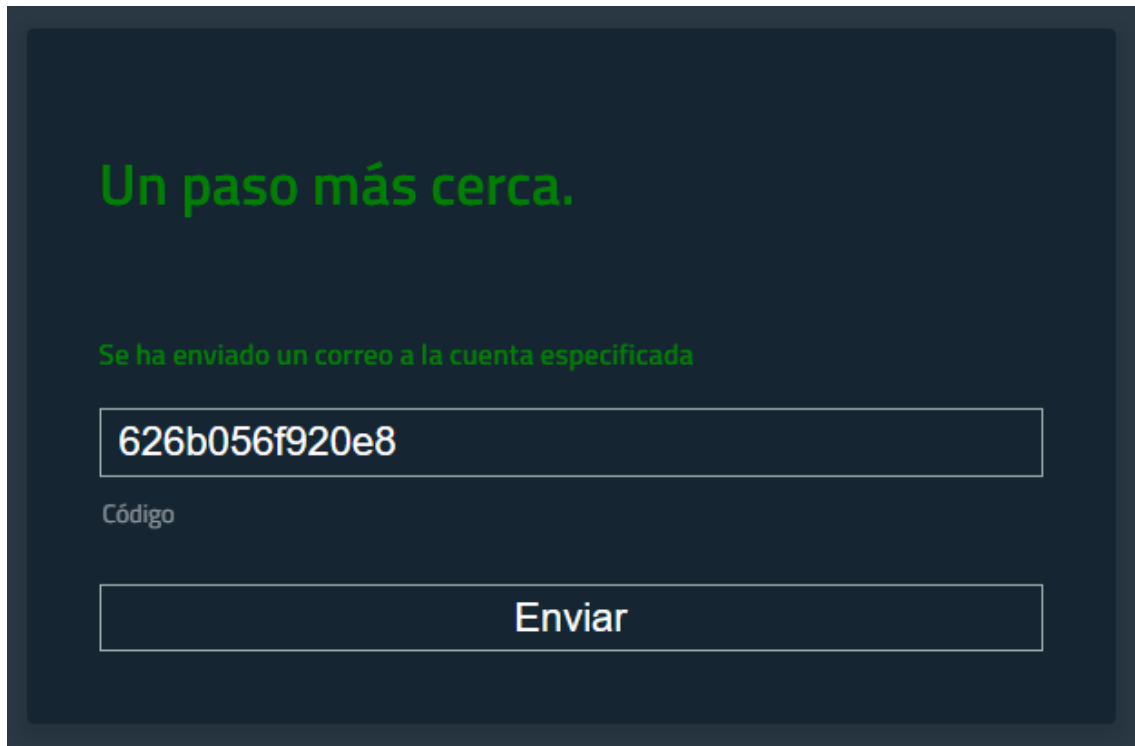


Al recibir el correo de solicitud de cambio de contraseña, el nuevo código generado renueva al código antiguo de la base de datos, impidiendo así que el código de recuperación sea el mismo una vez cambiada la contraseña.

En todo caso, para que el correo sea enviado se comprobará que dicho correo existe, en el caso de que no exista, se le notificará al usuario de un error para el envío del correo.

Si dicho correo es enviado correctamente, se le notificará al usuario de que el correo se envió sin problemas y aparecerá automáticamente un formulario para la validación del código de verificación.

Mediante una llamada a la base de datos, se verificará que el código introducido en el formulario es correcto, en dicho caso, el usuario será dirigido a la página de actualización de la contraseña, en cambio, si el código es incorrecto, aparecerá un mensaje de error.



Un paso más cerca.

Se ha enviado un correo a la cuenta especificada

626b056f920e8

Código

Enviar

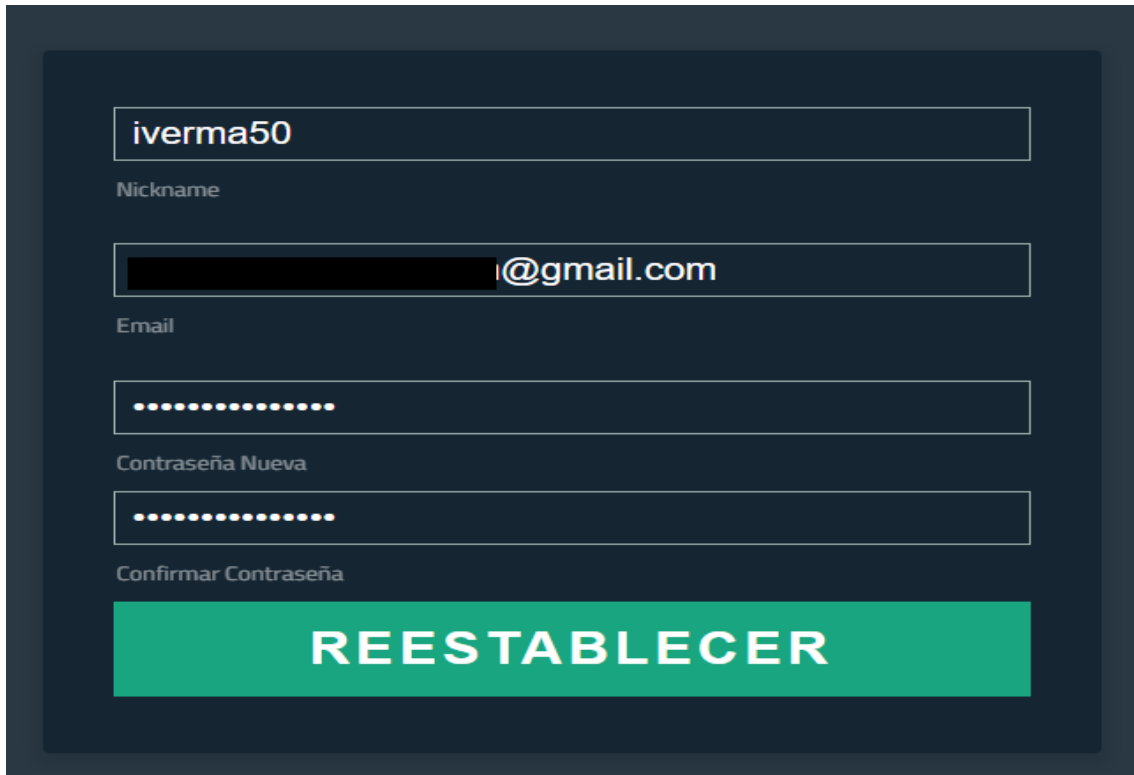
```
<?php
require_once '../../_Sesion.php';
require_once '../../_Varios.php';

if (isset($_REQUEST["email"]) && isset($_REQUEST["identificador"])){
    $email = $_REQUEST["email"];
    $identificador = $_REQUEST["identificador"];
} else {
    $email = "";
    $identificador = "";
}

if (isset($_REQUEST['cod'])){
    redireccionar("CorreoContrasenna.php");
} else {
    $correoEnviado =
    enviarMensajeCorreoPassword($email,$identificador);
}
?>
```

Comprobación para enviar el correo al usuario con el código

- Correo Contraseña (PHP)



The image shows a dark-themed web form for password reset. It contains four input fields: 'Nickname' with the text 'iverma50', 'Email' with a partially redacted address ending in '@gmail.com', 'Contraseña Nueva' (New Password) with masked dots, and 'Confirmar Contraseña' (Confirm Password) also with masked dots. Below the fields is a large green button with the text 'REESTABLECER' in white capital letters.

Este script sirve para la actualización de la contraseña del usuario, una vez introducido el código recibido correctamente, el usuario será enviado a esta página donde deberá introducir como doble verificación su nickname y su correo electrónico, a continuación, el usuario deberá introducir su contraseña nueva la cual deberá contener una letra mayúscula, una letra minúscula, un número y con una longitud mínima de ocho caracteres.

La contraseña, al ser cambiada es cifrada y después insertada en la base de datos, en ningún momento la clave de un usuario es descifrada para una mayor seguridad.







Para comprobar la existencia del nickname y correo electrónico del usuario se realiza una llamada AJAX la cual llama a un script PHP el cual obtiene de la base de datos el nickname y correo electrónico del usuario, si no existen, los campos se mostrarán en rojo con un texto para el usuario el cual contiene: "El nickname/correo electrónico no existe".







En cambio, si los datos introducidos existen, los campos se mostrarán en verde con un texto el cual contiene: "Datos correctos".




Para dicha comprobación de los datos, también se necesita otro script JS el cual muestra los mensajes de error.

También, se realiza una llamada independiente a otro script JS para la confirmación de la contraseña en los dos campos (contraseña y confirmar contraseña).

Una vez todos los cambios sean validados, la contraseña se cambiará en la base de datos:

<input type="checkbox"/>				4	Iverma	"[redacted]"@gmail.com	iv	NULL
<input type="checkbox"/>				5	iverma50	"[redacted]"@gmail.com	Contra1234	NULL

<input type="checkbox"/>				4	Iverma	"[redacted]"@gmail.com	iv	NULL
<input type="checkbox"/>				5	iverma50	"[redacted]"@gmail.com	ContraNueva1234	NULL

↑ ☐ Seleccionar todo Para los elementos que están marcados:  Editar  Copiar  Borrar

- Validación Formulario Contraseña (JS)

Nickname

El nickname introducido NO EXISTE, por favor, introduce otro

Email

REESTABLECER



Utiliza un formato que coincida con el solicitado

Introduce una contraseña con al menos un numero, una letra minúscula, otra mayúscula y con una longitud mínima de 8 caracteres

The image shows a dark-themed web form for password recovery. It contains two input fields: the first is labeled 'Nickname' and contains the text 'Iverma'; the second is labeled 'Email' and contains the text 'a@gmail.com'. Below the email field, there is a red error message that reads 'El email introducido NO EXISTE, por favor, introduce otro'. At the bottom of the form is a large green button with the text 'REESTABLECER' in white capital letters.

La funcionalidad de este script permite la validación del nickname y correo electrónico introducido por el usuario para su recuperación de la contraseña.

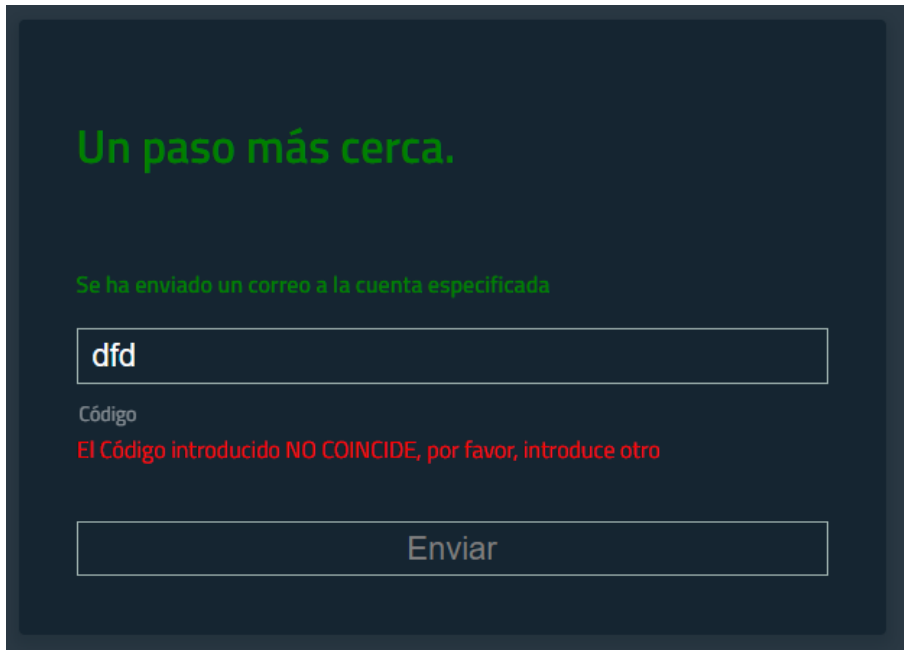
Para ello, al cargar todos los elementos de la página, se le añade a los campos un evento focus (permite realizar una opción cuando el usuario tiene el foco en dicho campo) el cual llamará a una función la cual añadirá un evento focusout (permite realizar una opción cuando el foco del usuario cambia de campo) el cual validará si los datos introducidos en ese campo son correctos o no.

La función validarCampos desactivará (`disabled = true`) el botón para continuar si uno o todos los campos son incorrectos, en caso de que todos los datos sean correctos se le notificará mediante un mensaje al usuario de que puede continuar.

Para validar los datos, esta función realiza una llamada AJAX a un script PHP el cual recibe el nickname y correo electrónico y a sobre la base de datos ejecuta una consulta, comprobando si el nickname y correo electrónico existen.

La llamada AJAX devolverá un número según el resultado de la consulta, si el número devuelto es -1 quiere decir que el nickname no existe, si es un -2, el correo electrónico no existe y en caso de cualquier otro número, la consulta ha encontrado el nickname y correo electrónico de dicho usuario.

- Validación del Código de usuario (JS)

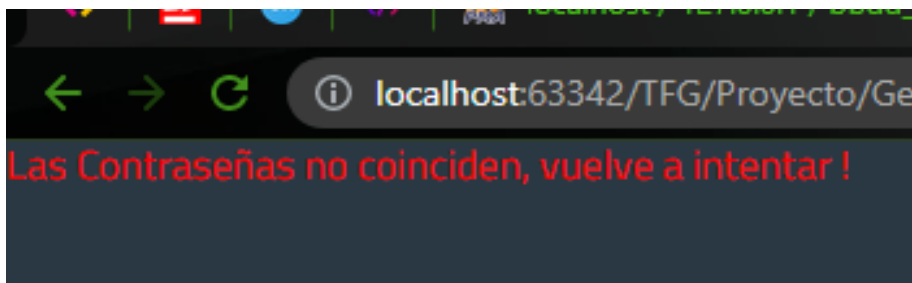


The screenshot shows a dark-themed web interface. At the top, the text "Un paso más cerca." is displayed in green. Below it, a message states "Se ha enviado un correo a la cuenta especificada" in green. A text input field contains the code "dfd". Below the input field, the label "Código" is shown in a small, light gray font. A red error message reads "El Código introducido NO COINCIDE, por favor, introduce otro". At the bottom, there is a button labeled "Enviar".

Este script verifica que el código introducido por el usuario sea correcto. Al haberse creado la cuenta cada usuario recibe un código de recuperación (el usuario no le conoce a no ser que tengo errores con el acceso a su cuenta).

Mediante una llamada AJAX se recoge de la base de datos el código del usuario que desea cambiar su contraseña, al ser obtenido, se comprueba si el código introducido en el formulario es el mismo o no, en caso de que no sea, se le notificará al usuario de que el código no es correcto y deberá introducir el código correctamente. Si el código es el correcto, el usuario accederá a un formulario de cambio de contraseña.

- Confirmación Contraseña (JS)



La funcionalidad de este script comprueba que la nueva contraseña sea verificada e insertada correctamente.

Para ello se recogen los valores de los campos contraseña y contraseña Confirmar, si el valor es distinto entre los dos campos, aparecerá un mensaje al usuario de que

las contraseñas no coinciden y deshabilitando el botón de cambio de contraseña, en caso de que coincidan, aparecerá un mensaje al usuario y el botón de cambio de contraseña se activará.

- Gestión Menú Principal (JS)

```
function inicializar() {  
  
    //Obtengo todos los videojuegos de la base de datos  
    llamadaAjax("ObtenerJuegosBBDD.php", "",  
        function(texto) {  
            // debugger  
            var videojuegos = JSON.parse(texto);  
            for (let i = 0; i < videojuegos.length; i++) {  
                // console.log(videojuegos[i]);  
                insertarVideojuego(videojuegos[i]);  
                addVideojuegoSelectFiltro("selectTipos",  
videojuegos[i]);  
                todosLosDatosCargados = true;  
  
document.getElementById("selectTipos").addEventListener("click",  
realizarFiltro, false);  
            }  
        },  
        function(texto) {  
            notificarUsuario("Error Ajax al cargar las tarjetas de los  
juegos: " + texto);  
        });  
}
```

PARTE 1: Este script permite crear las tarjetas personalizadas de los videojuegos, asignando automáticamente un estilo CSS de un script CSS ya creado.

Primero, al cargar todos los elementos de la página, se llamará a la función inicializar la cual realiza una llamada AJAX a un script PHP (ObtenerJuegosBBDD), la cual obtiene todos los datos sobre cada videojuego, cada vez que los datos de un videojuego es recogido, mediante un bucle se envían los datos de cada videojuego a una función llamada insertarVideojuego donde se crearán mediante JavaScript todos los elementos de la tarjeta y a la función addVideojuegoSelectFiltro la cual llenará un select con las categorías de los videojuegos para obtener un buscador de videojuegos por categorías.

Cuando todos los datos estén cargados, el select creado tendrá un evento click el cual permitirá mostrar los videojuegos de cada categoría al ser pulsada una categoría específica.

```

function obtenerTodasLasCategoriasAJAX(select, videojuegoActual) {
    //obtengo todas las categorias de la base de datos
    //y creo los options del select (uno por cada tipo DISTINTO de
    categoria que aparezca)
    llamadaAjax("ObtenerCategoriasBBDD.php", "",
        function(texto) {
            // debugger
            categorias = JSON.parse(texto);
            // console.log("categorias al inicializar tiene: " +
            categorias.length);
            var optionsExistentes = select.options;
            var existe = false;
            for (let i = 0; i < optionsExistentes.length; i++) {
                if (optionsExistentes[i].value ==

                    videojuegoActual.categoriaId) {
                        existe = true;
                    }
            }
            if (!existe) {
                for (let i = 0; i < categorias.length; i++) {
                    if (categorias[i].id ==

                        videojuegoActual.categoriaId) {
                            var opcion = new

                                Option(categorias[i].categoria, 

                                    videojuegoActual.categoriaId);
                                    select.appendChild(opcion);
                                }
                    }
                }
            },
            function(texto) {
                notificarUsuario("Error Ajax al cargar las tarjetas de los

                    juegos: " + texto);
            });
        }
    }
}

```

PARTE 2: La función obtenerTodasLasCategoriasAJAX realiza una llamada AJAX al script PHP ObtenerCategoriasBBDD, el cual realiza una consulta en la base de datos y devuelve todas las categorías existentes.

Al recoger todos los datos se añade cada categoría en cada option del select creado mediante un bucle, creando un option por cada categoría y añadiéndolo al select con la función appendChild().

```

function realizarFiltro(e) {
    eliminarTodosLosHijosDivDatos();
    //Aquí obtengo el valor que elije el usuario para hacer el
    filtrado
    var filtrarPor = e.target.value;
    console.log("filtrar por vale: " + filtrarPor);
    if (filtrarPor == "Todos" && !todosLosDatosCargados) {
        //si el usuario ha filtrado por "Todos" y no estan ya cargados
        todos los datos entonces muestro todos los datos
        llamadaAjax("ObtenerJuegosBBDD.php", "",
            function(texto) {
                productosInicio = JSON.parse(texto);
                // debugger
                for (var i=0; i<productosInicio.length; i++) {
                    insertarVideojuego(productosInicio[i]);
                }
                todosLosDatosCargados = true;
            }
        );
        document.getElementById("selectTipos").addEventListener("click",
            realizarFiltro, false);
    },
    function(texto) {
        // alert(productosInicio);
        notificarUsuario("Error Ajax al cargar al inicializar:
" + texto);
    }
    );
    } else {
        console.log("filtro de: " + filtrarPor)
        //si el usuario a seleccionado un filtro determinado aplico la
        busqueda segun dicho filtro
        llamadaAjax("ObtenerJuegosFiltrados.php?filtro="+filtrarPor,
        "",
            function(texto) {
                var videojuego = JSON.parse(texto);

                for (var i=0; i<videojuego.length; i++) {
                    insertarVideojuego(videojuego[i]);
                }
                addVideojuegoSelectFiltro("selectTipos",videojuego[i]); //-----
                -----
            }
        ),
        function(texto) {
            // alert(productosInicio);
            notificarUsuario("Error Ajax al cargar al inicializar:
" + texto);
        }
        );
    }
}

```

PARTE 3: La función realizarFiltro realiza una llamada AJAX al script PHP ObtenerJuegosBBDD y otra llamada AJAX al script PHP ObtenerJuegosFiltradosBBDD con el valor de un option seleccionado para mostrar los videojuegos de una categoría en concreto.

Si el filtro de categorías tiene como valor por defecto realizará la primera llamada AJAX, en caso contrario realizará la llamada AJAX de filtrado.

Catálogo

Filtrar por Tipo

Todos ▼

Todos

Survival-Horror

Plataformas-2D

Arcade

Después, para la creación de las tarjetas de cada juego, se obtiene mediante JS el div principal de la página llamado game-container, en el cual se crea un div principal de la tarjeta. Dentro del div de la tarjeta se van creando distintos divs en los que en cada uno está organizada la información de los datos del juego obtenidos de la base de datos.

Primero se crea un div donde se insertará el logo del videojuego (a partir del nombre de la imagen guardada en la base de datos), luego es creado un div de descripción del videojuego, donde se añade una breve descripción, el precio actual y el precio anterior (si el videojuego no posee ninguna oferta, el precio anterior no se muestra) del videojuego. Por último se añadirá en el div principal un botón con el texto “Ver Ficha” para acceder a la ficha de dicho videojuego y poder comprarse.

Al ser creados todos los elementos por JavaScript, todos los elementos en su creación tienen asignados una clase específica la cual, en la página principal tiene un link asignado a un fichero CSS concreto el cual da todo el formato y estilo a la tarjeta.



- Gestión de los videojuegos del usuario: compra

El sistema contempla la adquisición de videojuegos por parte de los usuarios. Cada videojuego presente en la plataforma web tiene una ficha asociada; dicha ficha permite la opción de “compra”.

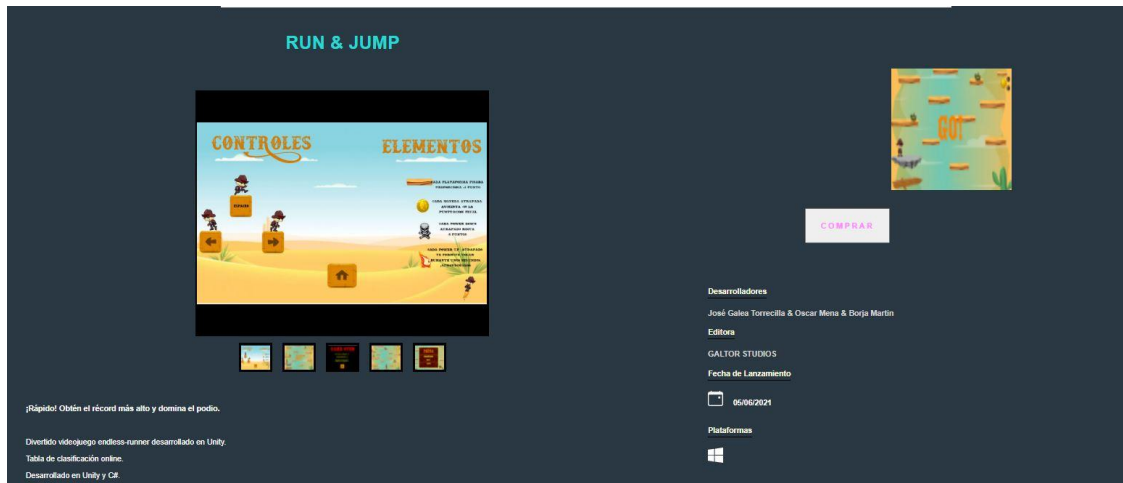


Imagen de una de las fichas de los videojuegos. A la derecha, el botón de compra.

Cuando el usuario realiza la adquisición, el sistema envía el id del videojuego comprado y valida mediante llamadas a la BBDD que el usuario no disponga ya de este producto (en cuyo caso se le informaría y la acción no se llevaría a cabo).

En caso de que sea la primera vez que intenta adquirir el videojuego, el sistema procederá a añadir el id del mismo a la lista de ids de los videojuegos adquiridos por el usuario. De esta forma, cada usuario registrado en la plataforma podrá adquirir los videojuegos que estime oportunos de forma eficiente y sin comprometer el rendimiento del sistema.

De cualquier modo, si se produjese algún error durante la operación de compra por parte del usuario el sistema le informaría de ello.

```

//OBTENGO EL ID DEL VIDEOJUEGO QUE SE VA A COMPRAR
$idVideojuegoaAnnadir = (int) $_REQUEST["idVideojuego"];

//OBTENGO LOS VIDEOJUEGOS QUE EL USUARIO YA HA COMPRADO
$sidsVideojuegos = DAO::misVideojuegoObtenerTodos(DAO::usuarioObtenerId(), sololdsJuegos: true);

$stringCodJuegos = "";
foreach ($sidsVideojuegos as $idActual) {
    $stringCodJuegos .= $idActual.",";
}

//echo "Lo que mando a la BBDD es: ". $stringCodJuegos;

$yaComprado = false;
foreach ($sidsVideojuegos as $idJuegoActual) {
    if ($idJuegoActual == $idVideojuegoaAnnadir) {
        //SI YA LO TIENE AÑADIDO ES QUE YA LO HA COMPRADO
        $yaComprado = true;
    }
}

//SI NO LO TENIA YA COMPRADO ENTONCES:
if (!$yaComprado) {
    //AÑADO EL NUEVO JUEGO AL LISTADO DE VIDEOJUEGOS COMPRADO
    $stringCodJuegos .= $idVideojuegoaAnnadir;
    //ACTUALIZO EL NUEVO LISTADO EN LA BASE DE DATOS
    if (DAO::videojuegoAnnadirIdAUsuario($stringCodJuegos,DAO::usuarioObtenerId())) {
        redireccionar( url: "JuegoCompradoExito.php");
    } else {
        redireccionar( url: "JuegoCompraError.php");
    }
}

//echo json_encode($resultado);
} else {
    redireccionar( url: "JuegoYaCompradoDeAntes.php");
}
}

```

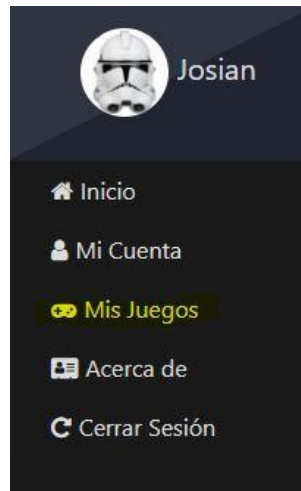
Captura del código encargado de validar si el usuario posee ya el videojuego que pretende adquirir. Como se puede observar, dependiendo del resultado de la operación el usuario será redirigido a una página o a otra.



Captura del sistema informando de la correcta adquisición del producto por parte del usuario.

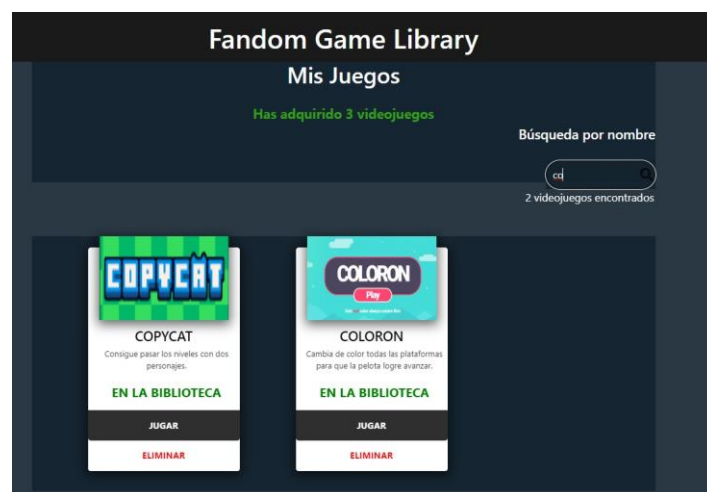
- Gestión de los videojuegos del usuario: eliminación

Además de adquirir nuevos videojuegos, la plataforma web permite que los usuarios eliminen de entre los que han adquirido los videojuegos que estimen oportuno. Para ello, cada usuario dispone de un espacio en el que poder interactuar con sus productos (ejecutar los videojuegos, eliminarlos, etc). Este espacio se encuentra en el apartado “Mis Juegos”.



Menú desplegable del usuario “Josian”

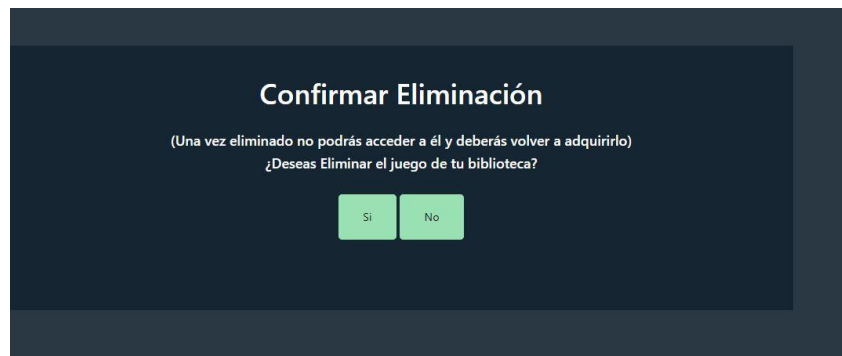
En este apartado, el usuario puede acceder a todos los videojuegos que ha adquirido. Para facilitarle la búsqueda y como enriquecimiento del espacio el sistema provee de un filtrado por nombre mediante el cual el usuario puede buscar videojuegos concretos sin necesidad de moverse por todos sus productos.



Captura del filtrado de “Mis Juegos”

Cada videojuego presente en este apartado tendrá dos opciones: “JUGAR” y “ELIMINAR”.

En caso de que el usuario decida eliminar el videojuego, la plataforma web le solicitará confirmación y, en caso de que quiera proceder, el sistema realizará la operación inversa a la llevada a cabo para la adquisición; es decir, se obtiene el listado de ids de los videojuegos adquiridos por el usuario a través de consultas a la BBDD y se localiza el id del videojuego a eliminar. Una vez hecho esto, se procede a retirar el id de dicho videojuego de la lista del usuario y a informarle del resultado de la operación.



Captura del sistema pidiendo confirmación de eliminación al usuario. En caso de que el usuario pulse el botón de “No”, se le redirigirá de nuevo al espacio de “Mis Juegos”



Captura del sistema informando al usuario de la correcta eliminación del producto.

```

if (isset($_REQUEST["id"])){
    $idJuego = $_REQUEST["id"];
    $resultado = DAO::misJuegosBorrar(DAO::usuarioObtenerId(),$idJuego);

    $mensaje = "";
    if ($resultado){
        $mensaje = "VIDEOJUEGO ELIMINADO CORRECTAMENTE";
    } else {
        $mensaje = "ERROR, no ha sido posible eliminar el videojuego";
    }

} else {
    redireccionar( url: "MisJuegos.php");
}

?>

```

Captura del código encargado de eliminar el videojuego seleccionado por el usuario.

```

public static function misJuegosBorrar(int $idUser, int $idJuego): bool
{
    $datos = Self::misVideojuegoObtenerTodos($idUser, $solidsJuegos: true);

    for($i=0;$i<count($datos);$i++){
        if($datos[$i] == $idJuego){
            unset($datos[$i]);
        }
    }

    //Contrario de explode hacer string
    $stringIDS = implode( separator: ',', $datos);

    $borrar = Self::videojuegoAnnadirIdAUsuario($stringIDS,$idUser);

    if ($borrar){
        return true;
    } else {
        return false;
    }
}

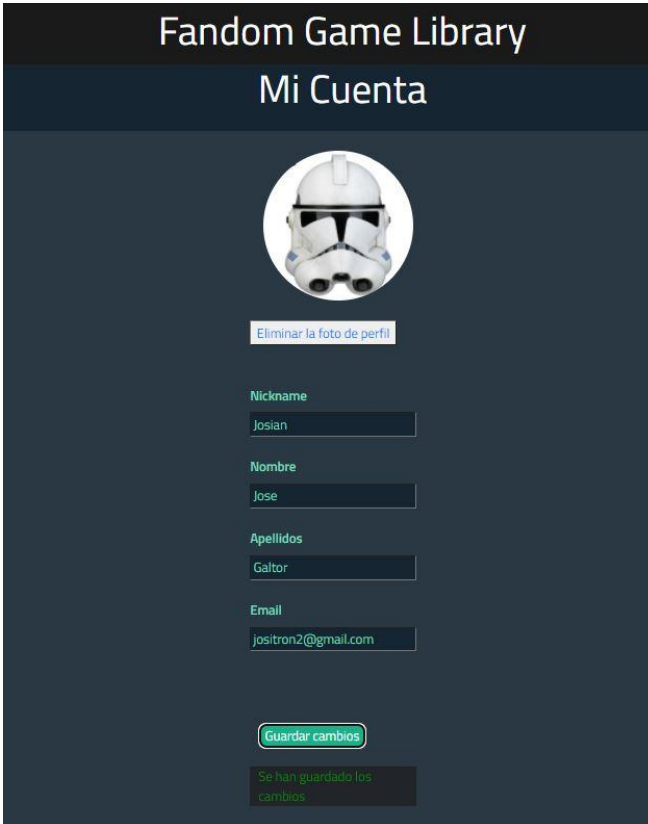
```

Captura del contenido del método “misJuegosBorrar”

- Ajustes de usuario

Para modificar los datos personales del usuario (nombre, apellido, email, etc.) la plataforma facilita un apartado de ajustes llamado “Mi Cuenta”. En este apartado se muestran los datos actuales del usuario en diferentes elementos “input” que el propio usuario podrá modificar.

Una vez que el usuario haya terminado de modificar sus datos personales tendrá la posibilidad de guardarlos, actualizando así el sistema, pulsando el botón “Guardar cambios”. Una vez hecho esto la plataforma informará del resultado al usuario sin que este tenga que actualizar la página.



Captura del sistema informando de la correcta actualización de los datos por parte del usuario.

Para asegurar el correcto funcionamiento del sistema, la plataforma realiza una serie de validaciones con los datos del usuario. Para ello, comprueba que no existan ya usuarios con el nickname ni con el correo introducido consultando los registros existentes en la BBDD del sistema. En caso de que el email no esté en uso, el sistema también realiza una validación de formato mediante el uso de expresiones regulares para garantizar que sea un dato válido.

```
function validarEmail(email) {
    if (/^([a-zA-Z0-9_\.\\-])+\@((([a-zA-Z0-9\\-])+\.)+([a-zA-Z0-9]{2,4})+)$/i.test(email)){
        console.log("La dirección de email " + email + " es valida.");
        return true;
    } else {
        console.log("El email introducido no es valido.");
        return false;
    }
}
```

Método encargado de validar el formato del email introducido por el usuario

```
function comprobarEmail() {
    email = document.getElementById("email").value;
    //Antes de nada compruebo que el texto introducido sea valido como email
    if (validarEmail(email)) {
        llamadaAjax("ComprobarEmailAlActualizarlo.php?email=" + email, "",
            function (texto) {
                // debugger
                var existeEmail = JSON.parse(texto);
                console.log("Esto envio: ComprobarEmailAlActualizarlo.php?email=" + email);
                console.log("recibo de PHP esto: " + existeEmail);
                if (existeEmail) {
                    textoAvisoEmail.innerHTML = "El email ya esta en uso, introduzca otro";
                    textoAvisoEmail.style.color = "red";
                    actualizarDatosUsuario(false);
                } else {
                    textoAvisoEmail.innerHTML = "";
                    //Si la validacion del email es correcta entonces procedo a validar el nickname
                    comprobarNickName();
                }
            },
            function (texto) {
                notificarUsuario("Error Ajax: " + texto);
            });
    } else {
        textoAvisoEmail.innerHTML = "El email introducido no es valido";
        textoAvisoEmail.style.color = "red";
    }
}
```

Captura del método JS encargado de la validación completa del email

```
public static function existeUsuarioConEsteNickName($nickname)
{
    $id = $_SESSION["id"];
    //Este metodo devuelve true si existe un usuario con el nickname especificado y false si no
    $rs = Self::ejecutarConsulta(
        sql: "SELECT id FROM usuario WHERE identificador = ? AND id != ?",
        [$nickname, $id]
    );
    if (empty($rs)) {
        return false;
    } else {
        return true;
    }
}
```

Captura del método PHP que consulta en la BBDD del sistema si el nickname introducido ya está en uso.

En caso de que no se cumplan los requisitos para la modificación de los datos el sistema informará al usuario de ello en tiempo real vía AJAX sin necesidad de actualizar la página.

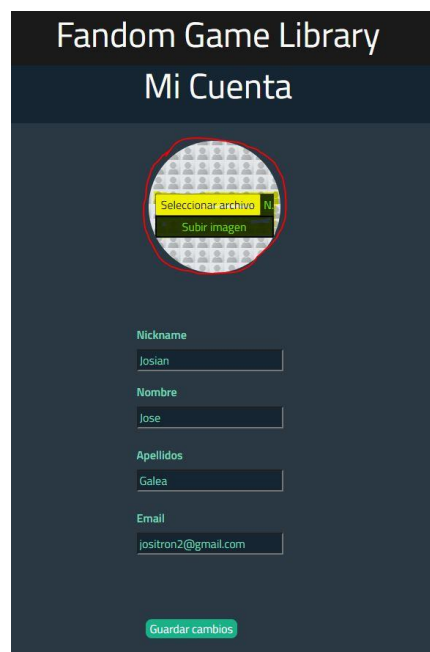
En esta situación, los datos introducidos por el usuario son rechazados por el sistema y la BBDD no se modifica.



A screenshot of a user profile update form. The form has a dark blue background. It contains several input fields: 'Nombre' with the value 'Jose', 'Apellidos' with the value 'Galtor', and 'Email' with the value 'usuario@gmail.com'. Below the email field, there is a red error message that reads 'El email ya está en uso, introduzca otro'. At the bottom of the form, there is a green button labeled 'Guardar cambios' and a yellow message box that says 'No se ha modificado ningún dato'.

Captura del sistema informando al usuario de que el email introducido ya está en uso

Además de la información personal, el sistema también contempla el uso de una foto de perfil por parte del usuario.



A screenshot of the 'Mi Cuenta' (My Account) section in the Fandom Game Library. The page has a dark blue header with the text 'Fandom Game Library' and 'Mi Cuenta'. Below the header, there is a circular profile picture placeholder with a grid of small icons. A red circle highlights the placeholder, and a yellow button labeled 'Seleccionar archivo' and a green button labeled 'Subir imagen' are visible. Below the profile picture, there are several input fields for personal information: 'Nickname' with the value 'Josian', 'Nombre' with the value 'Jose', 'Apellidos' with the value 'Galea', and 'Email' with the value 'jositron2@gmail.com'. At the bottom of the form, there is a green button labeled 'Guardar cambios'.

Captura del apartado de "Mi Cuenta" que permite seleccionar una imagen de perfil.

Cuando el usuario selecciona una imagen de su dispositivo y pulsa el botón de “Subir imagen” el sistema ejecuta, vía AJAX y de forma totalmente transparente para el usuario, el archivo “SubirFotoUsuario.php” que recibe el archivo a subir y, tras comprobar que tenga el tamaño y la extensión adecuados, almacena la imagen en la plataforma web.

Para garantizar la correcta organización de las imágenes de cada usuario la plataforma renombra la imagen del usuario por el id del usuario seguido por la extensión “.PNG”. De esta forma cada vez que un usuario modifica su foto de perfil, el sistema crea un archivo con las condiciones mencionadas para que en caso de que ya exista una imagen con el mismo nombre se sobrescriba. Llevando a cabo este procedimiento se garantiza que en la plataforma sólo pueda haber una imagen de perfil por usuario evitando así una sobrecarga de datos innecesaria.

```
//Si la imagen es correcta en tamaño y tipo
//Se intenta subir al servidor con el nombre del id del usuario y con "png" como extension para organizarme
if (move_uploaded_file($temp, '../Imagenes/Usuarios/'.$_SESSION["id"].".png")) {
    //Cambiamos los permisos del archivo a 777 para poder modificarlo posteriormente
    chmod( filename: 'images/'.$archivo, permissions: 0777);

    //actualizamos la base de datos
    if (DAO::usuarioAñadirFotoPerfil($_SESSION["id"])) {
        redireccionar( url: "Perfil.php?archivo");
    } else {
        redireccionar( url: "Perfil.php?errorSubirArchivo");
    }
}
```

Captura del apartado del fichero “SubirFotoUsuario.php” que se encarga de almacenar la imagen del usuario en el servidor.



Captura de la plataforma informando al usuario que la imagen seleccionada no cumple los requisitos del sistema.



Captura del sistema actualizando la imagen de perfil del usuario. Como se puede observar en la captura, al subir la imagen al servidor de forma correcta, el sistema proporciona la posibilidad de eliminarla en caso de que el usuario quiera subir una nueva o quitar la existente.

- Página Oculta

En la aplicación web existe una página a la que no se puede acceder por medio del menú lateral sino introduciendo una URL concreta. En dicha página se muestran todos los usuarios y juegos que existen en la base de datos.

Fandom Game Library

4 Usuarios registrados

Identificador: Iverma Nombre: Ivan Apellidos: Delas Email: dbdtodoeldia@gmail.com
Identificador: Josian Nombre: Josian Apellidos: Galtor Email: jositron2@gmail.com
Identificador: ManCa Nombre: Manuela Apellidos: Carrasco Email: carrasMan@gmail.com
Identificador: pLotudo Nombre: Pepe Apellidos: Lotudo Email: pepelot@gmail.com

© 2022 The FGL Association
Todos los derechos reservados a sus respectivos dueños.

Fandom Game Library

14 videojuegos

Nombre: COLORON
Nombre: COPYCAT
Nombre: EL AHORCADO
Nombre: INFINITE RUNNER
Nombre: MARBLE LABYRINTH
Nombre: MAZE GRIDPURE
Nombre: OUTRUN
Nombre: PLATFORMS GAME
Nombre: PLATFORMS WORLD
Nombre: RUN & JUMP
Nombre: STICK HERO
Nombre: SWEET MEMORY
Nombre: THE DUNGEON GAME

© 2022 The FGL Association
Todos los derechos reservados a sus respectivos dueños.

```
$videojuegos = DAO::videojuegoObtenerTodos();
$usuarios = DAO::usuariosObtenerNombres();
$numUsuarios = sizeof($usuarios);
$numJuegos = sizeof($videojuegos);
```

Para ello se realiza dos llamadas a la base de datos, una obtiene todos los usuarios y la otra todos los videojuegos, al haber recogido todos los datos, mediante bucles foreach se recorren los datos y se muestran en la pantalla con la función echo.

```
<h3><?=$numJuegos?> videojuegos</h3>
<?php
    foreach ($videojuegos as $videojuegoActual) {
        echo "<div style='border-color: #1ab188;
            border-style: solid'>";
        echo "<br>";
        echo "<li>Nombre: <b>".$videojuegoActual-
            >getNombre()."</b></li>";
        echo "<br>";
        echo "</div>";
    }
?>

<h3><?=$numUsuarios?> Usuarios registrados</h3>
<?php
    foreach ($usuarios as $usuarioActual) {
        echo "<div style='border-color: #ec7600; border-style:
            solid'>";
        echo "<br>";
        echo "<li>Identificador: <b>".explode(";",
            $usuarioActual)[0]."</b></li>";
        echo "<br>";
        echo "<li>Nombre: ".explode(";",
            $usuarioActual)[1]."</li>";
        echo "<br>";
        echo "<li>Apellidos: ".explode(";",
            $usuarioActual)[2]."</li>";
        echo "<br>";
        echo "<li>Email: <b>".explode(";",
            $usuarioActual)[3]."</b></li>";
        echo "<br>";
        echo "</div>";
    }
?>
```


- Modo Claro/Oscuro

Para mejorar el aspecto visual de la plataforma web se ha implementado la opción de cambiar entre dos apariencias: modo claro y modo oscuro. Dicho cambio se puede llevar a cabo en cada una de las páginas web de la plataforma. Al realizar el cambio en una de ellas dicha modificación se hace extensible al resto.

En la BBDD del sistema, se almacena el registro “modo” para cada usuario; dicho registro alterna su contenido entre “claro” y “oscuro” según el usuario haya seleccionado un modo u otro. De esta forma, el modo seleccionado por cada usuario es almacenado de manera persistente para que tras cerrar sesión y volver a iniciarla dicho usuario conserve el modo que seleccionó.

Para gestionar esta implementación el sistema utiliza un fichero CSS llamado “darkMode” que, en conjunto con un fichero JS que presenta el mismo nombre, se encarga de dar a cada página su correcta apariencia.

```
function inicializar() {  
    let themeToggler = document.getElementById('theme-toggler');  
  
    if (themeToggler.className == "fas fa-moon fa-sun") {  
        console.log("el modo oscuro esta activado");  
        darkMode();  
    } else if (themeToggler.className == "fas fa-moon") {  
        console.log("el modo claro esta activado");  
        lightMode();  
    } else {  
        console.log("algo raro pasa. Ningún modo activado");  
    }  
}
```

Método del fichero “DarkMode.js” que se ejecuta al cargar la página web y que, dependiendo de la clase HTML encontrada, llamará al método encargado de cargar el modo claro o al encargado de cargar el modo oscuro.

```

let themeToggler = document.getElementById('theme-toggler');

themeToggler.onclick = () => {
  themeToggler.classList.toggle('fa-sun');

  if (themeToggler.classList.contains('fa-sun')) {
    almacenarCambioModoClaroOscuro("oscuro");
    darkMode();
  } else {
    almacenarCambioModoClaroOscuro("claro");
    lightMode();
  }
};

```

Método presente en el fichero “DarkMode.js” encargado de capturar el evento en el que el usuario cambia de modo para actualizar la BBDD del sistema con la nueva información y llamar al método encargado de cargar el nuevo modo seleccionado.



Capturas de pantalla del botón encargado de modificar el modo claro/oscuro indicando con sus distintas apariencias el modo seleccionado. Cuando el usuario pulsa el botón se desencadena el evento mencionado en la captura de código anterior.

5 | PRUEBAS

Tabla 3. Recopilación de pruebas llevadas a cabo

RF (Requisito Funcional)	Prueba realizada	Resultado Obtenido	Resultado aceptado
RF-01 (Inicio de sesión)	Se inicia sesión en la plataforma web con un usuario de prueba.	Se accede al menú principal de la plataforma con el usuario de prueba con éxito.	TRUE
RF-02 (Menú desplegable)	Una vez en el menú principal de la plataforma web, se procede a desplegar el menú y a comprobar que los enlaces funcionan correctamente.	El menú se despliega correctamente y los enlaces funcionan como se espera.	TRUE
RF-03 (Cerrar sesión)	Se cierra la sesión del usuario de prueba.	La sesión se cierra con éxito y se redirige al usuario al inicio de sesión	TRUE
RF-04 (Fichas de videojuegos)	Se accede a la ficha de cada videojuego para comprobar que la disposición de los elementos y su contenido es el adecuado.	Las fichas presentan la disposición deseada y el contenido es el correcto.	TRUE
RF-05 (Adquirir videojuegos)	Se comprueba si los distintos usuarios pueden adquirir videojuegos y si el resultado de esta operación se almacena correctamente.	Cada usuario puede adquirir nuevos videojuegos. Los datos pertinentes son almacenados en la BBDD de forma correcta. Los usuarios pueden acceder a los videojuegos adquiridos mediante el apartado correspondiente.	TRUE
RF-06 (Eliminar videojuegos)	Se comprueba si los distintos usuarios pueden eliminar los videojuegos previamente adquiridos y si el resultado de esta operación se almacena correctamente.	Cada usuario puede eliminar los videojuegos adquiridos que considere oportunos. Los datos pertinentes son almacenados en la BBDD de forma correcta. Los usuarios pueden volver a adquirir el videojuego eliminado en el apartado correspondiente.	TRUE

RF-07 (Búsqueda de videojuegos)	Se comprueba si los métodos de búsqueda de videojuegos funcionan correctamente.	El filtrado por categoría funciona correctamente. La búsqueda por nombre funciona correctamente.	TRUE
RF-08 (Envío correo electrónico)	Se comprueba que el envío de emails funciona correctamente creando un usuario de prueba.	El email introducido para el usuario de prueba recibe el correo notificando el acceso a la plataforma correctamente.	TRUE
RF-09 (Creación nuevas cuentas)	Se comprueba que se pueden crear nuevas cuentas respetando las condiciones de validación establecidas y que estas se integren de forma eficiente en el sistema creando una cuenta de usuario de prueba.	La cuenta se crea correctamente. La validación de datos reacciona de la manera esperada. La información se almacena en la BBDD como se estableció.	TRUE
RF-10 (Ajustes de usuario)	Se comprueba que los usuarios puedan modificar los datos de su cuenta.	Los datos modificados por el usuario se guardan y se actualizan correctamente	TRUE
RF-11 (Recordar contraseña usuario)	Se comprueba que la plataforma web permita modificar la contraseña y recuperar la cuenta a los usuarios que así lo quieran.	La contraseña se resetea y se actualiza con los nuevos datos de forma correcta.	TRUE
RF-12 (Modo Claro/Oscuro)	Se comprueba que los usuarios puedan cambiar la apariencia de la plataforma web de claro a oscuro y viceversa y que sea una operación individual para cada usuario.	El usuario puede cambiar el modo de claro a oscuro correctamente. La operación se realiza de manera individual para cada usuario.	TRUE

6 | TRABAJO FUTURO

Dentro del trabajo futuro en este proyecto pueden destacarse varias implementaciones:

- **Adquirir un servicio de Hosting:** De esta forma, la plataforma web estará disponible al público y se podrá acceder en cualquier momento.
- **Adquisición de puntos por parte del usuario:** Los usuarios podrán obtener puntos por horas de actividad o comprarlos en el apartado de “Comprar puntos”.
- **Aplicación de Escritorio para Administradores:** Los administradores podrán gestionar videojuegos y usuarios de forma segura y transparente. La aplicación estará sincronizada con la plataforma web.
- **Foro de usuarios:** La plataforma web presentará un espacio dedicado a responder las posibles preguntas de los usuarios relativas al funcionamiento de la plataforma web o de los videojuegos que contiene.

7 | CONCLUSIONES

Este capítulo se dedica a conclusiones finales y personales del proyecto, extraídas a lo largo de todo el proceso de desarrollo del mismo.

Se han cumplido todos los objetivos y requisitos mencionados en los capítulos anteriores. El principal de ellos era crear una plataforma web que fuera estable y funcional con un rendimiento óptimo capaz de ofrecer un objetivo a cumplir y unos medios para llevarlo a cabo.

Hemos profundizado en las tecnologías y lenguajes de programación ya vistos a lo largo del grado (PHP, JavaScript, CSS, HTML, PhpStorm, GitHub, etc...) por lo que la implementación del proyecto, una vez comprendidas y valoradas las necesidades del proyecto, ha sido relativamente sencilla.

Además de haber aprendido a desarrollar un proyecto de principio a fin, pasando por todas sus fases y adquiriendo nuevos conocimientos acerca de la programación web; hemos hecho hincapié en la optimización y sencillez del código asemejándolo lo máximo posible a un proyecto real.

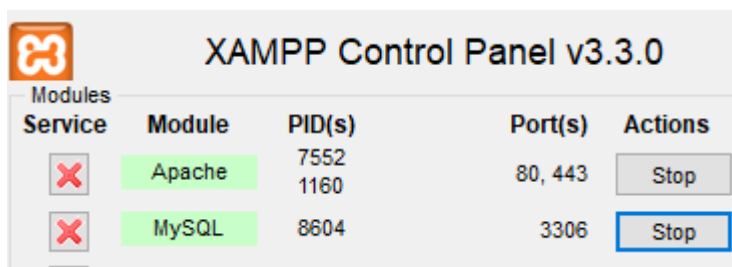
8 | ANEXO

8.1 CONFIGURACIÓN Y DESPLIEGUE



Para el correcto funcionamiento del software, se deberá instalar XAMPP, el cual es una distribución de Apache con el cual se puede albergar un servidor web apache o sistemas relacionales de administración de bases de datos como MySQL.

Una vez realizada la instalación por defecto de XAMPP, se debe albergar el software en la ruta **C:\xampp\htdocs**, esto permitirá poder ejecutar el software en el servidor web apache.

Después, se ejecutará XAMPP y se activarán los siguientes servicios:



The screenshot shows the XAMPP Control Panel v3.3.0 interface. It features a table with columns for Service, Module, PID(s), Port(s), and Actions. Two services are listed: Apache and MySQL. Both have a red 'X' icon in the Service column, indicating they are not running. The Apache service is running on ports 80 and 443 with PIDs 7552 and 1160. The MySQL service is running on port 3306 with PID 8604. Both services have a 'Stop' button in the Actions column.

Service	Module	PID(s)	Port(s)	Actions
	Apache	7552 1160	80, 443	Stop
	MySQL	8604	3306	Stop

Una vez activados, en un navegador deberá escribir en la barra de búsqueda superior **http://localhost/dashboard/** y pulsar en la opción **phpMyAdmin**. Al entrar podrás gestionar las bases de datos que posees.

Para importar la base de datos del proyecto, deberá entrar en la carpeta del proyecto y seguir esta ruta: **C:\xampp\htdocs\TFG\Proyecto__Documentacion**, habrá un archivo llamado **BBDD_TFG.sql**. Una vez localizado, volviendo al navegador en la pestaña anterior existirá un botón llamado **Importar**, al pulsar seleccionamos el archivo mencionado anteriormente y pulsamos **Continuar**, una vez finalizado, ya tendremos la base de datos creada con toda su estructura.

Después, para ejecutar el proyecto, en un navegador realizaremos la siguiente búsqueda en la barra superior:
http://localhost/TFG/Proyecto/GestionUsuario/SesionUsuario/SesionFormulario.php

Esta ruta nos permitirá iniciar en el formulario de Inicio de Sesión.

Para que XAMPP pueda enviar correos a las cuentas, se deberán modificar ficheros del propio XAMPP los cuales son los siguientes:

C:\xampp\php\php.ini

Encontrar la línea: **;extension=php_openssl.dll** y quitarle el punto y coma

Encontrar las líneas **SMTP, smtp_post, sendmail_from** y **sendmail_path** y poner lo siguiente si utilizarás Gmail como smtp:

- **SMTP=smtp.gmail.com**
- **smtp_port=587**
- **sendmail_from = tu_direccion_de_correo_electronico@gmail.com**
- **sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"**

C:\xampp\sendmail\sendmail.ini y configurarlo de la siguiente manera:

- **smtp_server=smtp.gmail.com**
- **smtp_port=587**
- **error_logfile=error.log**
- **debug_logfile=debug.log**
- **auth_username=tu_direccion_de_correo_electronico@gmail.com**
- **auth_password=el_password_de_la_direccion_de_correo_electronico**
- **force_sender=tu_direccion_de_correo_electronico@gmail.com**

Si utilizas Gmail para el envío de correos de manera local debes tener el cuenta que debes activar la opción "Permitir que aplicaciones menos seguras accedan a tu cuenta" de lo contrario no podrás enviar los correos electrónicos con Gmail, dejo el link a la documentación

<https://support.google.com/accounts/answer/6010255?hl=es-419>

9 | BIBLIOGRAFÍA

1. Documentación Modulo Desarrollo Web en Entorno Servidor
2. Documentación Modulo Desarrollo Web en Entorno Cliente
3. Documentación Modulo Diseño de Interfaces Web
4. [Documentación Oficial PHP](#)
5. [Documentación Oficial JavaScript](#)
6. [Documentación Xampp](#)
7. [Configuración de envío de correos Xampp vía Gmail](#)
8. [W3Schools](#)
9. [Documentación Oficial CSS](#)