

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the text '2º DAW'. Below the banner, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

2º DAW

TRABAJO DE FIN DE GRADO

DESARROLLO DE APLICACIONES WEB

Juan Piñero Burguera
IES LA VEREDA

L4P

LOOKING 4 PLAYERS

JUAN PIÑERO BURGUERA

ÍNDICE

1. Introducción.....	
1.1. Descripción del proyecto.....	
1.2. Módulos a los que implica	
1.3. Objetivos del proyecto.....	
1.4. Diagrama de Gantt.....	
1.5. Motivación del proyecto.....	
2. Desarrollo de la Web.....	
2.1. Tecnologías usadas.....	
2.1.1. FrontEnd.....	
2.1.2. BackEnd.....	
2.1.3. Base de datos.....	
2.1.4. Servidor.....	
3. Estructura y funcionalidades.....	
3.1. Capa de persistencias.....	
3.2. Capa lógica.....	
3.3. Capa de presentación.....	
4. Manal de usuario.....	
4.1. Manual de Administrador.....	
4.2. Manual de Usuario.....	
5. Conclusión.....	
5.1. Problemas encontrados.....	
5.2. Mejoras.....	
6. Bibliografía.....	

LISTADO DE ILUSTRACIONES

Ilustración 1. Diagrama de Gantt

Ilustración 2. Diagrama Entidad Relación

Ilustración 3. Tabla Usuario

Ilustración 4. Tabla Post

Ilustración 5. Tabla Juego

Ilustración 6. Tabla Juega

Ilustración 7. Tabla Gusta

Ilustración 8. Tabla Reporta

Ilustración 9. Código de registro

Ilustración 10. Código de login

Ilustración 11. Código de post

Ilustración 12. Estructura Twig

Ilustración 13. Extends base.twig

Ilustración 14. Login manual

Ilustración 15. Login error datos manual

Ilustración 16. Registro manual

Ilustración 17. Error Registro manual

Ilustración 18. Menú administrador Manual

Ilustración 19. Menú Reportes Manual

Ilustración 20. Reportes Juegos Manual

Ilustración 21. Reportes Inapropiados Manual

Ilustración 22. Verificar manual

Ilustración 23. Error Verificar Manual

Ilustración 24. Banear Manual

Ilustración 25. Alta Juego Manual

Ilustración 26. Error Alta Juego Manual

Ilustración 27. Escoger Juegos Manual

Ilustración 28. Añadir Juegos Manual

Ilustración 29. Menú Manual

Ilustración 30. Índice Manual

Ilustración 31. Contenido Post Manual

Ilustración 32. Ejemplo post Manual

Ilustración 33. Popup imagen manual

Ilustración 34. Hilo Manual

Ilustración 35. Desplegable Hilo Manual

Ilustración 36. Responder manual

Ilustración 37. Desplegable respuesta manual

Ilustración 38. Menú reportar manual

Ilustración 39. Desplegable reporte manual

Ilustración 40. Añadir Juegos usuario manual

Ilustración 41. Perfil usuario

Ilustración 42. Post perfil manual

Ilustración 43. Perfil otro user manual

Ilustración 44. Perfil juego manual

Ilustración 45. Nuevos Juegos manual

1- Introducción

1.1- Descripción del proyecto

Hay muchas **redes sociales** enfocadas en diversos tipos de temas: **Instagram** se enfoca en subir imágenes y vídeos con múltiples efectos fotográficos, **Tinder** en que sus usuarios puedan comunicarse con otras personas basándose en sus preferencias para charlar y concretar citas o encuentros, **WhatsApp** para enviar mensajes de texto y multimedia entre sus usuarios, etc. Pero no hay ninguna red social popular enfocada en un mundo tan expandido y emergente como los videojuegos. Esta es la esencia de **L4P**, hacer una plataforma donde jugadores de todo tipo puedan adentrarse a la comunidad de sus juegos favoritos y conocer gente nueva con la cual interactuar tanto fuera como dentro de ellos.

L4P es una red social donde una vez el usuario se haya registrado e iniciado sesión podrá seleccionar sus juegos favoritos e interactuar con gente con los mismos gustos. Además, podrá realizar posts, dar me gusta a otros, e incluso comentar en ellos.

L4P tiene un sistema de interacción donde puedes conocer gente, chatear con ellos, jugar y dar opinión sobre tu experiencia con ese jugador, así se evita al máximo las malas prácticas de las redes sociales convencionales.

1.2- Módulos a los que implica

L4P está centrado en el FullStack así que se utiliza lo aprendido tanto en Entorno Servidor como en Entorno Cliente. Por la parte del servidor, estructurarlo todo de manera precisa dentro de la jerarquía del proyecto, conseguir un sistema eficiente con la base de datos y además, seguro ha sido lo más determinante. Y por la parte de cliente saber manejar DOM ha sido lo más importante. A la hora del diseño de la web he utilizado lo aprendido en Diseño de Interfaces Web teniendo en cuenta que está enteramente hecho en Bootstrap. Para organizarme y estructurar todo lo aprendido, Entornos de primer año me ha sido bastante útil. Por último, Despliegues me ha servido para instalar y utilizar las herramientas más precisas para este proyecto.

Cabe destacar, que el haber realizado la dual ha sido determinante a la hora de agilizar la realización del proyecto y desenvolverme en ámbitos desconocidos.

1.3- Objetivos del proyecto

Este proyecto se centra en la idea de crear una red social donde se genere una comunidad de gente del mundo del videojuego y permitir a gente nueva introducirse de mejor manera.

L4P es una aplicación centrada en unos objetivos concretos que se complementan los unos a los otros, estos objetivos son:

- **Ayudar** a los usuarios que no tiene con quien jugar a que conozcan gente y consiga encontrar un grupo con el cual disfrutar más de los juegos que le gustan.
- Tener una **plataforma** con la cual informarte de todo lo relacionado con los videojuegos.
- El usuario se tiene que ver envuelto en un mundo donde solo haya información relacionada con los **videojuegos**.
- El usuario va a conocer **noticias** y va a leer **posts** de los juegos que más le gustan

En la versión de inicio de esta aplicación se verán todos estos objetivos unificados en una red social orientada al máximo en el mundo gaming.

1.4- Diagrama de Gantt

El diagrama de Gantt es una herramienta útil para planificar proyectos. Al proporcionarte una vista general de las tareas programadas, todas las partes implicadas sabrán qué tareas tienen que completarse y en qué fecha.

Este es el diagrama de Gantt realizado para este proyecto:

Actividad / Semana	1	2	3	4	5	6	7	8	9	10	11	12
Objetivos y captación de información	✓	✓										
BBDD y lenguajes			✓	✓								
BackEnd					✓	✓	✓	✓				
login y registro							✓	✓				
FrontEnd								✓	✓	✓		
Diseño									✓	✓		
Documentación											✓	✓
Pruebas											✓	✓

Ilustración 1. Diagrama de Gantt

Se puede observar como está organizada por semanas. Es muy importante tener en cuenta porque se ha escogido este orden de realización de tareas. Cabe señalar que la documentación, aunque solo está expuesta las últimas dos semanas, se ha ido recogiendo datos fundamentales durante la realización de todo el proyecto.

-Es fundamental recaudar información y tener claros los objetivos del proyecto antes de la realización de este. De esta forma se tiene en cuenta mucho más claro la dirección que hay que tomar.

-La estructura principal en la BBDD y el lenguaje que se va a utilizar es muy importante, sin tener esto claro no se puede seguir avanzando.

-Particularmente, el login y el registro han llevado un tiempo elevado con la implementación de la autenticación por correo.

-Previo al front, es muy importante estructurar el back, ya que hay que tener en cuenta todas las llamadas necesarias con la base de datos, como toda la estructura. Para ejemplificarlo mejor, a un cuerpo no le puedes poner el musculo si no tiene esqueleto, pues en este caso el esqueleto es el back.

-Junto al front viene el diseño ya que es necesario complementarlos para comprobar un resultado final de la presentación del proyecto.

-Junto a la realización de la documentación llega lo más importante del proyecto antes de presentarlo: las pruebas. Las pruebas son fundamentales para comprobar el correcto funcionamiento de todas las funciones de la aplicación.

Usar un diagrama de Gantt en el proceso de gestión de proyectos ha proporcionado las siguientes ventajas:

- Claridad.
- Una vista general simplificada.
- Datos sobre el rendimiento.
- Una mejor gestión del tiempo.
- Flexibilidad.

1.5- Motivación del proyecto

Dos de los temas que más están en auge la última década entre la gente joven son las redes sociales y lo videojuegos. Ambas están en su punto más álgido y generando en el mundo expectativas cada vez más altas. Pero, tras un estudio es fácil darse cuenta de que aun estando muy relacionado uno con el otro no existe una plataforma que las junte lo suficientemente popular como para que se pueda tener en cuenta.

Para justificar la realización de este proyecto se va a tener en cuenta una serie de valores que se van a exponer a continuación.

Valor social

La mejora social con este proyecto es más que notoria. Además de incrementar las relaciones sociales, se premia el buen comportamiento dentro de la aplicación. Ánima a personas con dificultades sociales y con una forma de ser introvertida a que conozca gente con la cual relacionarse, teniendo en cuenta que la red social tiene como punto de partida los videojuegos, lo que provoca que ya se tenga algo en común entre los usuarios.

Valor práctico

Este proyecto no está creado sobre algo infundado. Es un proyecto que se podría realizar en una empresa real. El punto más fuerte de este proyecto es que está basado en una necesidad real, ya que muchos usuarios del mundo del videojuego han expuesto la necesidad de un tema como el presentado.

Valor teórico

Este proyecto es una aplicación del cuál puede aprender otras muchas empresas dueñas de redes sociales. El sistema de opinión implementado en el proyecto y, el sistema de reportes es un ejemplo innovador de lo que puedes sacar como valor teórico.

Todos estos valores, los cuales representan una idea totalmente innovadora y funcional han sido determinantes para realizar este proyecto.

2- Desarrollo de la Web

2.1- Tecnologías usadas

Este proyecto está basado en un Framework llamado Symfony. Este es un entorno de trabajo estandarizado que se utiliza para el desarrollo de aplicaciones web y

es de los más utilizados en el entorno de desarrolladores de aplicaciones. En otras palabras, es una herramienta para desarrolladores para crear aplicaciones en **PHP**.

La empresa creadora de este Framework fue SensioLabs y es utilizada actualmente por miles de empresas de desarrollo web en todo el mundo.

Una de las principales ventajas de Symfony es que posee una licencia MIT, o lo que es lo mismo que una licencia de software libre permisivo, que se puede utilizar dentro del software del propietario y al no poseer Copyright permite su modificación.

Pero ¿Por qué Symfony y no otro? Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Además, Symfony está diseñado para cumplir una serie de requisitos fundamentales en el proyecto:

- **Fácil** de instalar y configurar en la mayoría de las plataformas.
- **Independiente** del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza **programación orientada a objetos**, de ahí que sea imprescindible PHP 5.
- **Sencillo** de usar en la mayoría de los casos, está más indicado para grandes aplicaciones Web que para pequeños proyectos.
- Aunque utiliza **MVC** (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Basado en la premisa de “**convenir en vez de configurar**”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de **mejores prácticas y patrones** de diseño para la web
- **Preparado** para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- **Código fácil** de leer que incluye comentarios de **phpDocumentor** y que permite un mantenimiento muy sencillo.
- **Fácil de extender**, lo que permite su integración con las librerías de otros fabricantes.

2.1.1- FrontEnd

En Symfony se utiliza Twig. Twig es un motor de plantillas desarrollado para el lenguaje de programación PHP y que nace con el objetivo de facilitar a los desarrolladores de aplicaciones web que utilizan la arquitectura MVC el trabajo con la parte de las vistas, gracias a que se trata de un sistema que resulta muy sencillo de aprender y capaz de generar plantillas con un código preciso y fácil de leer.

Cuando hablamos de plantilla, nos estamos refiriendo a un archivo de texto que puede arrojar resultados en formatos como HTML, XML, CSV... y que está formado por expresiones de control y variables, las cuales serán reemplazadas por valores una vez que la plantilla sea evaluada.

Cuando nos referimos al lenguaje PHP, una de las plantillas más utilizadas es una plantilla PHP, en la que se mezcla texto interpretado por PHP y en el que se mezclan etiquetas HTML y código PHP para formar la vista que verá el usuario.

En el proyecto se utiliza también Bootstrap. Bootstrap es un framework CSS utilizado en aplicaciones FrontEnd para desarrollar aplicaciones que se adaptan a cualquier dispositivo. El propósito del framework es ofrecerle al usuario una experiencia más agradable cuando navega en un sitio.

Por esta razón, tiene varios recursos para configurar los estilos de los elementos de la página de una manera simple y eficiente, además de facilitar la construcción de páginas que, al mismo tiempo, están adaptadas para la web y para dispositivos móviles.

Por último, utilizaré JavaScript. JavaScript es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes, etc.

2.1.2- BackEnd

PHP es un lenguaje de programación de código abierto del lado del servidor cuyo principal uso es el de crear páginas web dinámicas. Una abreviatura que nació en un principio de “Personal Home Page Tools”, aunque actualmente es el acrónimo de “PHP Hypertext Preprocessor”. Los lenguajes del lado del cliente, como JavaScript o HTML, en primer lugar, son interpretados por el navegador web cuando los usuarios abren una página. Mientras, en PHP, un lenguaje de programación del lado del servidor, el código es ejecutado en el servidor, donde los scripts de PHP generan el código HTML que posteriormente se envía al navegador.

Este lenguaje de programación es ampliamente utilizado hoy en día para el desarrollo de aplicaciones y páginas web dinámicas. Aunque estos son sus principales usos, tiene otros, como la creación de aplicaciones de escritorio. Una de las principales ventajas que ofrece PHP a los desarrolladores es que se puede utilizar en todo tipo de plataformas.

Los beneficios de PHP son principalmente tres. Por un lado, el lenguaje de programación se puede utilizar con cualquier tipo de servidor, y con cualquier SO virtual. Por otro lado, soporta una gran cantidad de bases de datos diferentes. Y, por último, brinda un soporte completo para la comunicación del servidor con otros protocolos.

2.1.3- Base de Datos

En este proyecto se utiliza la base de datos de MariaDB ya que es la que te recomienda utilizar el XAMPP.

MariaDB es un software Open Source cuyas características muchas veces se desconocen y que todo equipo de desarrollo debería tener presente, puesto que mejora en muchas ocasiones el rendimiento que tendríamos con otras alternativas más conocidas y utilizadas tradicionalmente.

Sin duda, MySQL es uno de los motores de bases de datos más conocidos y usados a nivel mundial, compatible con decenas de lenguajes y frameworks. Sin embargo, debido al ir y venir de compras de empresas, ha surgido un problema en los últimos años con este software, y muy relevante para los desarrolladores. Básicamente, al caer en manos de Oracle, MySQL ya no es completamente Open Source, sino que se acerca al software comercial, aunque existe una versión Community Edition que sí se puede usar de manera gratuita. Ante esta situación, se creó MariaDB, que es básicamente una variante de MySQL que mantiene toda la filosofía del software libre y nos asegura dos objetivos muy importantes: mantener la compatibilidad con MySQL, de modo que MariaDB pueda usarse como reemplazo de MySQL sin necesidad de ninguna costosa migración, y que la Comunidad pueda continuar trabajando en el sistema gestor de base de datos, sin verse condicionados por las estrategias comerciales.

2.1.4- Servidor

XAMPP es un servidor independiente de plataforma de código libre. Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD ó FileZilla FTP Serve, etc. entre muchas cosas más.

3- Estructura y Funcionalidades

3.1- Capa de persistencia

Una capa de persistencia encapsula el comportamiento necesario para mantener los objetos. O sea: leer, escribir y borrar objetos en la base de datos. La persistencia de la información es la parte más crítica en una aplicación de software. MariaDB es la base de datos utilizada. A continuación, se observará el diagrama entidad-relación de esta base de datos

Diagrama Entidad-Relación

Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema.

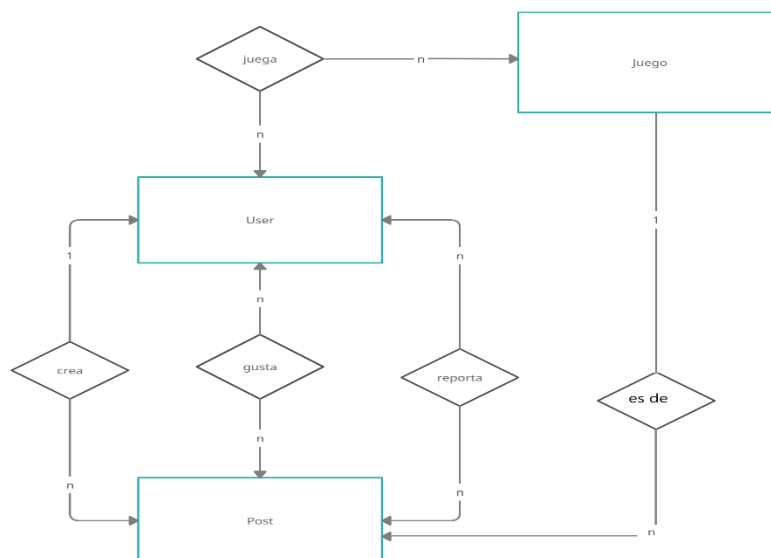


Ilustración 2. Diagrama Entidad Relación

USER		
Atributo	Tipo de parámetro	Descripción
id (PK)	int(11)	Id de la tabla user.
username (PK)	varchar (180)	Nombre de usuario.
roles	longtext	Tipo de usuario (admin o user)
password	varchar (255)	Contraseña de usuario.
email	varchar (100)	Correo electronico del usuario.
nombre	varchar (100)	Nombre de usuario.
verificado	tinyint(1)	Determina si esta verificado en la web.
foto	longblob	Foto de perfil.
descripcion	varchar (255)	Descripcion sobre el usuario.
ubicación	varchar (255)	Ubicación del usuario.

Ilustración 3. Tabla Usuario

POST		
Atributo	Tipo de parametro	Descripcion
id (PK)	int(11)	Id de la tabla post.
contenido	varchar(255)	Contenido del post.
fecha	datetime	Fecha en la que es publicado el post.
foto	longblob	Foto incluida en el post.
respuesta	tinyint(1)	Determina si el post es una repuesta o un post normal.
id_respuesta (FK)	int(11)	Id del post original.
id_juego_id (FK)	int(11)	Id del juego sobre el post.
id_user_id (FK)	int(11)	Id del usuario que ha publicado el post.

Ilustración 4. Tabla Post

JUEGO		
Atributo	Tipo de parametro	Descripcion
id(PK)	int(11)	Id de la tabla juego.
nombre	varchar(100)	Nombre del juego
foto	longblob	Foto del juego
descripcion	varchar(255)	Descripcion del juego
categoria	varchar(100)	Categoria del juego.

Ilustración 5. Tabla Juego

JUEGA		
Atributo	Tipo de parametro	Descripcion
id(PK)	int(11)	Id de la tabla juega.
username_user_id(FK)	int(11)	Id del usuario.
id_juego_id (FK)	int(11)	Id del juego.

Ilustración 6. Tabla Juega

GUSTA		
Atributo	Tipo de parametro	Descripcion
id(PK)	int(11)	Id de la tabla gusta.
id_user_id (FK)	int(11)	Id del usuario que le ha dado me gusta al post.
id_post_id (FK)	int(11)	Id del post al que se le ha dado me gusta.

Ilustración 7. Tabla Gusta

REPORTA		
Atributo	Tipo de parametro	Descripcion
id(PK)	int(11)	Id de la tabla reporta.
id_user_id (FK)	int(11)	Id del usuario que ha realizado un reporte.
id_post_id (FK)	int(11)	Id del post que ha sido reportado.
tipo_report	int(11)	Tipo de reporte.

Ilustración 8. Tabla Reporta

Como se puede observar, el diagrama consta de 3 tablas principales: Usuario donde van a estar almacenados los usuarios que se registren y los administradores, Post donde se va a guardar tanto los posts creados como la respuesta a estos, y por último, Juego, donde se va a preservar los juegos de la red social.

Hay varias relaciones entre tablas, entre ellas están: Juega, creado para almacenar los juegos que le interesan a cada usuario; Crea, que es la relación que sirve para conectar los posts con el usuario que los ha creado; Reporta, que sirve para que los usuarios puedan reportar posts inapropiados; Gusta, que se utiliza para almacena los likes realizados a los posts ; Finalmente tenemos a Es de que sirve para relacionar de que juego estas hablando a la hora de realizar el post.

Algunas relaciones al ser de tipo N-N, es decir, de muchas a muchas, hace que se tengan que crear tablas nuevas en la base de datos. Estas serían Juega, Reporta y Gusta.

3.2- Capa lógica

Los componentes de software de las aplicaciones se pueden visualizar como residentes en un número de capas lógicas. Estas capas representan la independencia física y lógica de los componentes de software en función de la naturaleza de los servicios que proporcionan. A continuación, se van a explicar las capas que componen:

Registro

```
/**
 * @Route("/register", name="app_register")
 */
public function registro(UserPasswordHasherInterface $passwordHasher, Request $request, EntityManagerInterface $em): Response
{
    $user = new User();
    $form = $this->createForm(RegisterType::class, $user);

    $form->handleRequest($request);
    $error = "";
    if ($form->isSubmitted() && $form->isValid()) {
        $a = ['ROLE_USER'];
        $user->setRoles($a);
        $shashesPassword = $passwordHasher->hashPassword(
            $user,
            $form->get("password")->getData()
        );
        $user->setPassword($shashesPassword);
        $user->setEmail($form->get("email")->getData());
        $user->setNombre($form->get("nombre")->getData());
        $user->setVerificado(false);
        $usu = $em->getRepository(User::class)->findOneBy(array('username'=>$user->getUserIdentifier()));
        if (!$usu) {
            if ($form->get("password")->getData() == $form->get("repetir_password")->getData()) {
                try {
                    $em->persist($user);
                    $em->flush();
                } catch (\Exception $e) {
                    return new Response("Esto ha petao");
                }
                return $this->redirectToRoute('login');
            } else {
                $error = "Las contraseñas no coinciden";
            }
        } else {
            $error = "Ya existe ese nombre de usuario";
        }
    }

    return $this->render('register/register.html.twig', [
        'controller_name' => 'FormularioController',
        'form' => $form->createView(),
        'error' => $error
    ]);
}
```

Ilustración 9. Código de registro

El registro consta de un formulario que rellenas con tus datos, donde al terminar haces un submit. Desde el front, antes de realizar el submit se comprueba que los datos del formulario están en un formato correcto. Una vez los datos están en un formato correcto se comprueba si el usuario existe ya en la BBDD. En caso de no existir, se guardan los datos del formulario, entre ellos la contraseña encriptada.

Login

El Login es un formulario sencillo donde pones tu usuario y contraseña y aprietas al botón de submit. El front comprueba que están rellenos los dos inputs y, una vez comprobado esto, se comprueba que el usuario existe en la BBDD. Si existe, se comprueba que la contraseña encriptada coincide con la introducida por el formulario. Si coincide, se observa si se trata de un usuario normal o un administrador y, dependiendo de que rol tenga, entra en su debido log.

```

/**
 * @Route("/login", name="login")
 */
public function login(AuthenticationUtils $authenticationUtils, EntityManagerInterface $em): Response
{
    $error = $authenticationUtils->getLastAuthenticationError();
    if($error!=null){
        $error="Has introducido mal tus datos";
    }
    $lastUsername = $authenticationUtils->getLastUsername();
    return $this->render('login/login.html.twig', [
        'controller_name' => 'LoginController',
        'error' => $error,
        'last_username' => $lastUsername
    ]);
}

```

Ilustración 10. Código de login

Realizar Post

```

public function buildForm(FormBuilderInterface $builder, array $options): void
{
    $builder
        ->add(
            'idJuego',
            EntityType::class,
            [
                'class' => Juego::class,
                'choice_label' => 'nombre',
                'choice_name' => 'id',
                'expanded' => false,
                'multiple' => false,
                'label'=>'Juego'
            ],
        )
        ->add('contenido',TextareaType::class,array('attr' => ['class' => 'foo']))
        ->add('foto',FileType::class,array('required' => false , 'attr' => ['class' => 'foo']))
        ->add('save',SubmitType::class,array('label'=>'Insertar','attr' => ['class' => 'quitarb']))
    ;
}

```

Ilustración 11. Código de post

El usuario tiene un formulario donde pone un texto con un máximo de caracteres, puede subir una imagen y, tiene que elegir el videojuego del cual está hablando. Una vez realizado esto aprietas a subir y antes de subirse el front comprueba que todo tenga el formato correcto. Los posts se pueden subir normal o en forma de respuesta.

Reporte

El reporte tiene un formulario donde indicas que un post es inapropiado o contiene el tema de un juego del que no se habla. El reporte tiene un texto donde puedes escribir la descripción de tu reporte. Tras que el front compruebe que el formato del formulario es correcto, se sube el reporte a BBDD.

Modificar Perfil

En tu perfil habrá un formulario donde podrás insertar una imagen para tu foto de perfil, una descripción de ti mismo y una ubicación.

Comprobar reportes

Aparecen todos los reportes hechos en orden cronológico desde el más antiguo al más nuevo. Si solo ha habido un cambio en el juego se modifica el post por el indicado. En caso de que sea un post inapropiado se elimina el post y todas las respuestas a este. Además, se le aplica un strike. Al tercero se le baneará la cuenta.

Verificar cuentas

Los administradores podrán hacer que los usuarios se conviertan en cuentas verificadas. Esto significa que la aplicación confirmó que es la cuenta auténtica del personaje público, la celebridad o la marca que representa.

3.3- Capa de Presentación

La capa de presentación es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo proceso.

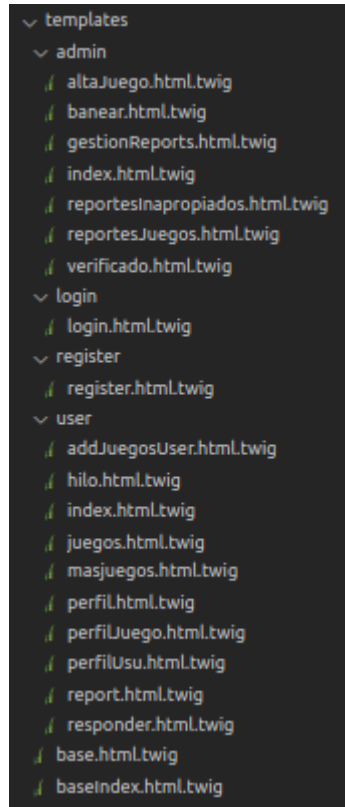


Ilustración 12. Estructura Twig

En el proyecto se utiliza el modelo twig, explicado anteriormente. En ella se utiliza lenguaje html. Para tener una mayor idea de como realizar el proyecto, se realiza la estructuración de este. La estructura realizada es la siguiente:

```
{% extends 'base.html.twig' %}

{% block title %}L4P{% endblock %}

{% block body %}
```

Ilustración 13. Extends base.twig

Se puede observar que hay dos archivos llamados base. Estos son archivos que se extienden en los demás dependiendo de cual de los dos quieres usar. En estos están los scripts para utilizar Bootstrap, para referenciar al archivo css, etc.

4- Manual de usuario

El proyecto desarrollado consta de una red social donde pueden existir dos tipos de usuarios: El usuario común y un administrador. Por lo que se van a realizar dos manuales. Uno para cada tipo de usuario.

Acceso a la aplicación

Para empezar a interactuar con la aplicación, primero, el usuario debe estar añadido a la base de datos. Para acceder a la cuenta deberá de rellenar un formulario de inicio de sesión donde va a tener que introducir su usuario y contraseña.



Inicia sesión en L4P

Usuario

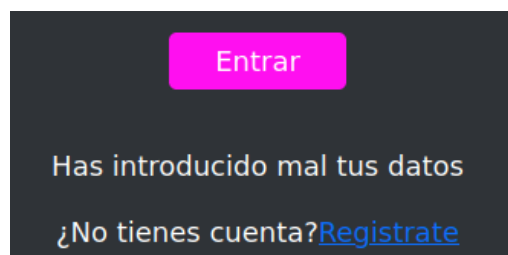
Contraseña

Entrar

¿No tienes cuenta?[Registrate](#)

Ilustración 14. Login manual

En caso de fallar con el usuario o la contraseña la aplicación te lo dirá.



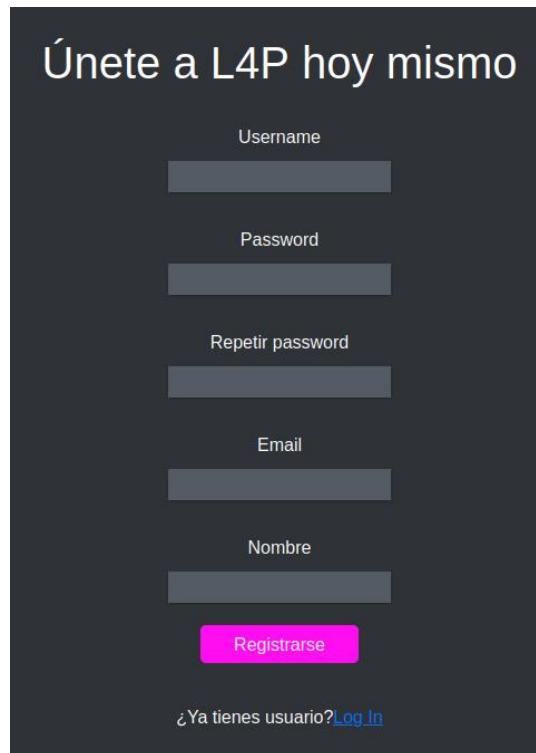
Entrar

Has introducido mal tus datos

¿No tienes cuenta?[Registrate](#)

Ilustración 15. Login error datos manual

También se puede dar el caso de que no se haya dado de alta al sistema. Entonces deberá pulsar donde pone “Registrate”. Aparecerá un formulario de registro, donde, si rellenas bien los datos, el usuario se dará de alta.



Únete a L4P hoy mismo

Username

Password

Repetir password

Email

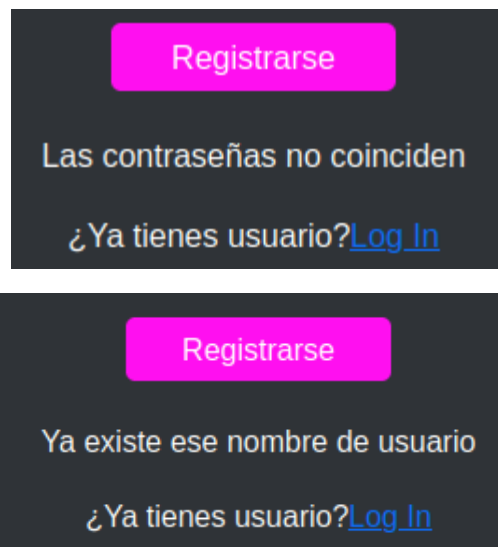
Nombre

[Registrarse](#)

¿Ya tienes usuario? [Log In](#)

Ilustración 16. Registro manual

En caso de fallar al repetir el password, o poner un usuario ya existente, la aplicación te lo dirá.



[Registrarse](#)

Las contraseñas no coinciden

¿Ya tienes usuario? [Log In](#)

[Registrarse](#)

Ya existe ese nombre de usuario

¿Ya tienes usuario? [Log In](#)

Ilustración 17. Error Registro manual.

4.1- Manual de administrador

Hemos accedido correctamente mediante el login como un usuario de administración, donde al entrar se podrá ver el índice del administrador.

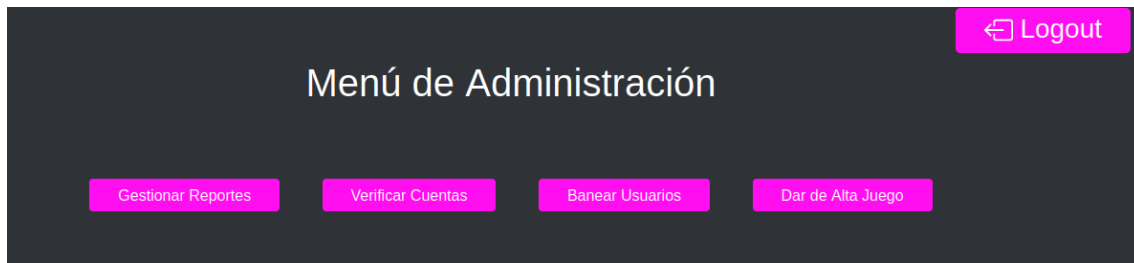


Ilustración 18. Menú administrador Manual

En este menú hay varios botones. En la esquina superior derecha está el botón para deslogearse. Este hace que cierres sesión en la aplicación y te manda al formulario de login.

Gestionar Reportes

Al apretar te aparece otro menú ya que hay dos tipos de reportes: los reportes por contenido inapropiado y los reportes donde hay posts con juegos diferentes del que se habla.

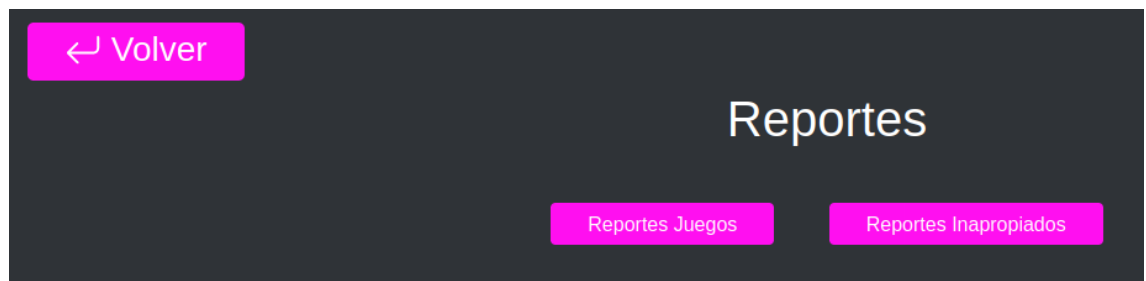


Ilustración 19. Menú Reportes Manual

Se puede observar que en la esquina superior izquierda ha aparecido un nuevo botón donde pone “Volver”, como su nombre indica, este te hará volver a la página anterior.

Reportes Juegos

Al apretar a “Reportes Juegos” aparecerá la lista de los posts reportados de este tipo. En la parte inferior de cada post aparecerá un desplegable donde podrás señalar a que juego pertenece el post. En caso de que se tenga que cambiar el juego se apretará

al botón de “Cambiar juego” y en caso de que el juego sea correcto se tiene que apretar al boto de “Pasar”. Tras apretar a alguno de estos dos botones, el post desaparecerá.

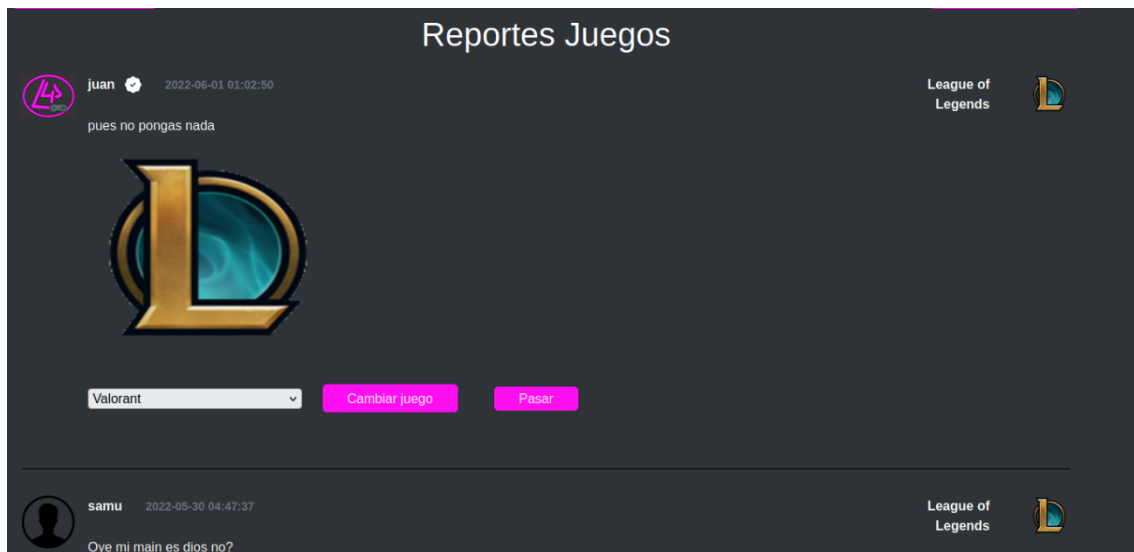


Ilustración 20. Reportes Juegos Manual

Reportes Inapropiados

Al apretar a “Reportes Inapropiados” aparecerá la lista de los posts reportados de este tipo al igual que en el otro tipo de reporte. La única diferencia es que no aparecerá ningún desplegable, solo aparecerá la opción de “Eliminar Post” donde, para no romper la cadena de respuestas se eliminará su contenido y si tiene una imagen su imagen. Y la opción de pasar.



Ilustración 21. Reportes Inapropiados Manual

Verificar cuentas

Al apretar el botón “Verificar cuentas” aparecerá un formulario donde tendrás que poner el nombre de usuario y, si no existe, al apretar el botón “aceptar” esta se verificará.

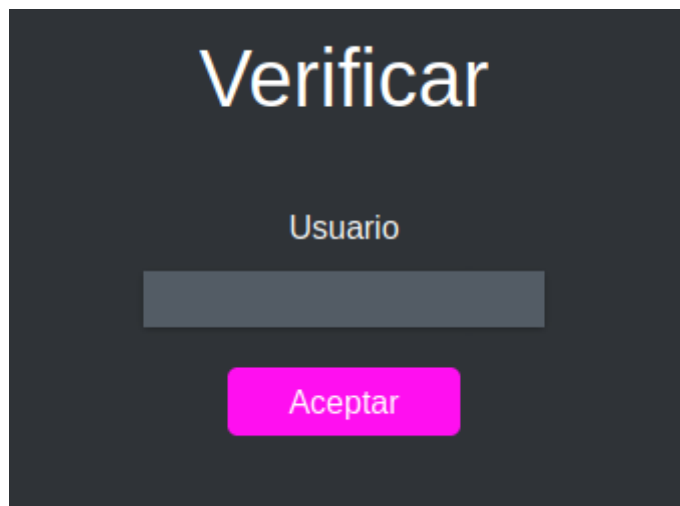


Ilustración 22. Verificar manual

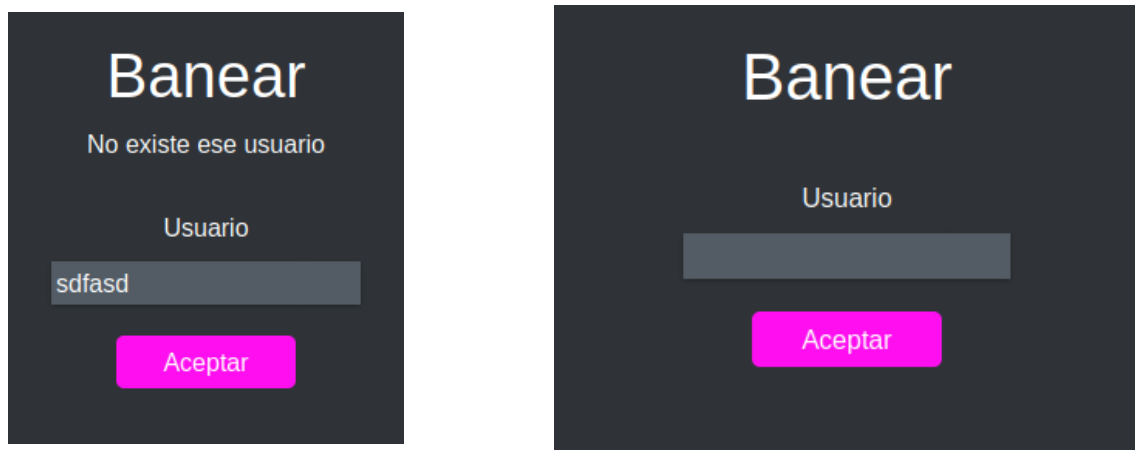
En caso de que no exista el usuario, la aplicación te lo dirá



Ilustración 23. Error Verificar Manual

Banear Usuarios

Al apretar el botón “Banear Usuarios”, al igual que en verificar aparecerá un formulario donde se tendrá que poner el nombre del usuario y, en caso de que no exista, la aplicación te lo dirá.



The image displays two side-by-side screenshots of a web form titled "Banear".

The left screenshot shows the form after a submission. It features the title "Banear" at the top, followed by the error message "No existe ese usuario" in red. Below this is a label "Usuario" and a text input field containing the text "sdfasd". At the bottom is a red button labeled "Aceptar".

The right screenshot shows the form in its initial state. It features the title "Banear" at the top, followed by a label "Usuario" and an empty text input field. At the bottom is a red button labeled "Aceptar".

Ilustración 24. Banear Manual

Dar de alta juego

Al apretar el botón “Dar de alta juego”, aparecerá un formulario donde, rellenando correctamente los parámetros, el juego se creará.

Alta Juego

Nombre

Descripcion

Categoria

Foto

Examinar... No se ha seleccionado ningún archivo.

Insertar

Ilustración 25. Alta Juego Manual

En caso de que el juego ya exista, la aplicación te lo indicará.

Alta Juego

Ya existe ese juego

Nombre

Valorant

Ilustración 26. Error Alta Juego Manual

4.2- Manual de Usuario

Hemos accedido correctamente mediante el login como un usuario normal. Al entrar puede ocurrir dos cosas. Si eres un usuario nuevo, la aplicación te obliga a elegir los juegos que te interesan.

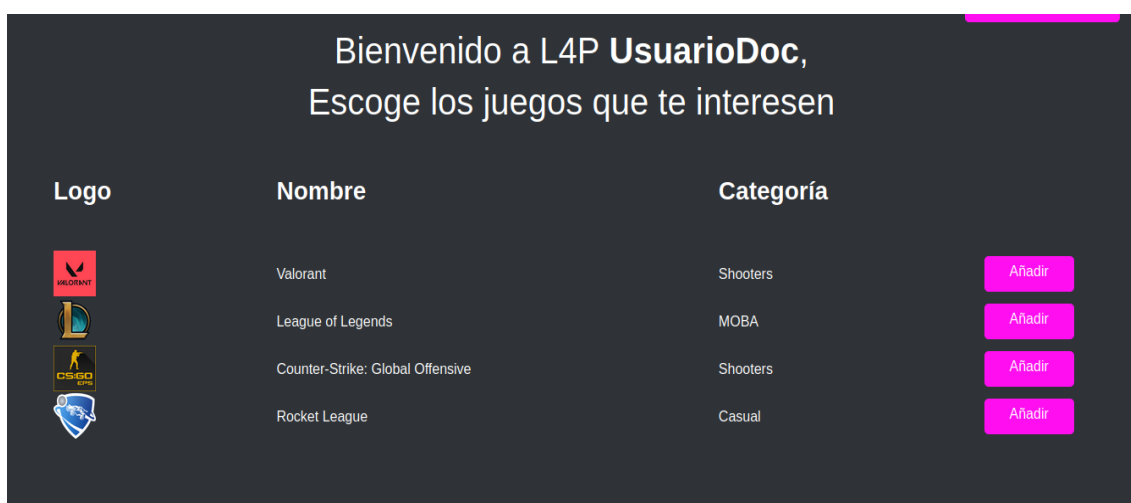


Ilustración 27. Escoger Juegos Manual

Al apretar a cualquier "Añadir" aparecerá un nuevo botón llamado "Empezar", que al apretar se te iniciará la aplicación. Además, el botón "Añadir" se deshabilitará.

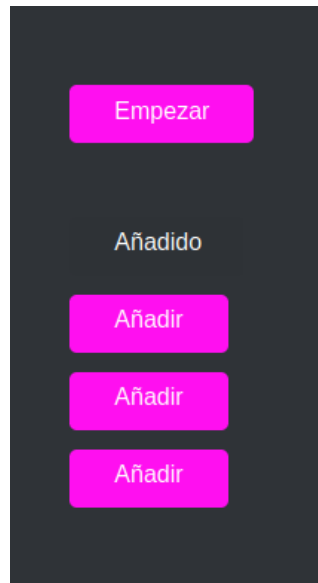


Ilustración 28. Añadir Juegos Manual

Menú

Para presentar todas las funcionalidades del usuario, vamos a ir escalándolo desde el menú.



Ilustración 29. Menú Manual

El botón de inicio te lleva al índice de la aplicación donde se muestra una caja para postear y una lista de posts ordenados por antigüedad.

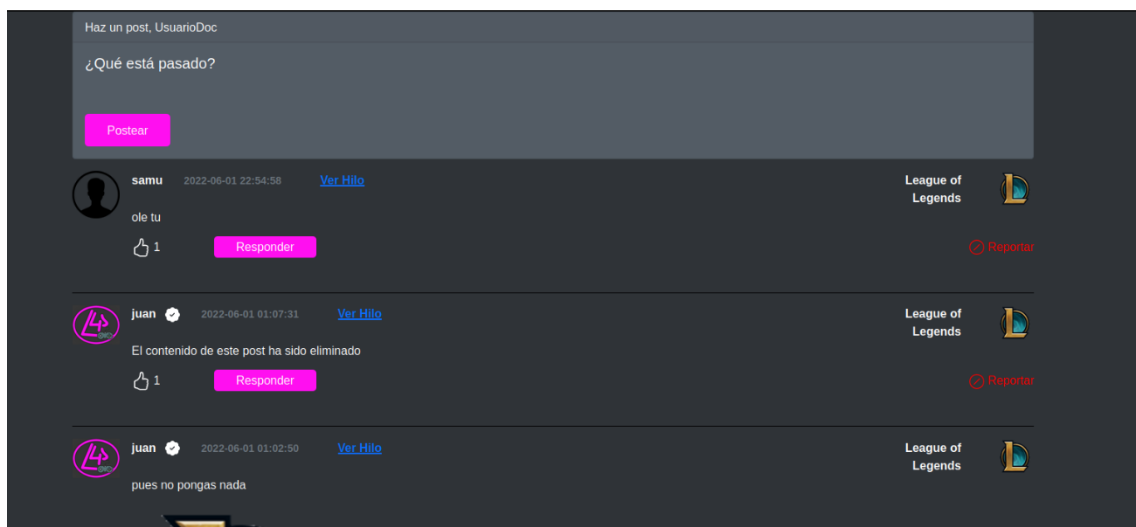


Ilustración 30. Índice Manual

Al apretar el botón “Postear” se despliega un popup con un formulario para añadir un post.

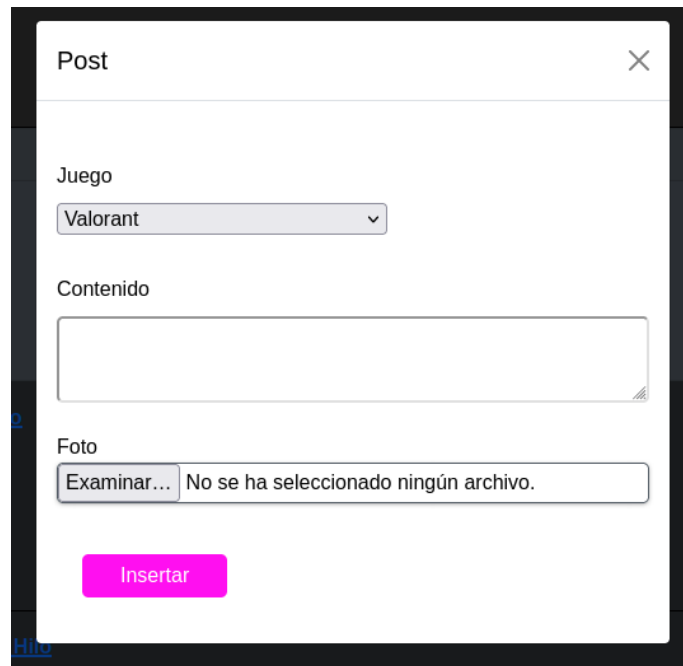
A screenshot of a 'Post' modal window. At the top, it says 'Post' with a close button (X). Below is a 'Juego' (Game) section with a dropdown menu showing 'Valorant'. Underneath is a 'Contenido' (Content) section with a large text input field. Below that is a 'Foto' (Photo) section with a button labeled 'Examinar...' and a message 'No se ha seleccionado ningún archivo.' (No file has been selected). At the bottom is a pink button labeled 'Insertar'.

Ilustración 31. Contenido Post Manual

En el formulario se define el juego del que trata tu post, el contenido y, si quieres una foto. Al apretar “Insertar” si se ha rellenado el campo “Contenido”, se creará el post.

Post

Cada post tiene una serie de funcionalidades.

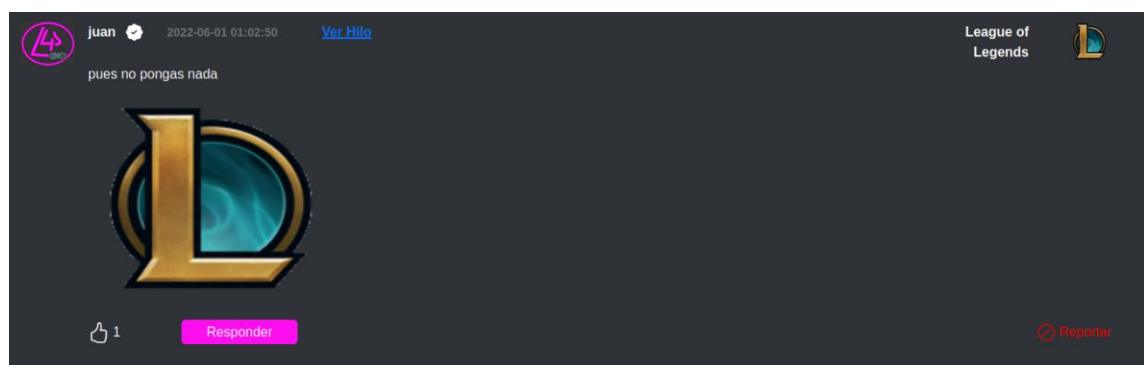


Ilustración 32. Ejemplo post Manual

En la esquina superior izquierda aparece la foto de perfil del usuario que ha creado el post. Justo a su derecha aparece su nombre de usuario que al apretar podremos ver su perfil. Más adelante cuando expliquemos los perfiles se entrará más

en profundidad. Podemos visualizar que a la derecha del nombre está el icono de verificado, lo que significa que este usuario ha sido verificado por un administrador. Si continuamos hacia el mismo lado observamos la fecha de publicación del post. Y finalmente, si el post ha sido una respuesta, aparecerá un link al hilo con el post original.

En la esquina superior derecha aparece el juego del que trata el post y el icono del juego.

En el centro del post aparece el contenido de este y, si en el post se ha subido una imagen, aparece esta. Si clicas a la imagen se te despliega un popup donde se puede ver mejor.

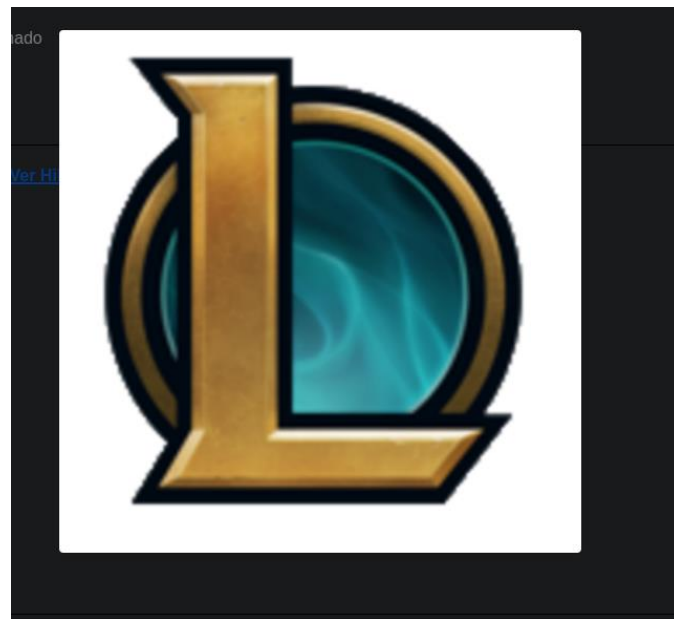


Ilustración 33. Popup imagen manual

En la esquina inferior izquierda aparece el icono de me gusta, que en caso de que el usuario le haya dado me gusta se vera relleno, y la cantidad de me gustas que tiene el post. Luego, hay un botón llamado “Responder” que, como su nombre indica te lleva a un formulario de respuesta.

En la esquina inferior derecha está el botón de reportes, donde se te abrirá el menú de reportes.

Hilo

Al apretar a “Ver Hilo” se verá lo siguiente:

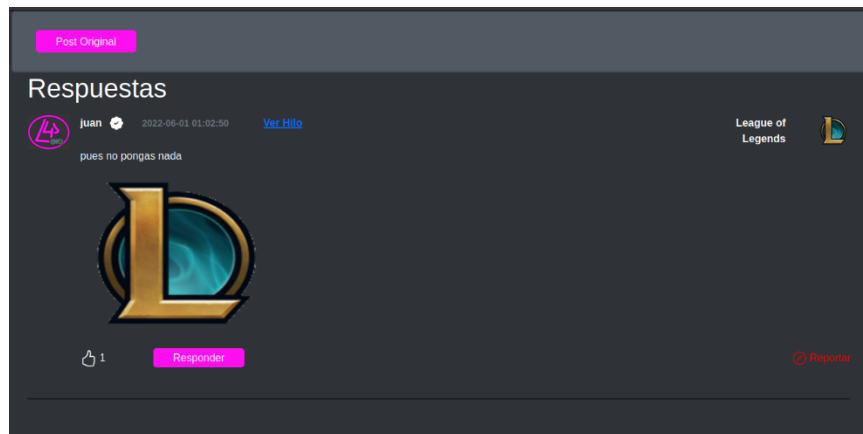


Ilustración 34. Hilo Manual

Al apretar el botón “Post Original” este se despliega enseñándote el contenido del post original. Además, aparece por orden cronológico todas las respuestas al post original.

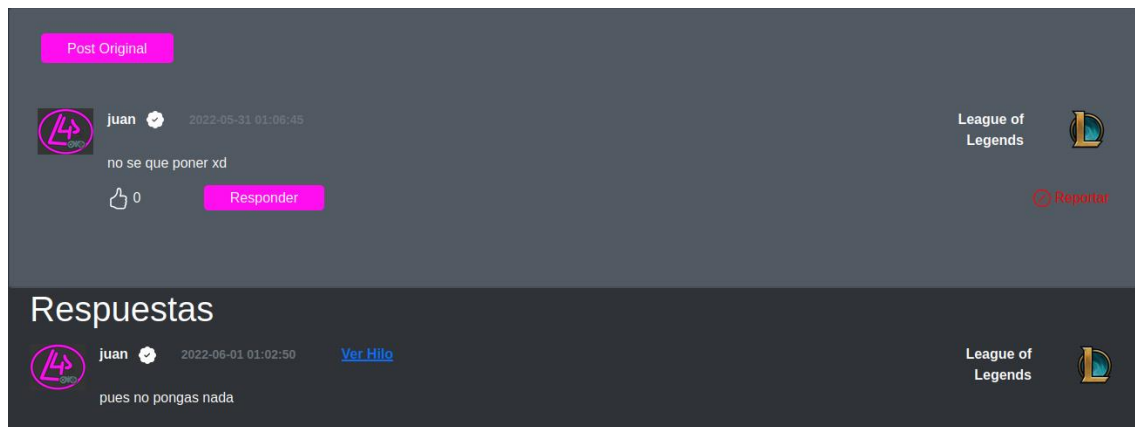


Ilustración 35. Desplegable Hilo Manual

Responder

Al apretar “Responder” aparece lo siguiente

Ilustración 36. Responder manual

Es un formulario donde pones el contenido de la respuesta y, si quieres, una imagen. Al apretar “Post a contestar” se despliega como en los hilos, el post original.

Reportar

Ilustración 37. Desplegable respuesta manual

Al clicar al botón “Reportar” aparece lo siguiente:

Ilustración 38. Menú reportar manual

Como anteriormente, al apretar en el botón “Post a reportar”, el post original se despliega.

Se tiene que apretar al tipo de reporte que se quiera.

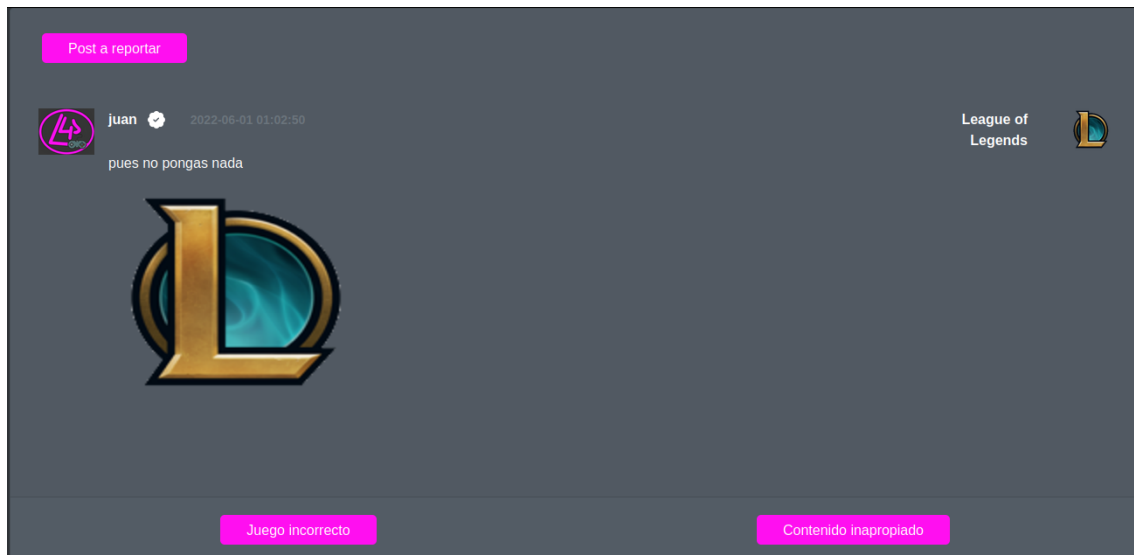


Ilustración 39. Desplegable reporte manual

Juegos

Siguiendo por el menú, clicando en “Juegos” aparece esto:



Ilustración 40. Añadir Juegos usuario manual

Aparecen dos listas. Una con los juegos que tienes seleccionados. Y otra con los juegos que no se han seleccionados.

Al apretar a cualquiera de los botones “Ver Perfil” se iría a los perfiles de los juegos que serán explicados más adelante.

Perfiles

En el menú, al apretar a “Perfil”, se te abre un desplegable. Al apretar a “Ver Perfil” se abre el perfil del usuario.

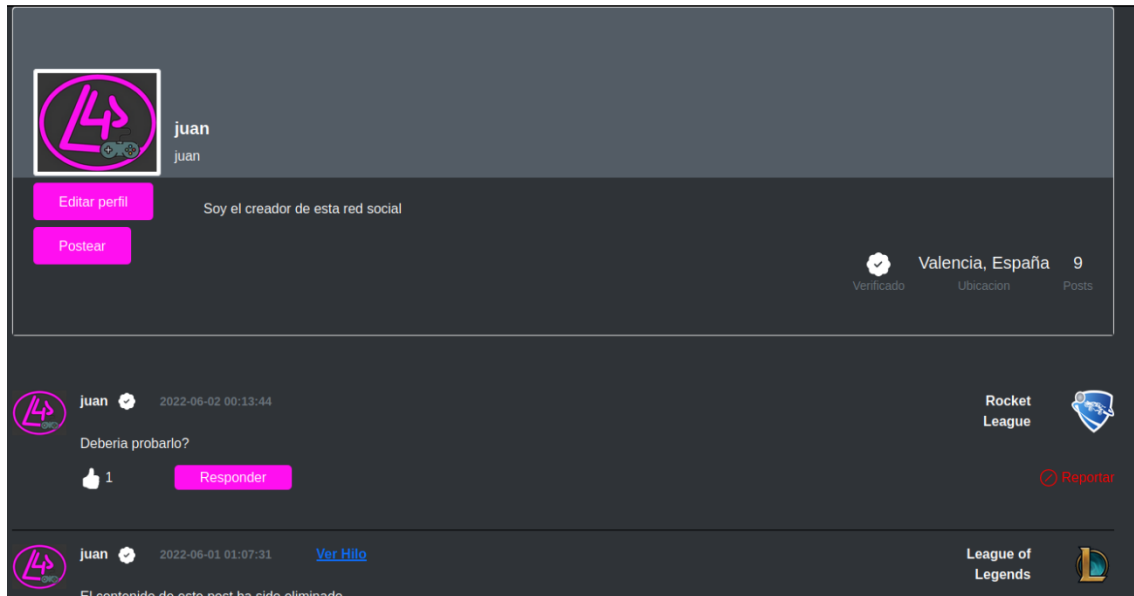


Ilustración 41. Perfil usuario

Al apretar “Editar perfil” puedes añadir una descripción, una foto de perfil y una ubicación con un desplegable que aparece al apretar. Además de actualizarlos si quieres cambiar.

A screenshot of a modal form titled 'Post' with a close button (X) in the top right corner. The form contains several input fields: 'Nombre' with the value 'juan', 'Foto' with a button 'Examinar...' and the text 'No se ha seleccionado ningún archivo.', 'Descripcion' with the value 'Soy el creador de esta red social', and 'Ubicacion' with the value 'Valencia, España'. At the bottom of the form is a pink button labeled 'Modificar'.

Ilustración 42. Post perfil manual

Al apretar “postear” aparece un popup con el contenido del post explicado anteriormente.

Abajo aparece la lista de los posts creados por el usuario ordenados cronológicamente.

Perfil de juegos y usuarios

Son como los perfiles normales, pero en estos no tienes los dos botones de editar perfil y de crear posts. En el de los usuarios se ven los posts creados por esos usuarios:

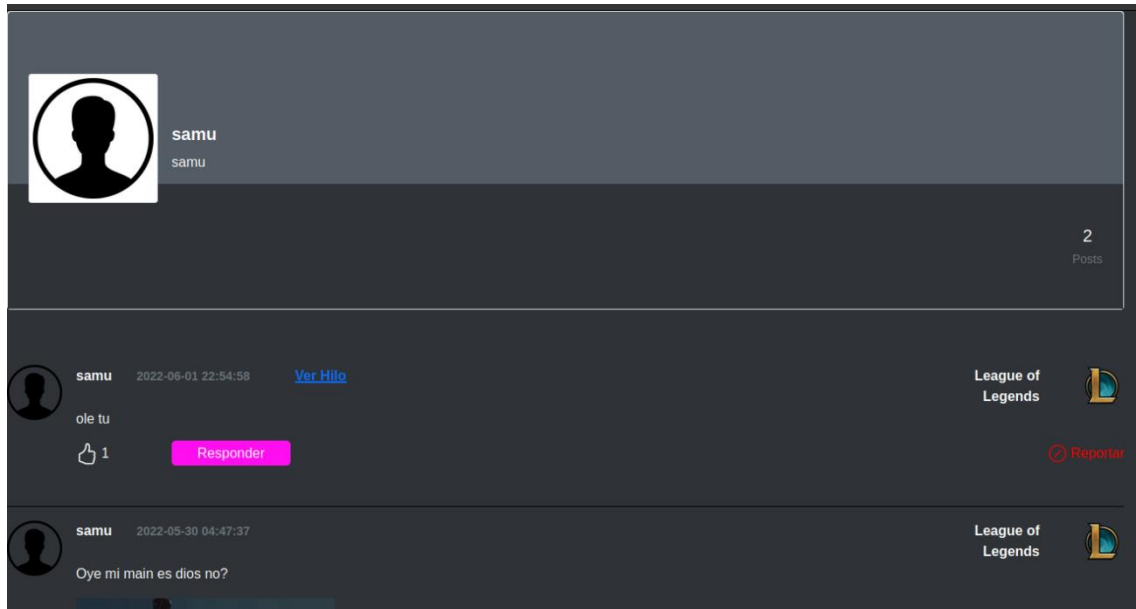


Ilustración 43. Perfil otro user manual

El de los juegos aparecen los posts creados sobre esos juegos:



Ilustración 44. Perfil juego manual

Añadir juegos

Aquí te aparece el mismo menú que cuando eres nuevo, pero con los juegos que no tienes seleccionados. Tiene la misma funcionalidad

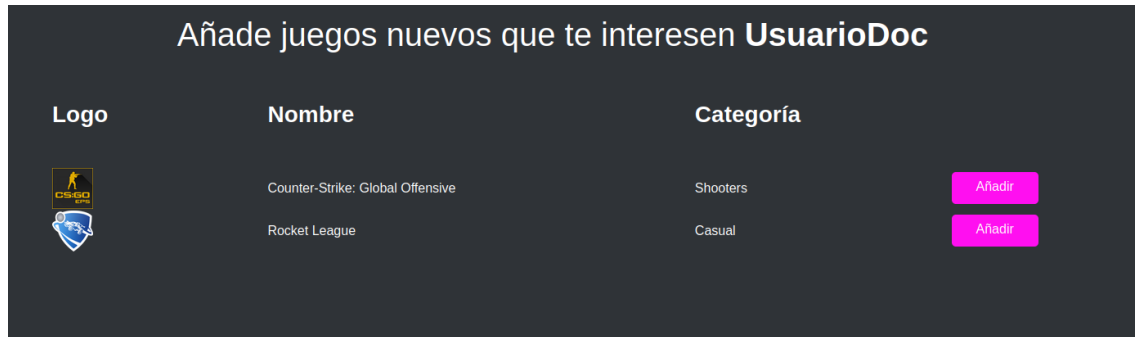


Ilustración 45. Nuevos Juegos manual

5- Conclusión

5.1- Problemas encontrados

Durante la realización del proyecto se han encontrado una serie de problemas que han dificultado su propio desarrollo. Estos son:

- Al principio se pensó en desarrollarlo sobre el lenguaje C#, pero el desplegarlo sobre Windows resultaba muy costoso, finalmente se decidió desarrollarlo en Symfony, ya que este se despliega en Linux y con mucha más facilidad.
- Este proyecto tiene una gran cantidad de formularios de todo tipo. Para ello se ha utilizado lo denominado en Symfony como FormTypes. Estos han sido un problema a la hora de tener que poner en los formularios datos más complejos. Entre ellos el FileType con imágenes, o el choice(option) con los selects sobre datos de una Entidad. También ha sido muy complejo que las imágenes se guardasen correctamente en base de datos ya que es muy diferente a la manera que se utiliza comúnmente en PHP.

- A la hora de los me gustas, para reflejarlos en twig ha sido un problema, ya que se ha tenido que guardar una variable creada directamente dentro de twig para reflejar si el usuario le había dado me gusta al post ya.
- Tras muchos intentos ha sido imposible poder hacer el sistema de chats a tiempo para la presentación del proyecto ya que crear un chat entre dos usuarios era posible pero, el tener varios activos a la vez no se ha podido conseguir.

5.2- Mejoras

Lo mejor de este proyecto es la gran cantidad de opciones extra e implementaciones que se pueden aplicar. Durante el desarrollo se barajaron la implementación de muchas mejoras para una futura versión. Entre ellas están:

- El sistema de ordenado de post según me gustas, importancia, etc.
- Un buzón de notificaciones para el usuario donde poder ver todas las interacciones nuevas que se realizan sobre su perfil.
- Sistema interactivo para administradores donde pueda manejar nuevas funcionalidades viendo la página inicial.
- Implementación del chat para los usuarios
- Sistema de opiniones en los perfiles de los usuarios.
- Sistema de matcheo para encontrar gente con la que jugar.

6- Bibliografía

<https://symfony.com/doc/current/frontend/encore/copy-files.html>

<https://symfony.com/doc/current/doctrine.html>

<https://stackoverflow.com/questions/16255070/symfony2-doctrine-querybuilder-where-in>

<https://stackoverflow.com/questions/26361918/set-limit-to-selected-rows-from-db>

<https://www.doctrine-project.org/projects/doctrine-orm/en/2.8/reference/query-builder.html>

<https://www.doctrine-project.org/projects/doctrine-orm/en/2.8/reference/query-builder.html>

<https://symfony.com/doc/current/reference/forms/types/form.html#attr>

<https://symfony.com/doc/current/reference/forms/types/choice.html#choice-name>

<https://stackoverflow.com/questions/11868497/force-a-field-to-not-be-required>

<https://getbootstrap.com/docs/5.0/components/modal/>

<https://getbootstrap.com/docs/5.0/components/navbar/>

<https://symfony.com/doc/current/reference/forms/types/text.html>

<https://stackoverflow.com/questions/7913086/how-to-set-default-value-for-form-field-in-symfony2>

<https://stackoverflow.com/questions/40983353/the-forms-view-data-is-expected-to-be-an-instance-of-class-but-is-an-stri>

<https://stackoverflow.com/questions/70774198/symfony-how-to-set-a-file-type-in-an-edit-form>