

# DESARROLLO DE APLICACIONES WEB CURSO 2DAW 2022 / 2023



**Florida**

Universitària

Proyecto de Desarrollo de Aplicaciones Web

## Memoria de Proyecto



**Raül Sales Herranz**

17 / 05 / 2023



## Indice

1 Introducción.....	4
2 Resumen de el proyecto.....	4
2.1 Lenguaje Español.....	4
2.2 English language.....	5
3 Justificación y objetivos.....	6
Que se busca conseguir con este proyecto?.....	6
Como y cuales son los requisitos de la aplicación?.....	6
3.1 Análisis de mercado.....	7
3.1.1 Competencia.....	8
Indice de figuras Nº1.....	8
Indice de figuras Nº2.....	9
Indice de figuras Nº3.....	9
3.1.2 Competidores.....	10
3.1.3 Valor añadido.....	11
3.2 Metodologías.....	11
Primer Sprint ( 27-03-2023 ).....	12
Segundo Sprint (20-04-2023):.....	12
Tercer Sprint (11-05-2023):.....	12
3.3 Componentes de la aplicación.....	13
Estructura de la red social Cliente:.....	13
Cabecera Red Social.....	13
Cabecera Administración red social.....	13
Cuerpo Principal de contenido Red Social.....	13
Cuerpo Principal de contenido Administración de Red Social.....	13
Pie de página.....	13
Front-end:.....	14
HTML5 Y CSS3.....	14
Angular.....	14
Back-end:.....	14
Base de Datos mysql:.....	15
Doctrine:.....	15
Twig.....	15
Symfony:.....	15
Control de versiones:.....	15
Funcionalidades lado cliente.....	16
Funcionalidades lado administración.....	16
Evolución de el proyecto.....	17
Metodologías empleadas:.....	17
Scrum:.....	18
Versiones de mockup.....	18
Base de datos.....	19
Guía de estilos.....	20
Problemas/dificultados encontrados durante el desarrollo:.....	22
Análisis de el resultado:.....	23
Conclusiones:.....	23
Partes Positivas:.....	23
Partes Negativas:.....	23
Desarrollo largo plazo:.....	24

Bibliografía o webgrafía:.....	24
Glosario de términos:.....	25
Anexo 1:.....	26
Anexo 2:.....	27

# 1 Introducción

El proyecto de Kurigram (mensajería), trata de una red social que aprovecha el auge de este tipo de aplicaciones, explotando una idea que puede implementarse en nuestro día a día.

Para ello se aprovechan ciertos factores como la experiencia de usuario en aplicaciones de una misma línea de actividad. Este proyecto sigue una continuidad con la competencia, pero cuenta con un factor diferencial, debido a la idea de administración de Kurigram, y como línea de futuro, la conexión con una plataforma de streaming llamada Streamcube, y la retrocompatibilidad de cuentas, experiencias de usuario parecidas etc.

## 2 Resumen de el proyecto

### 2.1 Lenguaje Español

El proyecto consistirá en una red social web, el cual será accesible vía navegador, y un panel de administración, el cual se dividirá en dos proyectos, Streamcube, y Kurigram.

La aplicación de cara a el administrador, o equipo de administración, contará con la capacidad de gestionar los usuarios, información sobre ellos, como sus publicaciones, sus seguidores, los usuarios que sigue, cambiar datos de estos, y eliminar-los en caso de ser necesario.

Podrá encontrar, un apartado de publicaciones, el cual se listará información sobre las publicaciones en general, y la posibilidad de crear uno de ser necesario.

Además de encontrar un apartado de gestión de eventos.

La aplicación de cara a el cliente, permitirá hacer registro de usuarios, y acceso a sus perfiles.

Para acceder a ellos, será necesario conocer el email y contraseña de el usuario en cuestión, o crear uno, realizando el cumplimiento de la nueva ley LOPDGDD (Ley Orgánica de Protección de Datos y Garantía de Derechos Digitales) . En la cabecera de esta, se podrá , a parte de iniciar sesión o registro de usuarios, encontrar un listado de los usuarios, a los cuales podrá seguir.

Habiendo accedido a la página como usuario, podrá encontrar en cuestión los eventos con su fecha titulo e imagen, también se podrá encontrar las publicaciones, con la posibilidad de marcar con me gusta la publicación. En el sector izquierdo, encontramos una zona perfil.

## **2.2 Abstract**

The project will consist of a web-based social network, which will be accessible via browser, and an administration panel, which will be divided into two projects, Streamcube, and Kurigram.

The application for the administrator, or administration team, will have the ability to manage users, information about them, such as their publications, their followers, the users they follow, change their data, and delete them if necessary.

You will be able to find a section of publications, which will list information about publications in general, and the possibility of creating one if necessary.

You will also find an event management section.

The application for the customer, will allow user registration and access to their profiles.

To access them, it will be necessary to know the email and password of the user in question, or create one, in compliance with the new law LOPDGDD (Organic Law on Data Protection and Guarantee of Digital Rights). In the header of this, you can, apart from login or user registration, find a list of users, which you can follow.

Having accessed the page as a user, you can find in question the events with their date, title and image, you can also find the publications, with the possibility of marking the publication with "I like it". In the left sector, there is a profile area.

### 3 Justificación y objetivos

Kurigram es una red social, intuitiva, fácil de utilizar, que trata correctamente los datos introducidos por el usuario, y que cuenta con un sistema de encriptado de contraseñas. Además de una proyección bastante grande, debido a sus utilidades, impresiones, y abanico de posibilidades de cara a su puesta en uso.

#### **Que se busca conseguir con este proyecto?**

Una red social fácil de usar, completa, intuitiva y accesible para el público. Y a su vez, un panel de administración de esta aplicación, para los trabajadores/administradores de esta. De esta manera, y aprovechando las oportunidades, que nos ofrece el día a día, podemos a una mayor cantidad de personas, de todas las edades, pero principalmente perfiles jóvenes.

Todo esto formando un proyecto o aplicación muy completa, que aplique conocimientos adquiridos, y que dictamine, la capacidad total de montar una aplicación increíble, y con mucho potencial.

#### **Como y cuales son los requisitos de la aplicación?**

Durante varias reuniones, se matizaron cuales eran los principales objetivos, funcionalidades que nos gustaría tener, como y con que tecnologías solucionaríamos este problema o necesidad, y otra serie de puntos relacionados con el tipo de cliente final.

Tenemos un requisitos mínimos recogidos, los cuales piden concretamente poder gestionar usuarios , su información, publicaciones, actividades, y un sistema de chat entre usuarios (como parte optimista) .De esta forma garantizamos el cumplimiento de las necesidades planteadas, y con una estructura de trabajo.

##### **- Requisitos mínimos:**

La aplicación tendría que poder iniciar sesión como usuario, registrar usuarios, listar cuentas para futura funcionalidad de seguir/dejar de seguir. Las contraseñas deben ser encriptadas como motivo de seguridad, y los datos deben de no ser transferidos de forma comercial. Se deben poder listar publicaciones y eventos.

En cuanto a administración, se debe poder solamente iniciar sesión con cuenta existente, gestionar los usuarios, y sus datos como publicaciones que hagan. Los eventos se deben poder gestionar, y también se debe poder crear publicaciones, a poder ser paginadas.

#### **- Requisitos deseables:**

La aplicación tendría que tener un sistema de usuarios de tipo iniciar sesión y registrarse. Pero, no se le requerirá estar registrado o con sesión abierta, para poder ver listados de usuarios, publicaciones, o eventos. Todo esto utilizando e implementando un guard de control. Podría seguir otros usuarios mediante un botón para seguir o dejar de seguir en el enlace a gente. Cuando este usuario esté registrado, o con sesión abierta, podrá dar me gusta o e incluso, crear nuevas publicaciones. Además de ello, debería poder filtrar por roles de usuario, en caso optimista.

#### **- Requisitos sobre funcionalidades a futuro**

La aplicación incorporará la interfaz de mensajería de usuarios, con una interfaz, tanto en cliente como administración. Deberá haber seguridad y restricciones en la parte de administración, además de poder acceder a una carpeta compartida de imágenes publicadas. Además se implementarán funcionalidades de editar, borrar o actualizar datos del usuario actual desde la red social, o la administración. Y se podrá cerrar sesión de forma correcta

### **3.1 Análisis de mercado.**

En el mercado, podemos encontrar varios tipos de aplicaciones, tanto multiplataforma, como web, que se desarrollan en un mismo ámbito que Kurigram.

Para ello necesitamos conocer que ofrece la competencia, que podemos virtudes y defectos podemos destacar sobre estas, como podemos atraer a sus clientes, en

resumen una especie de DAFO sobre estas (Debilidades, Amenazas, Fortalezas y Oportunidades).

Según el mercado podemos encontrar

### 3.1.1 Competencia.

Si analizamos la competencia, encontramos que su sostenibilidad es mediante anuncios, tienen una interfaz minimalista, tipografías legibles, sensación de apertura y limpieza, se aprecia Gestalt, y realmente no más de una gama de color. Teniendo en cuenta que en una paleta de colores el blanco y negro son colores neutros.

Siguen un patrón de funcionamiento, y encontramos demasiadas similitudes, esto favoreciendo la experiencia de usuario.

### Índice de figuras Nº1

Este gráfico, trata básicamente como indicador de el auge total de algunos competidores. Cabe recalcar que la tendencia de aplicaciones como Facebook o Whatsapp está en decadencia.

Actualmente las redes con más usuarios serían Tik Tok, o Instagram.

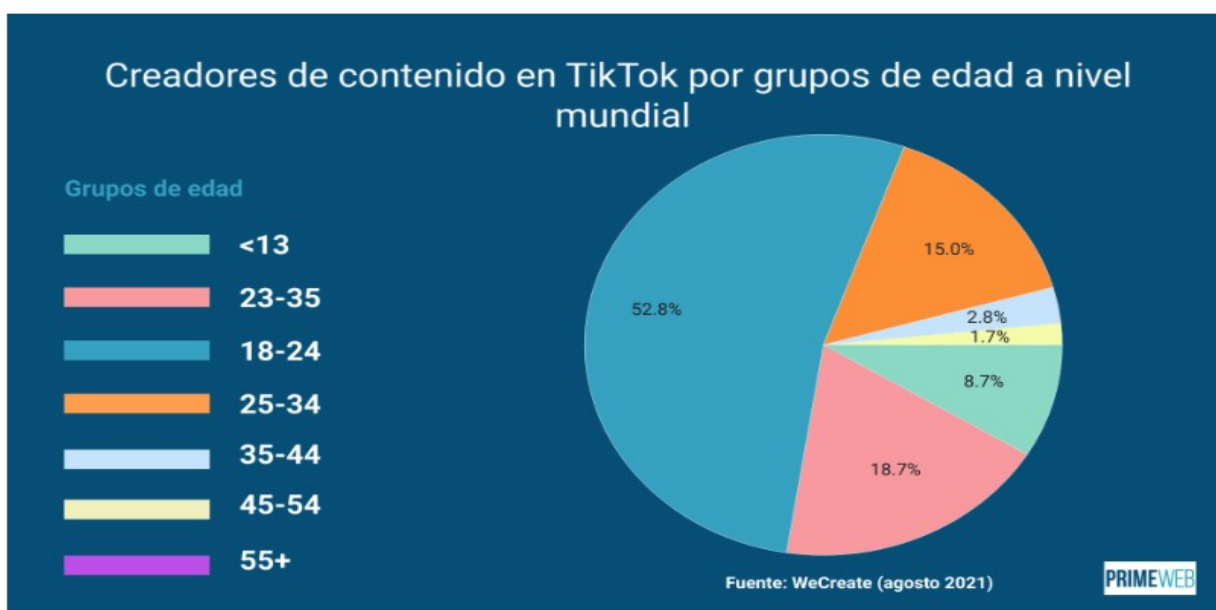




## Indice de figuras N°2

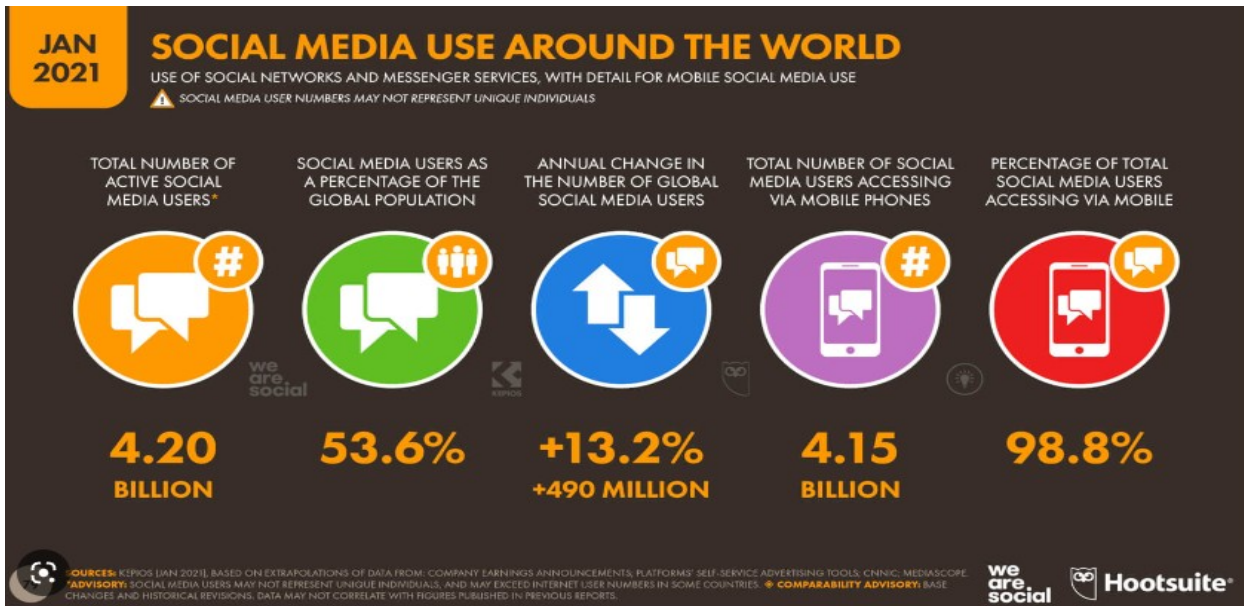
Sobretudo, a pesar de el peso de otros competidores, la población joven tiene como opción preferida Tik Tok, debido a que trata de videos cortos, liberales y que generan mucho dinero, patrocinadores y influencia.

Aquí podemos ver una pequeña clasificación de grupos de edades de Creadores de contenido en Tik Tok.



## Indice de figuras N°3

Datos sobre el uso de redes sociales, que comparten entre ellos



### 3.1.2 Competidores

Encontramos varios competidores, los tres principales de la compañía son:

**Twitter:** La red social por excelencia, cuando hablamos de popularidad, en la cual encontramos una aplicación gratuita en la cual podemos encontrar, unos algoritmos de búsqueda mediante etiquetas, usuarios o temas en concreto. En esta red social se pueden publicar imágenes, mensajes, videos, seguir usuario

**Instagram:** Una red social parecida a Twitter, pero con la peculiaridad, de que los usuarios pueden publicar videos o imágenes cortos, que se eliminan a las 24 horas de publicación, y quedan guardados en el perfil, con posibilidad de destacar-los. En cuanto a la mensajería entre usuarios, encontramos la posibilidad de realizar llamadas, video-llamadas, llamadas y video-llamadas grupales...

Cabe destacar que también posee ciertos algoritmos de búsqueda, el cual retiene a los usuarios, con temas que le puedan interesar.

**Reddit:** Una red social parecida a las anteriores, pero más centrada a la diversión de los usuarios. Esta destaca por hacer también actividades colectivas, como que cada usuario puede poner un pixel en un mural, y puede hacer algunas

creaciones de éxito. Esto es destacado debido a que son actividades que generan un aumento de la producción de la dopamina.

### 3.1.3 Valor añadido

El valor añadido de esta aplicación, es la recopilación de todos los contenidos, y actividades que a el usuario le provocan la generación de mayor cantidad de dopamina. Es decir, todas aquellas cosas que producen un estímulo al cliente, y le provocan la necesidad de tener que estar adherido a la actividad.

Para ello implementamos las funcionalidades de cualquier red social, pero con las pretensiones de poder estar vinculados a una plataforma de streaming, el cual están en auge, y atraen a una mayor cantidad de gente joven. También posee una interfaz minimalista, que aprovecha la experiencia de el usuario (UX) para que no requiera ningún tipo de dificultad, ni adaptación al usuario, es decir, “se siente como en casa”.

Como idea de sostenibilidad de el producto creado, y con curva de desarrollo a futuro, esta será gratuita, siendo sostenible mediante patrocinadores. En un principio no sería considerado un producto sin ánimo de lucro, aunque , de cara a futuro y buscando una mayor sostenibilidad, se podrían añadir planes para usuarios, implementando verificados mediante subscripciones(3.99€), y también la posibilidad de que pagando una subscripción de destacado de usuarios, se pueda conseguir llegar a más gente, con ayuda de un futuro algoritmo (7.99€).

## 3.2 Metodologías.

Se ha empleado la metodología ágil, en el que se ha puesto en practica un tablero Kanban. Utilizado para marcar, definir y poner en marcha tareas, con una finalidad organizativa.

Para ello, nuestro scrum master, nos ha ido delimitando ciertos sprints (fechas en las cuales se expone, defiende y corrige nuestro trabajo y orientación). En las siguientes fechas, se va a resumir que puntos, y objetivos se han realizado.

Dentro de la metodología ágil, encontramos la implementación de patrones S.O.L.I.D, los cuales han sido implementados, para garantizar que en el caso de crecimiento de el

proyecto o aplicación, e sta no genere problemas a el equipo de desarrollo, sea más mantenible y sostenible.

### **Primer Sprint ( 27-03-2023 ):**

- Maquetado y maquetado simple de la web. (requisito)
- Guía de estilos a seguir para este proyecto. (requisito)
- Fichero de documentación de análisis de requisitos. (requisito)
- Documento de estado del arte. (requisito)
- Montaje y estructura básica de proyecto de administración. (requisito)

### **Segundo Sprint (20-04-2023):**

- Relaciones entre el usuario y los eventos.
- Relaciones entre el usuario y las publicaciones.
- Creación , edición, detallado, listado y borrado de usuarios (CRUD)
- Creación, edición, detallado, listado y borrado de eventos (CRUD)
- Creación , y listado de publicaciones.

### **Tercer Sprint (11-05-2023):**

- Estructura de controladores, repositorios, y modificaciones para el sistema de mensajería.
- Mejorando ciertos aspectos de seguridad, routing y sistema de seguir y dejar de seguir usuarios.
- Mejorando el mostrado de publicaciones, más guardado de imágenes en carpeta local.
- Montando estructura del front-end
- Mostrando los usuario mediante peticiones a la api.
- Mostrando los eventos, con sus respectivos datos mediante peticiones a la api.
- Conexión front-end/back-end. (mixto)
- Registro e Inicio de sesión de usuarios. (cliente)

- Construcción de Guard Para futuro control de seguridad. (cliente)
- Interfaz de envío/recibimiento de mensajes entre usuarios.

### **3.3 Componentes de la aplicación.**

#### **Estructura de la red social Cliente:**

##### ***Cabecera Red Social***

Como toda aplicación web, contamos con una cabecera, el cual contendrá el logo de la aplicación como elemento identificativo, a su vez también podremos encontrar un apartado denominado inicio, un apartado llamado gente, y los enlaces a iniciar sesión, o registrar usuarios

##### ***Cabecera Administración red social.***

Esta contará con un fondo azul ultramarino, el cual tendrá los elementos logo, inicio, perfil y cerrar sesión. De forma estructural, y implementando diseño adaptativo.

##### ***Cuerpo Principal de contenido Red Social***

Contendrá tres secciones, las cuales se dividen en la zona usuario, la zona de listado publicaciones, y la zona de listado de eventos.

##### ***Cuerpo Principal de contenido Administración de Red Social***

Tendrá varias tarjetas. Estas para ramificar o separar funcionalidades que se puedan realizar en el panel. Cabe recalcar, que mediante el uso de herencia de plantillas, este menú variará dependiendo de la funcionalidad de la tarjeta que se ha accedido. Es decir, si accedo a usuarios, cambiará el contenido de la página por las opciones de usuarios con su respectiva ruta.

##### ***Pie de página***

En este caso, no se ha incorporado ningún elemento delimitador de fin de páginas, debido a que las redes sociales nunca llegan a un final en el cual un usuario pueda llegar a un fin de página

## **Front-end:**

Para el front-end, se ha utilizado las tecnologías html, css y la librería de javascript. Todo el lado cliente estará desarrollado y programado con el framework Angular. También cabe recalcar, que mediante unos endpoints realizados con Postman, una herramienta que hemos utilizado para la creación de un API, el cual permite conectar el lado cliente, y servidor para que trabajen en una misma base de datos.

### ***HTML5 Y CSS3***

Se ha empleado el uso de hojas de estilo en cascada y lenguaje para estructurado web HTML . Utilizando correctamente la estructura semántica de la versión html5. Esta trata de no implementar muchos div innecesarios, sino dividir por secciones como main, article, aside, footer header de forma wue quede todo bien organizado y entendible, sin necesidad de usar identificadores para cada uno de ellos.

### ***Angular***

Angular es un framework de javascript, el cual proporciona una estructura de directivas, rutas y la posibilidad de combinaciones entre componentes con código de hojas de estilos, y html. Esta comparte con symfony el que ambas trabajan con el modelo MVC.

Generalmente este framework se utiliza para desarrollar aplicaciones, completas, potentes, de calidad y un buen rendimiento, ya que es de tipo dinámico y muy complementario.

A parte, ofrece la posibilidad de uso de librerías como Angular materials, la cual utilizaremos en nuestro front-end, para todo aquello que le mostramos al usuario.

## **Back-end:**

La estructura de el Back-end la forman, una base de datos MSQl conectando tablas mediante campos relacionados, y un proyecto Symfony. En este proyecto recibe una gran importancia Doctrine, este facilitando el trabajo con base de datos y utilizando el motor de plantillas Twig para toda la parte tanto estructural, como visual de estilos y elementos web.

***Base de Datos mysql:***

MySQL es un sistema de gestión de bases de datos relacionales de código abierto, utilizado para almacenar, organizar y acceder a datos en una variedad de aplicaciones y entornos.

Utilizada para almacenar y gestionar todo tipo de datos de la aplicación, la utilizo para trabajar con datos de ejemplo, que son necesarios para probar la funcionalidad de la aplicación cliente. Tanto para actividades, como para la gestión de usuarios.

Esta utilizada de forma relacional, para acceder a campos desde otra tabla, de esta forma uniendo dos tablas por un campo.

***Doctrine:***

Doctrine es un conjunto de herramientas de persistencia de objetos para PHP que se utiliza para interactuar con una base de datos relacional. Es una biblioteca de código abierto que ofrece una capa de abstracción de bases de datos, lo que permite a los desarrolladores trabajar con bases de datos de manera más intuitiva y sencilla,

***Twig***

Twig es un motor de plantillas de código abierto, el cual nos permite la utilización y/o complemento de código con php, html xml u otros formatos. Se utiliza principalmente, con el framework de aplicaciones symfony. Este tipo de motor de plantillas, nos permite la herencia entre otras plantillas, control de flujo y filtrado, muy favorable para nuestro tipo de aplicación, y que queremos renderizar.

***Symfony:***

Symfony es un framework de aplicaciones web, que utiliza el lenguaje de programación php. Este ofrece muchas facilidades y sencillez a la hora de montar un proyecto web sólido y consistente. Además de contar con una enorme cantidad de documentación. Para el caso nos será muy útil para todo lo que gira entorno a el usuario.

***Control de versiones:***

Para el control de versiones, se ha utilizado Github, que es un sistema en el cual se pueden publicar repositorios , tanto públicos , como privados, además de poder realizar otras facetas como testing e incluso automatizar acciones mediante, Github Actions.

## Funcionalidades lado cliente

Podemos encontrar varias funcionalidades desarrolladas en el lado cliente

Listado de usuarios

Listado de eventos

Listado de posts

Registro de usuarios

Inicio de sesión con usuarios.

## Funcionalidades lado administración

En este caso podemos dividir en secciones, tal cual podemos encontrar en la aplicación:

-Inicio de sesión de Usuarios

Comprobamos si lo introducido por formulario, realmente se encuentra en base de datos. En caso de ser correcto, por permite acceder como usuario normal

-Usuarios:

Listado de usuario + datos básicos. Hecho utilizando la función nativa `findAll()` para recoger todos los usuarios de la entidad Usuarios y renderizando todo lo necesario en una ruta de elmotor de plantillas Twig.

Borrado de usuarios. . Realizado usando la función nativa

Edición de usuarios. Hecho recogiendo el valor de el usuario actual con sus detalles, y haciendo uso de la función de el repositorio , `save()` , recogiendo lo que se pasa por formulario (request) y aplicándolo mediante un `setter()`.

Inserción de un nuevo usuario. Hecho haciendo uso de la función de el repositorio , `save()` , recogiendo lo que se pasa por formulario (request) y aplicándolo mediante un `setter()`.

Publicaciones de los usuarios. Haciendo uso de la función `save()`, esta permite insertar un nuevo objeto post.



Seguidores de los usuarios. Muestra todos los usuarios que siguen a un usuario en concreto mediante el id.

Usuarios que sigue. Muestra todos los usuarios que le siguen a un usuario en concreto mediante el id.

Posibilidad de consultar datos de nuestro perfil. Leemos información de el usuario que ha iniciado sesión

#### -Publicaciones

Listado de publicaciones. Nos permite leer las publicaciones, y información de el usuario

Inserción de nuevas publicaciones. Nos permite insertar nuevas publicaciones mediante `save()`

Almacenamiento de las imágenes de publicación en carpeta. A la hora de insertar la imagen, le pedimos que se guarde en una carpeta local dentro de public llamada upload

#### -Eventos

Listado de eventos. Nos permite leer todos los eventos mediante el uso de `findAll()`

Inserción de eventos. Nos permite insertar nuevos eventos mediante la función `save()`

Edición de eventos. Nos permite editar el contenido mediante getters y setters, y contenido de formulario

Borrado de eventos. Nos permite borrar mediante el id

## **Evolución de el proyecto**

### **Metodologías empleadas:**

Se ha utilizado la Metodología Agile, debido a que permite mediante un nivel organizativo mayor, una mayor adaptabilidad, transparencia y mayor colaboración.

## Scrum:

Scrum, se considera metodología agile. Permite adoptar una estructura de trabajo flexible y adaptable. Utiliza iteraciones cortas y fijas de trabajo llamadas sprints, donde se entregan incrementos de producto o servicio de forma regular.

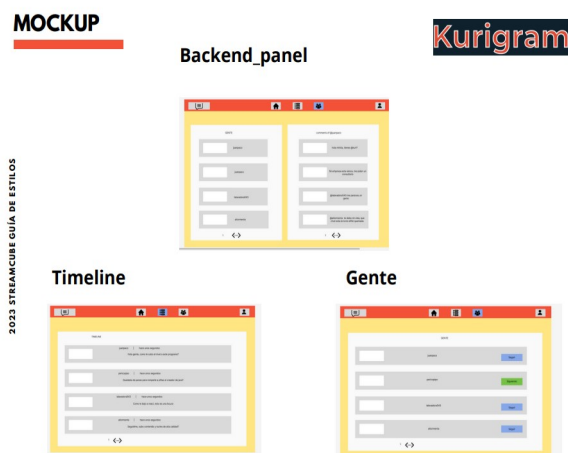
Dentro de la metodología scrum, vamos a hablar de la herramienta utilizada, llamada Trello.

Trello es una herramienta de gestión de proyectos basada en tableros visuales.

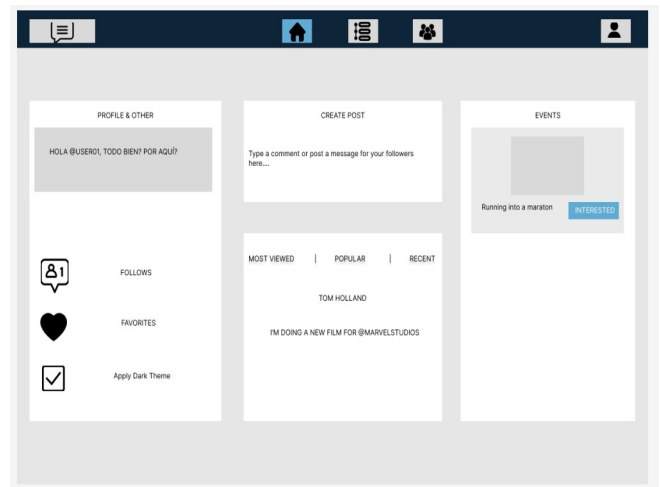
Proporciona una interfaz intuitiva y fácil de usar que permite a los equipos organizar y dar seguimiento a sus tareas y proyectos.

## Versiones de mockup

-Inicial:

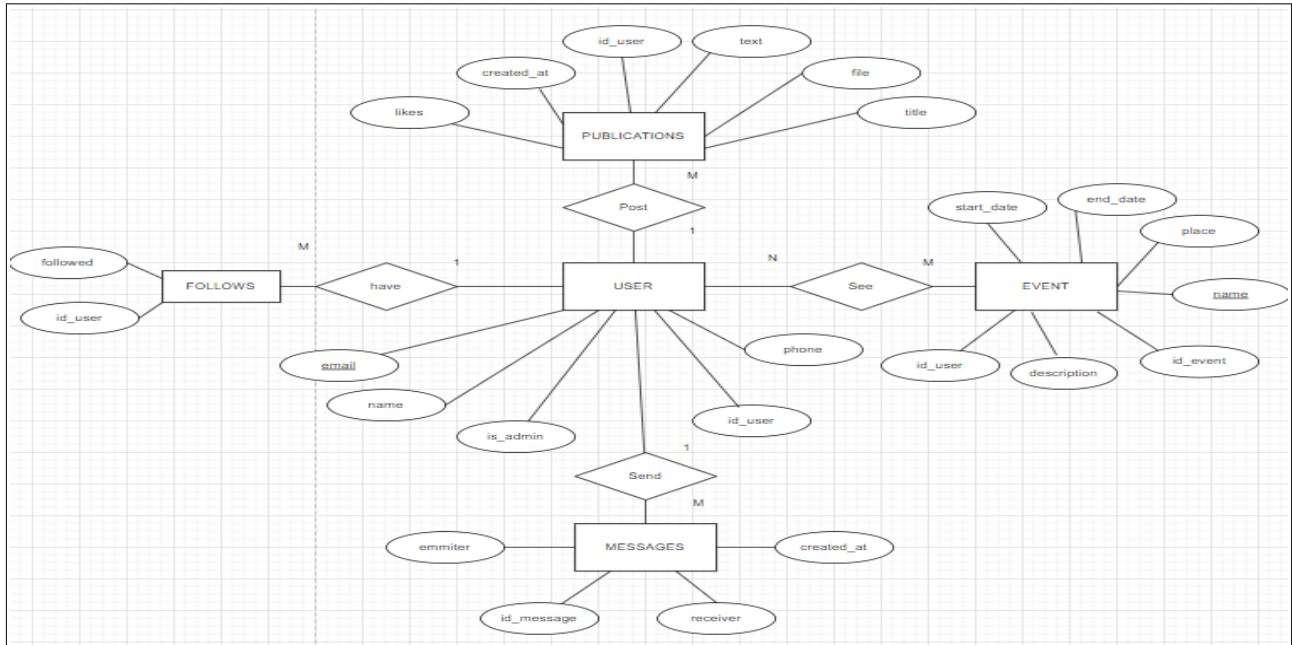


-Final:



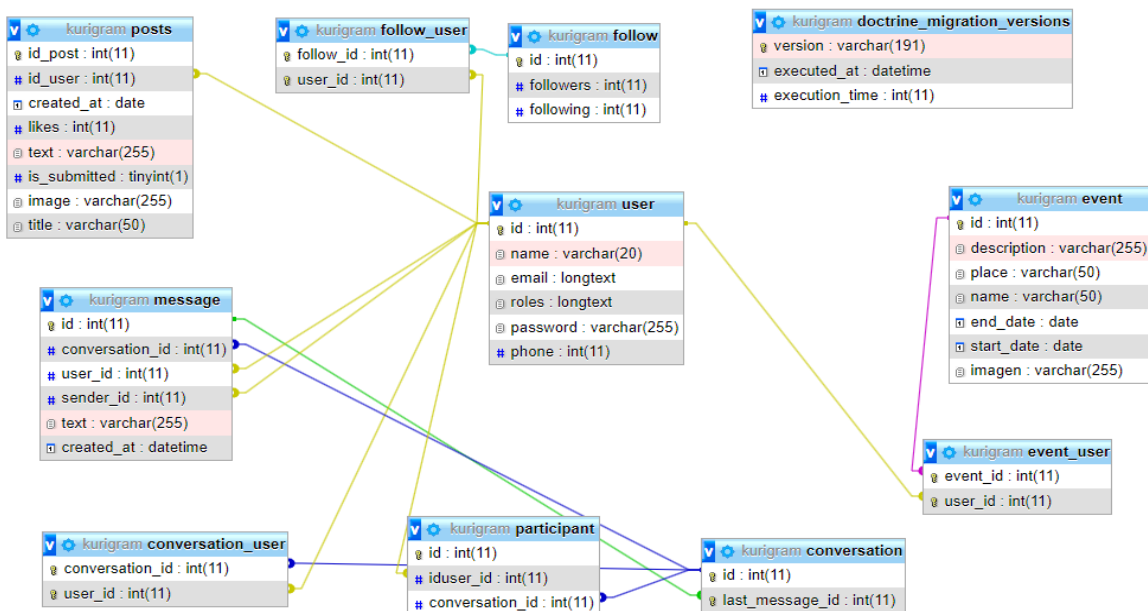
## Base de datos

-Diagrama E-R inicial:



En este diagrama podemos encontrar las tablas User, Follows, Event, Messages y Publications con sus respectivas relaciones y cardinalidades

-Resultado final de la base de datos con campos relacionados:



En este caso debemos considerar que ha habido que cambiar la tabla messages, ya que para la futura funcionalidad de mensajería, debemos guardar los participantes, la conversación, el contenido y el mensaje.

También cambios como la tabla publicaciones, que ahora se llama posts, e incluso encontrando como ahora las tablas tienen más campos. Esta modificación está planteada, de esta manera, debido a que está abierta a modificaciones, a un desarrollo y funcionalidades que abarquen una mayor cantidad de opciones.

Interesante marcar, que la tabla vacía y no relacionada, es auto-generada por doctrine en las relaciones muchos a muchos , pero no realiza ninguna función que necesitamos conocer.

## **Guía de estilos**

Para la realización de el proyecto, se han seguido una serie de reglas, marcadas en una guía de estilos. Esta contempla todas las reglas básicas sobre estilizado, maquetación, posicionamiento de elementos y además, aporta un valor estratégico, e identificativo de la aplicación, con el objetivo de poder seguir las pautas para combinar el trabajo con otras estrategias de el producto.

Cabe recalcar, que en todo momento, esta guía es un documento vivo, el cual puede crecer y cambiar.

En este apartado se va a citar brevemente cuales han sido estas:

### **Tipografía:**

Se ha utilizado dos principalmente, Roboto y Poppins.

Legibilidad: Ambas fuentes están diseñadas para ser altamente legibles en diferentes tamaños y dispositivos. Tienen una excelente claridad y facilidad de lectura en pantalla, lo que las hace ideales para proyectos web donde la legibilidad del texto es crucial.

Compatibilidad: Roboto y Open Sans son fuentes web ampliamente compatibles y se pueden renderizar correctamente en la mayoría de los navegadores y sistemas operativos modernos. Esto asegura una experiencia consistente para los usuarios, independientemente de la plataforma que estén utilizando

**Versatilidad:** Estas fuentes son versátiles y funcionan bien en una amplia gama de contextos y estilos de diseño. Tanto Roboto como Open Sans ofrecen múltiples pesos y estilos, lo que permite una gran variedad de opciones de diseño y jerarquía visual.

**Diseño moderno:** Ambas fuentes tienen un aspecto moderno y limpio que se adapta bien a los diseños contemporáneos. Roboto es la fuente predeterminada en el sistema operativo Android, por lo que a menudo se utiliza para crear una apariencia coherente en las aplicaciones y sitios web que se dirigen a ese sistema operativo.

### **Colores:**

#0E2439

Es un tono oscuro orientado a delimitar la sección header, a su vez permite contrastar.

#1190CB

Tono intermedio para diseño minimalista y limpio implementado todo generalmente en todo lo que tenga que ver con la marca debido a su relación con el logo

#61ABD4

Tono pastel para diseño minimalista y limpio , poco agresivo y combinativo implementado para botones y funciones extra

### **Tamaños:**

Para los títulos : 24 píxeles

Para los textos normal: 16 píxeles

Para los títulos de Segundo nivel: 20 píxeles

### **Accesibilidad:**

Para la adaptabilidad de la web, a parte de hacer diseño adaptativo para todo tipo de teléfonos, Nos comprometemos a la adaptación de contenidos, teniendo en cuenta las condiciones del usuario final:

Visibilidad reducida

Dificultad de aprendizaje

Ceguera

Todo ello para favorecer una experiencia gratificante de el usuario final, su crecimiento y/o desarrollo de conocimientos. Cumpliendo su reconocimiento como página educativa, tanto en niveles funcionales, como de conceptos finales.

## **Problemas/dificultados encontrados durante el desarrollo:**

Cors Policy a el conectar Symfony con Angular. Este error fue solucionado, accediendo a el repositorio de github de nelmioextrabundle, y implementandolo en el backend. Aparte se añadieron a los endpoints varias cabeceras.

Uso de Guards en Angular para restringir el acceso a un usuario, si no hay un registro o inicio de sesión en medio de estas acciones.

Restringir acceso en el panel de administración a los usuarios.

Funcionalidad seguir/dejar de seguir (lado Cliente).

Seguir y dejar de seguir usuarios (lado Admin).

Compartir carpeta que guarda imágenes.

## **Análisis de el resultado:**

### **Conclusiones:**

Este proyecto ha cumplido los requisitos mínimos necesarios, para que la web sea funcional, usable, y además se ha acercado a los requisitos deseables, en caso de superar las expectativas, las cuales serán implementadas en una línea cara a futuro.

Esto da como resultado una página web cliente favorecida por una experiencia de usuario favorecida, por otras similares. Que cumple con lo prometido en ese análisis de requisitos desarrollado con anterioridad, pero con una ligera falta de características, que no han podido desarrollarse durante este periodo de trabajo.

En cuanto a el desarrollo , han habido funcionalidades que eran primordiales antes que cualquier otra funcionalidad extra. El framework de php utilizado ha sido una delicia, facilitando mucho ciertas funcionalidades de el panel de administración.

Cabe recalcar, que el resultado es satisfactorio, cumpliendo con las necesidades y/o requerimientos mínimos de el proyecto. Todas las otras funcionalidades restantes, quedan cedidas a la línea de desarrollo a futuro.

### **Partes Positivas:**

He consolidado conceptos adquiridos durante el grado superior, he aplicado conceptos aprendidos durante las FTC's, y he aprendido a buscar y solucionar errores, reduciendo las horas muertas de trabajo. También he podido demostrarme a mi mismo, mis capacidades a la hora de lanzar una señora aplicación.

### **Partes Negativas:**

Ciertas funcionalidades han generado problemas y han ralentizado el rumbo de el proyecto, al igual que el desconocimiento de ciertas funcionalidades desarrolladas.

## Desarrollo largo plazo:

Para el futuro están consideradas varias funcionalidades, estas pueden ser divididas en dos, lado cliente, y lado de administración.

En el lado servidor encontramos en el apartado de comentarios, el cual trataría de una sección para poder revisar ciertos mensajes entre usuarios, en caso de ocurrir algún problema, los administradores podrían tomar acciones legales. (requeriría consentimiento de almenos un usuario)

En el apartado de el perfil de el usuario, contaríamos con la capacidad de poder editar o borrar nuestro perfil o cuenta.

En el lado Cliente sería donde más funcionalidades podríamos realizar.

En el apartado de ajustes, podríamos realizar cambios, como cambiar el tema de la interfaz a oscuro, solicitar ayuda enviando un mensaje a los administradores, cambiar el idioma utilizando la dependencia en angular llamada ngx translate .

En el apartado de nueva publicación, el formulario nos permitiría insertar un nuevo post, guardando la imagen en un apartado de la página web, compartido con Symfony, de manera que los dos pudieran visualizar la imagen insertada.

En la zona de perfil, podríamos consultar nuestra información, así como nuestros seguidores, likes totales recibidos etc.

En mensajes, podríamos enviar, recibir o consultar nuestros últimos mensajes con otros usuarios.

En el apartado de gente, podríamos seguir o dejar de seguir usuarios de forma final.

## Bibliografía o webgrafía:

Memoria→ <https://www.esdesignbarcelona.com/actualidad/disenio-publicitario/8-pasos-seguir-para-hacer-una-memoria-tecnica-de-diseno>



CONSULTA DE EJEMPLOS DE BBDD PARA MONTAR LA SQL:

<https://evilnapsis.com/2019/05/08/modelo-de-base-de-datos-de-un-sistema-de-red-social-basico/>

COMPORTAMIENTO DE EL USUARIO EN GENERAL:

<https://es.semrush.com/blog/usuarios-de-redes-sociales/>

COMO ESTAMOS ATADOS POR LAS REDES SOCIALES:

<https://www.elcorreo.com/vivir/tecnologia/consiguen-enganchemos-redes-20210226131312-ntrc.html>

Hacer un header y un footer completo

<https://madridnyc.es/elementos-header-y-footer-web/>

Porque las redes sociales no tienen footer?

Leído en libro de WORLD SOCIAL acompañado de un documento de footer recogido y corroborado desde el buscador y mediante bing tools.

Ley de protección de datos digitales:

[https://www.cookiebot.com/es/lopdgdd/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=esgeneric&utm\\_device=c&utm\\_term=ley%20de%20proteccion%20de%20datos%20espa%C3%B1a&utm\\_content=esspalopdlopdgdd&matchtype=e&gclid=CjwKCAjw3ueiBhBmEi wA4BhspJA6tJGEOct68cfZqvqOEluV2e7zGjX9XUKPEPNMQayaOJDmIcDfPRoCmzMQA vD\\_BwE](https://www.cookiebot.com/es/lopdgdd/?utm_source=google&utm_medium=cpc&utm_campaign=esgeneric&utm_device=c&utm_term=ley%20de%20proteccion%20de%20datos%20espa%C3%B1a&utm_content=esspalopdlopdgdd&matchtype=e&gclid=CjwKCAjw3ueiBhBmEi wA4BhspJA6tJGEOct68cfZqvqOEluV2e7zGjX9XUKPEPNMQayaOJDmIcDfPRoCmzMQA vD_BwE)

Gráfico de edades de uso de redes en auge:

pagina web -> <https://www.primeweb.com.mx/tiktok-estadisticas>

página web -> <https://yiminshum.com/social-media-redes-sociales-2021/>

## Glosario de términos:

-Template o plantilla: son un conjunto de código html, el cual puede implementar una serie de hojas de estilo, y que sirve como estructura visual tanto inicial como final.

-Framework→ conjunto de librerías de un lenguaje de programación, el cual permite muchas facilidades a el desarrollador, con la finalidad de facilitar el trabajo, o realizar tareas complejas.

-Metodología ágil:La metodología ágil es un enfoque de gestión de proyectos que se centra en la entrega de valor de manera rápida y continua, a través de la colaboración y la flexibilidad en el proceso de desarrollo.

-Scrum: és una metodología de gestión de proyectos ágil utilizada para desarrollar productos complejos de forma iterativa e incremental. Se basa en equipos autoorganizados y multifuncionales, que trabajan en sprints o iteraciones cortas de 1 a 4 semanas de duración.

-E-R:Un diagrama E-R es una herramienta visual que representa las entidades, atributos y relaciones de un sistema en la modelización de datos. Su objetivo es mostrar cómo están relacionados los elementos del sistema para una mejor comprensión y planificación.

-D.A.F.O: es un acrónimo que representa las siglas de las palabras Diseño, Análisis, Fabricación y Operación. Es una metodología de trabajo que se utiliza en la gestión de proyectos y la mejora continua de procesos, con el objetivo de optimizar la eficiencia y calidad del resultado final.

## Anexo 1:

```
export class GuardControlGuard {  
  constructor(private request: RequestService, private router: Router) {}  
  
  canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean |  
  UrlTree {  
    if (this.request.getCurrentUser()) {  
      return true;  
    } else {  
      this.router.navigate(['login']);  
      return false;  
    }  
  }  
}
```

Guard en angular. (Funciona como guardián de Acceso de usuarios a la aplicación).

## Anexo 2:

```
access_control:
  # - { path: /twig/listUser/, roles: [ADMIN, ROLE_USER] }
  # - { path: /twig/userPosts/*, roles: [ADMIN, ROLE_USER] }
  # - { path: /twig/detailUser/*, roles: [ADMIN] }
  # - { path: /twig/insertUser/*, roles: [ADMIN] }
  # - { path: /twig/deleteUser/*, roles: [ADMIN] }
  # - { path: /twig/updateUser/*, roles: [ADMIN] }
  # - { path: /twig/updateEvent/*, roles: [ADMIN] }
  # - { path: /twig/insertEvent/*, roles: [ADMIN] }
  # - { path: /twig/deleteEvent/*, roles: [ADMIN] }
  - { path: /twig, roles: [ADMIN, ROLE_USER] }
```

Seguridad en el Back-end (Funciona como restricción, si el usuario no es administrador)