



# NINEIMGBOARD

Borrador Memoria TFG

Clara Bujeda Muñoz

Tutora: Rosana Marín Berraondo  
Centro: Salesianos Nuestra Señora del Pilar  
Grado superior en aplicaciones multiplataforma



---

## RESUMEN

---

Un Image Board o tablón de imágenes llamado en español, es simplemente una aplicación web en la cual cualquier usuario puede publicar imágenes llamadas posts, así como buscar diferentes imágenes como medio de inspiración.

El primer tablón de imágenes llamado 2channel surgió en 1999, el cual sigue activo en la actualidad nombrado como 5ch. los últimos tableros de imágenes fueron creados sobre 2013 los cuales usualmente usan tecnologías como php que actualmente son inseguras y desfasadas, en este TFG se desea realizar un sistema de tablón de imágenes con el uso de tecnologías actuales como por ejemplo SpringBoot y Angular para mejorar el rendimiento y seguridad de la aplicación web. Además de lo anterior se pretende mejorar el aspecto visual usando frameworks debido a que usualmente los tableros de imágenes se constituyen con css básico.

# INDICE

## Índice General

1. Introducción .....	6
1.1.Contexto .....	6
1.2.Objetivos .....	7
1.3.Motivación .....	7
1.4.Tecnologías usadas.....	8
1.5.Organización de la memoria.....	9
2. Análisis del sistema .....	10
2.1.Sistema Inicial .....	10
2.2.Requisitos mínimos .....	11
2.3. Identificación de los actores .....	12
3. Arquitectura del sistema.....	15
3.1.Arquitectura Global .....	15
3.2.Diagrama estructural .....	16
3.2.1.General .....	16
3.2.2.Backend .....	17
3.2.3.Frontend.....	18
3.2.4.BBDD .....	19
4. Diseño e implementación del sistema .....	22
4.1.Datos y Estructura .....	22
4.2.Modelo lógico.....	22
4.3.Interfaces de usuario.....	23
5. Pruebas de funcionamiento .....	33
5.1.Explicación.....	33
5.2.Tabla de verificación .....	34
6. Conclusión.....	34
6.1.Resultados Obtenidos .....	35
6.2.Análisis de trabajo .....	36
6.3.Mejoras Posibles.....	38
6.4.Mercado.....	39
6.5.Tiempo estimado .....	41
6.6.Valoración Personal.....	42

---

7. Fuentes de Información.....	43
7.1.Bibliografía.....	43
7.2.Recursos .....	43
7.3.Documentación.....	43
8. Anexos.....	44
8.1. Índice de terminología.....	44
8.1. Ventana.....	46

---

## Índice de Figuras

Figura 2.1.1 Diagrama inicial usando Thymeleaf como front y SpringBoot .....	10
Figura 2.1.2 Diagrama ideático usado para el proyecto final .....	10
Figura 3.1.1 Arquitectura global del sistema .....	15
Figura 3.1.2 – Diagrama general de los servidores .....	16
Figura 3.2.1 Diagrama de movimiento de datos entre capas .....	17
Figura 3.2.2 Diagrama de obtención de datos y muestreo .....	18
Figura 3.2.3 Diagrama de la base de datos MySQL .....	20
Figura 3.2.4 Diagrama de la base de datos SQLite .....	21
Figura 4.2.1 Diagrama de flujo general de la aplicación .....	22
Figura 4.3.1 Ejemplo estructural de la pantalla home .....	23
Figura 4.3.2 Ejemplo de la pestaña posts .....	24
Figura 4.3.3 Ejemplo de los detalles de un post .....	25
Figura 4.3.4 Ejemplo de la ventana tags .....	26
Figura 4.3.5 Login en la aplicación .....	27
Figura 4.3.6 Registro en el sistema .....	28
Figura 4.3.7 Página 404 .....	29
Figura 4.3.8 Ventana Upload .....	30
Figura 4.3.9 Panel de usuario .....	31
Figura 4.3.10 Panel del sistema administrador .....	32
Figura 4.3.11 Panel de usuarios Idea general .....	32

## Índice de Tablas

Tabla 1.2.1 - Tabla de las tecnologías usadas .....	8
Tabla 2.2.1 – Resumen de requisitos mínimos .....	11
Tabla 5.2.1 – Pruebas Unitarias realizadas .....	34
Tabla 5.2.2 – Pruebas Visuales .....	34
Tabla 6.3.1 – Mejoras posibles .....	38
Tabla 6.5.1 – Tiempo estimado .....	41
Tabla A.1 – Índice de terminología .....	45

---

# Capítulo 1

## Introducción

---

Este documento tiene como objetivo brindar una explicación completa y detallada de la conformidad del proyecto NIB, así como de su estructura y los motivos que llevaron a su desarrollo. Para ello, se llevará a cabo un análisis exhaustivo del proyecto, que permitirá comprender de manera clara y concisa su alcance y objetivos.

Además, se abordarán aspectos específicos relacionados con la implementación y el desempeño del proyecto, incluyendo el diseño de la arquitectura, la metodología utilizada, la gestión de recursos y el control de calidad. También se describirá el proceso de evaluación de la conformidad del proyecto, así como las herramientas y tecnologías utilizadas para este fin.

En definitiva, este documento será una guía completa y detallada para comprender en profundidad la conformidad del proyecto NIB, y para valorar su relevancia y contribución en el ámbito en el que se desarrolla.

### 1.1. Contexto

Este proyecto se desarrolla de forma unipersonal en el centro escolar Salesianos Nuestra Señora del Pilar

---

## 1.2. Objetivos

Los objetivos en esencia de esta TFG es la experimentación de crear una aplicación web de tablón de imágenes usando tecnologías actuales para mejorar tanto el aspecto visual como el aspecto técnico, a continuación, se citan los objetivos generales que se espera conseguir con dicho proyecto.

1. Aprendizaje y funcionamiento de una web de tableros de imágenes así como podría ser <https://safebooru.org/> .
2. Mejorar diseño visual y estructura inspiradas además de modernizarlas.
3. Diseñar la aplicación en un sistema MVC para la facilidad de mantenimiento y actualización.
4. Estudiar un sistema de almacenamiento de datos MIME mediante guardado en BLOB en ficheros bd .
5. Estudiar un sistema para la separación del sistema de almacenamiento de binarios y datos textuales.

## 1.3. Motivación

Usualmente navegando por tableros de imágenes siempre me pregunte cuál era su funcionamiento, como comprimían los datos, como los enviaban o como almacenaban todo, en ese momento intente crear mi propio tablero de imágenes que surgió como un prototipo el cual solamente se podían subir imágenes y las guardaba en un sistema de ficheros, pero esto es simplemente ineficiente así como afán de querer rehacer dicho proyecto me surgió la idea de volverlo a realizar en ese TFG.



## 1.4. Tecnologías usadas

En este proyecto se ha querido optar por el uso de nuevas y diferentes tecnologías

Tipo	Tecnología	Definición
Server Backend	SpringBoot	Server el cual gestionará la api que permitirá la salida y la entrada de datos. Además de esto tramitará a el servidor de datos y a los diferentes servidores binarios. También tendrá la función del núcleo del programa.
Server Frontend	Angular	Server el cual gestionará el frontend de la web.
Server Datos	MySql	Server el cual almacenará y proveerá los datos de los usuarios además de información relacionadas con los posts y tags.
Servers Binarios	SQLite	Servers los cuales almacenarán los binarios de las imágenes de los usuarios para liberar tensión en el servidor MySql.
API S-A	API RESTful (JSON)	Método de comunicación entre el servidor SpringBoot y el Servidor Angular
Diseño	BootStrap	Framework con el cual se gestionará el diseño de la web (También se usará css puro para el funcionamiento correcto de la web)
Lanzadores	C# Windows Forms	Se uso C# Para la programación de un lanzador de la aplicación en un entorno visual o GUI para una mayor facilidad para la ejecución.
	.BAT / CMD	Se escribió un script para el lanzamiento del programa en modo consola para un entorno de Windows.
	.Sh / Bash	Se escribió un script para el lanzamiento del programa en modo consola para un entorno Linux.

Tabla 1.2.1 - Tabla de las tecnologías usadas

---

## **1.5. Organización de la memoria**

El primer capítulo de la memoria se ha dedicado a proporcionar una breve introducción y presentación del trabajo realizado. En el segundo capítulo se ha llevado a cabo un análisis exhaustivo que ha sido útil para establecer los objetivos y funcionalidades de la aplicación web. El tercer capítulo incluye una descripción detallada de la arquitectura general del proyecto, así como la arquitectura específica de la nueva aplicación web desarrollada. En el cuarto capítulo se detallan los pasos llevados a cabo en la segunda fase del proyecto, incluyendo la definición de los procesos involucrados, el modelo lógico de los datos y el diseño de la interfaz de usuario. Posteriormente en el quinto capítulo presenta todos los resultados obtenidos, proporciona una lista de posibles trabajos futuros y realiza una valoración personal sobre el trabajo realizado. En el capítulo sexto se proporciona una conclusión sobre los resultados finales de la aplicación y por último en el séptimo capítulo se muestra los diferentes recursos y fuentes de información usadas.

## Capítulo 2

### Análisis del sistema

#### 2.1. Sistema Inicial

En un inicio el proyecto se ideó inicialmente con un servidor SpringBoot el cual almacenaba los datos en un Servidor MySQL y los archivos binarios en una estructura de archivos local tal y como se muestra en la figura posterior.

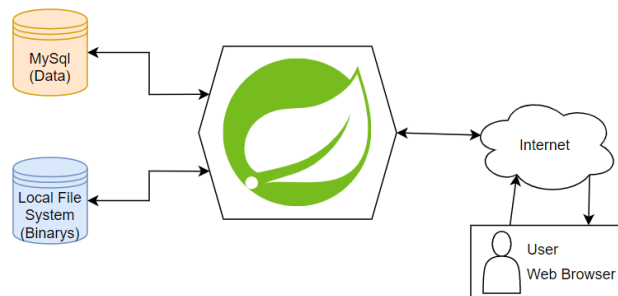


Figura 2.1.1 Diagrama inicial usando Thymeleaf como front y SpringBoot

Posteriormente al desear un mejor rendimiento tanto para el usuario como para la parte de datos y servidor se prefirió usar SpringBoot como servidor Backend y Angular para el servidor del front debido a que este tiene gran flexibilidad para la carga veloz de vistas webs, además de lo anterior se reemplazó el sistema de ficheros local con múltiples archivos SQLite debido a la facilidad de organización y mejora de rendimiento en cuanto a la búsqueda de los binarios tal y como se muestra en la siguiente figura.

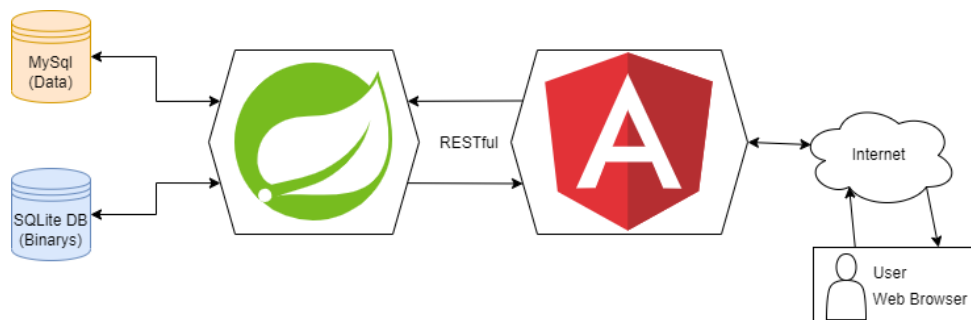


Figura 2.1.2 Diagrama ideático usado para el proyecto final

## 2.2. Requisitos mínimos

En este apartado se especificarán los requisitos mínimos necesarios para garantizar el correcto funcionamiento de la aplicación web en su servidor o computadora. Estos requisitos incluyen la versión mínima de los diferentes programas y otros requerimientos.

### Versión de software requerida:

- Java: Se requiere la versión 17 de Java para ejecutar la aplicación web sin problemas. Asegúrese de tener instalada esta versión en su sistema.
- Node.js: La versión mínima de Node.js necesaria es la v18.12.0. Este entorno de tiempo de ejecución es esencial para el funcionamiento adecuado de la aplicación web. Verifique que tenga instalada esta versión en su servidor o computadora.
- npm package manager: Para gestionar las dependencias de la aplicación web, se necesita el administrador de paquetes npm en su versión v8.19 o superior. Asegúrese de contar con esta versión instalada en su sistema.
- MySQL: Se requiere la versión 8 de MySQL como sistema de gestión de bases de datos para la aplicación web. Verifique que tenga instalada esta versión y que esté correctamente configurada en su servidor o computadora.

### Sistema operativo compatible:

- Windows 10 (+): Si su servidor o computadora utiliza el sistema operativo Windows, se recomienda que tenga instalada la versión 10 o superior para asegurar la compatibilidad y el rendimiento óptimo de la aplicación web.
- Linux: Si prefiere utilizar Linux como sistema operativo, asegúrese de tener una distribución compatible y actualizada para garantizar el correcto funcionamiento de la aplicación web aunque actualmente no se puede asegurar el correcto funcionamiento.

	Versión
Java	17
Node js	v18.12.0
Mysql	8
npm package manager	v8.19
SO	Windows 10 (+) / Linux

Tabla 2.2.1 – Resumen de requisitos mínimos

---

## 2.3. Identificación de los actores

En este apartado se expresa los diferentes tipos de usuarios y las acciones que pueden y no pueden hacer además de las páginas que contienen dichas funciones.

### Usuario No registrado

Un usuario en la aplicación podrá registrarse o no, en las siguientes tablas se muestra la identificación de los actores dependiendo de dichas causas.

#### ¿Qué puede hacer?

Un Usuario al conectarse a la aplicación web vera una pantalla principal en la cual podrá dirigirse a posts, logearse (login), registrarse (register), ver los tags o realizar una búsqueda.

#### Búsqueda

- En la pantalla principal al efectuar una búsqueda se le redirigirá al usuario a la ventana de posts mostrándole la búsqueda efectuada.

#### Posts:

- Al entrar en el apartado de posts el usuario podrá ver el escenario dividido en dos secciones en el primero en el cual podrá realizar una búsqueda y el segundo en el cual podrá visualizar dicha búsqueda, en caso de que el usuario no hubiera realizado una búsqueda, se le mostraran por default los posts más nuevos.
- En caso de clicar en un post se le redirigirá al usuario a una ventana de detalles del post en la cual podrá ver tanto la imagen como sus tags y su información específica.

#### Login

- En caso de entrar al apartado de login se le solicitará al usuario su nombre de usuario además de su contraseña, una vez hecho esto le redirigirá de forma automática a la pantalla principal de un usuario registrado.

#### Register

- Al acceder en al apartado de register se solicitará al actor su nombre de usuario además de su email y la contraseña, lo anterior de forma obligatoria, de forma opcional se le solicitará su nombre y sus apellidos.

#### Tags

En dicha ventana se le mostrará al usuario los tags disponibles para ejercer la búsqueda.

---

## Usuario Registrado

### ¿Qué puede hacer?

Un Usuario registrado será capaz de ver una pantalla principal en la cual podrá dirigirse a posts, Panel de usuario, Upload o la vista de los tags además de ser capaz de realizar una búsqueda.

#### Búsqueda

- En la pantalla principal al efectuar una búsqueda se le redirigirá al usuario a la ventana de posts mostrándole la búsqueda efectuada.

#### Posts:

- Al entrar en el apartado de posts el usuario podrá ver el escenario dividido en dos secciones en el primero en el cual podrá realizar una búsqueda y el segundo en el cual podrá visualizar dicha búsqueda, en caso de que el usuario no hubiera realizado una búsqueda, se le mostraran por default los posts más nuevos.
- En caso de clicar en un post se le redirigirá al usuario a una ventana de detalles del post en la cual podrá ver tanto la imagen como sus tags y su información específica además de poder comentar dicho post.

#### Panel de Usuario

- En dicha ventana el usuario deberá de ser capaz de modificar su configuración de usuario así como personalizar sus datos.

#### Upload

- La función de "Upload" es una herramienta esencial en cualquier aplicación web que permita a los usuarios compartir contenido con la comunidad. La ventana de "Upload" es un espacio donde los usuarios registrados pueden crear y subir un nuevo post.
- Esta función es especialmente útil para aquellos que quieren compartir sus ideas, experiencias o conocimientos con el resto del mundo. Con la posibilidad de agregar etiquetas, títulos y descripciones, el usuario puede personalizar su post con tags para que sea más fácilmente encontrado por otros usuarios interesados en el mismo tema.

#### Tags

- En la ventana de tags, el usuario puede encontrar una lista de palabras clave que se utilizan para categorizar y etiquetar los recursos en la plataforma. Estos tags están diseñados para ayudar al usuario a encontrar posts que mejor se adapten a sus.

---

## **Administrador**

Un administrador es aquel usuario el cual tendrá acceso a las funciones de control total de la aplicación.

### **¿Qué puede hacer?**

Un administrador debe de ser capaz de efectuar lo mismo que un usuario además de tener la capacidad de eliminar usuarios, eliminar posts y modificar la configuración asimismo de tener la capacidad de retocar los atributos del mismo.

### **Panel de administrador**

- En dicho panel el Administrador será capaz de modificar los datos del sitio web así como su configuración.
- Además de lo anterior en dicho panel será capaz de banear usuarios, una vez baneado un usuario constará como un usuario anónimo y no modificable.

## Capítulo 3

### Arquitectura del sistema

En este capítulo se describe la arquitectura del sistema mencionado.

#### 3.1. Arquitectura Global

El programa es capaz de correr sus partes tanto en el mismo servidor como en diferentes servidores tal y como se indica en la figura posterior.

La arquitectura en su esencia consta en 3 partes: BackEnd y Datos MIME, FrontEnd y por último datos de la aplicación. Todo esto esta maquetado con la estructura Modelo vista controlador (MVC) lo cual consta en que los datos, lo que controla los datos y lo que ve el usuario este estructurado de forma separada, con lo que conseguimos un mayor control en la estructuración del programa, mayor facilidad de actualización, arreglo de bugs y errores además de mayor seguridad en el sistema.

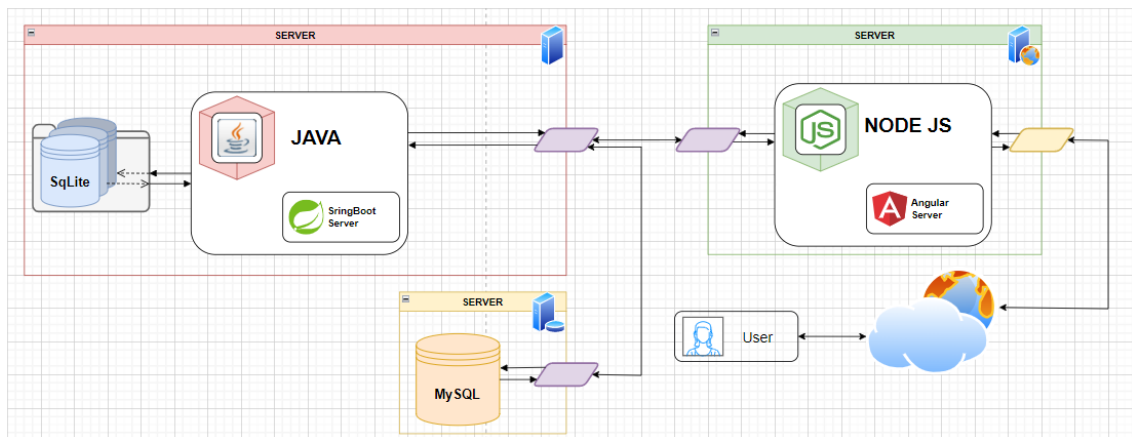


Figura 3.1.1 Arquitectura global del sistema



## 3.2. Diagrama estructural

En esta sección se presenta una representación gráfica de la estructura general del proyecto. El diagrama proporciona una visión detallada de las relaciones entre los diferentes componentes y módulos del sistema, así como su jerarquía y dependencias.

### 3.2.1. General

En esta sección se muestra la estructura general de la aplicación web la cual esta compuesta por diferentes partes.

- Datos: Dicha parte almacena los diferentes datos la cual consta de dos bases de datos, la primera MySQL la cual almacenara datos textuales y la segunda SQLite la cual almacenara principalmente datos de tipo BLOB.
- BackEnd: Esta sección de la aplicación se encarga del procesamiento de los datos y de proveerlos al frontend en caso de ser necesario, esto anterior esta desarrollado en SpringBoot debido al ser capaz de soportar más carga.
- FrontEnd: Esta área de la aplicación muestra los datos al usuario que han sido proveídos por el backend.

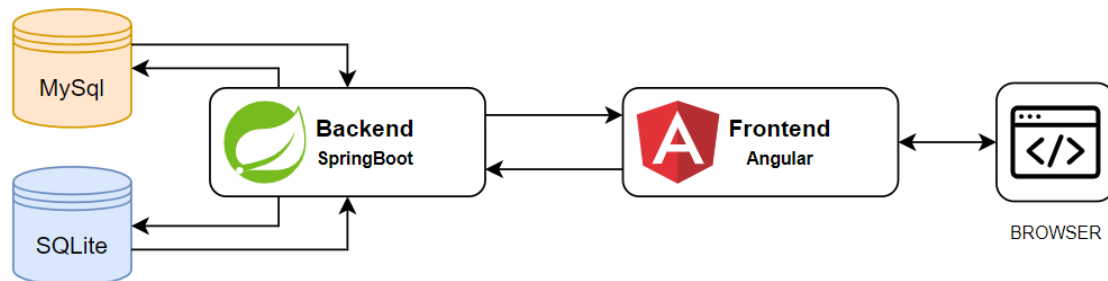


Figura 3.1.2 – Diagrama general de los servidores

### 3.2.2. Backend

El BackEnd de este proyecto ha sido desarrollado en SpringBoot.

Se ha elegido esta tecnología por varias razones, entre las que destaca su capacidad para facilitar la creación de aplicaciones robustas, escalables y seguras.

La estructura del backend ha sido subdividida en diferentes secciones. Los controladores los cuales manejan el uso de los datos proveyendo una api, los servicios e implementaciones los cuales manejan los datos que usan los controladores y por último la dao la cual usa JPA (CRUD) para comunicarse con las diferentes bases de datos usadas.

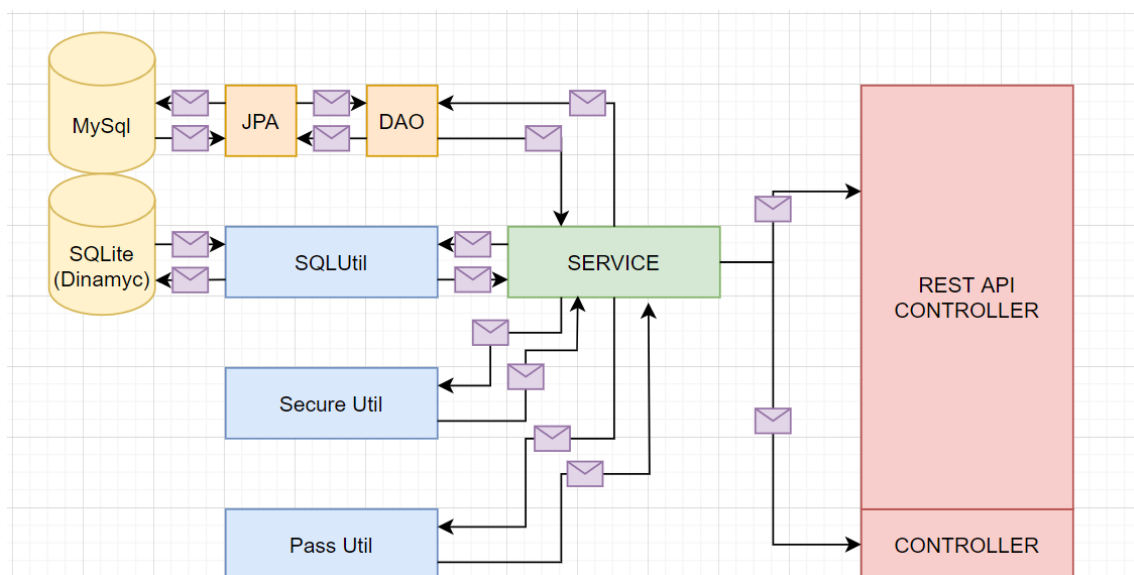


Figura 3.2.1 Diagrama de movimiento de datos entre capas

### 3.2.3. Frontend

El FrontEnd de este proyecto se ha realizado en Angular debido a sus numerosas ventajas y funcionalidades avanzadas que lo convierten en una herramienta ideal para el desarrollo de interfaces de usuario modernas y complejas.

Para la realización de dicha sección se ha usado los siguientes elementos propios del framework. Componentes, los cuales almacenan paginas ya sea para usarlas como layouts o como una ventana de la web, y servicios los cuales obtienen los datos, los procesan y los usan los componentes, además de dichos elementos propios, existen otros como archivos de configuración y diferentes scripts Ts.

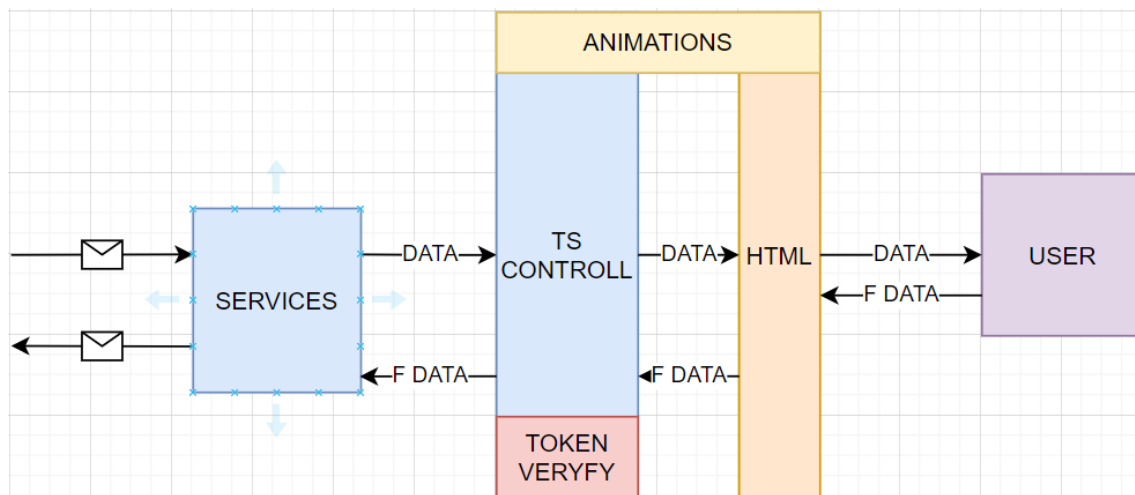


Figura 3.2.2 Diagrama de obtención de datos y muestreo.

---

### 3.2.4. BBDD

#### MySql

En la base de datos existen las siguientes tablas: usuario, post, tags, roles, comments, insult list. Cada una de estas tablas desempeña un papel fundamental en el funcionamiento y la organización de la base de datos.

La tabla "usuario" almacena la información de los usuarios registrados en el sistema. Aquí se guardan los datos como el nombre, dirección de correo electrónico, contraseña y otra información relevante que identifica a cada usuario de manera única.

La tabla "post" se utiliza para almacenar las publicaciones realizadas por los usuarios. Cada entrada en esta tabla contiene detalles sobre el contenido del post, como el título, el cuerpo del texto, la fecha de publicación y cualquier otro dato asociado.

La tabla "tags" permite categorizar los posts en función de palabras clave o etiquetas. Los tags proporcionan una forma eficiente de organizar y buscar contenido relacionado en la base de datos. Cada tag tiene una relación con uno o varios posts, lo que facilita la clasificación y el filtrado de información.

La tabla "roles" se utiliza para asignar diferentes niveles de permisos y privilegios a los usuarios. Por ejemplo, un usuario puede tener el rol de administrador, lo que le otorga acceso a funciones y acciones adicionales en el sistema. Los roles ayudan a gestionar la seguridad y la autorización en la base de datos.

La tabla "comments" se encarga de almacenar los comentarios realizados por los usuarios en los posts. Aquí se guardan los datos relacionados con cada comentario, como el autor, el contenido del comentario, la fecha y cualquier otro atributo relevante.

Por último, la tabla "insult list" se utiliza para almacenar una lista de insultos o palabras ofensivas que deben ser filtradas o moderadas en los comentarios o posts. Esta lista ayuda a mantener un ambiente respetuoso y seguro en la plataforma, evitando el contenido inapropiado.

En resumen, estas tablas en la base de datos sirven actualmente para organizar y gestionar información sobre los usuarios, publicaciones, etiquetas, roles, comentarios y controlar el contenido ofensivo, proporcionando una estructura sólida y eficiente para el funcionamiento del sistema.



## SQLite

Para lograr una eficiente gestión de datos, se ha implementado la creación de una base de datos de LiteSql para cada usuario en nuestro sistema. Estas bases de datos individuales almacenan información clave que nos permite organizar y acceder de manera efectiva a los datos relacionados con cada usuario.

En cada una de estas bases de datos, se guardan tres elementos fundamentales: el identificador externo del post (iddb\_binary), el binario asociado y el identificador del sistema. Estos datos nos permiten realizar un seguimiento preciso de la información relacionada con cada usuario y asegurarnos de que los datos estén correctamente enlazados y disponibles para su consulta.

Cada base de datos individual se nombra siguiendo una convención específica. El nombre de archivo se compone del "idw" del usuario, que es un identificador único asignado a cada usuario en nuestro sistema, seguido de la extensión ".db". Esta estructura de nomenclatura nos ayuda a mantener un orden claro y organizado, facilitando la identificación de las bases de datos correspondientes a cada usuario.

Con esta implementación, podemos asegurar una gestión eficiente de la información, manteniendo la integridad de los datos y proporcionando un acceso rápido y preciso a los mismos. Cada usuario tiene su propio espacio individualizado dentro de la base de datos, lo que nos permite ofrecer un servicio personalizado y adaptado a las necesidades de cada usuario en nuestro sistema.

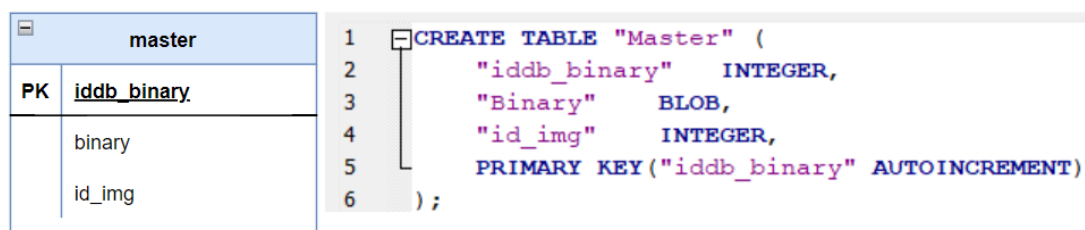


Figura 3.2.4 Diagrama de la base de datos SQLite

## Capítulo 4

### Diseño e implementación del sistema

#### 4.1. Datos y Estructura

La aplicación cuenta con varias tablas, cada una con un propósito específico y un conjunto de datos asociados. En estas tablas se almacenan diferentes tipos de información que son necesarios para el correcto funcionamiento de la aplicación.

#### 4.2. Modelo lógico

La aplicación web en general se compone en diferentes partes, debido a esto el ejecutable iniciara primero un configurador el cual establecerá las preferencias del sistema, y a posteriori iniciara los servidores en diferentes Threads esto en caso de que el inicio sea conjunto.

El Backend una vez iniciado entrara en modo de escucha para recibir diferentes peticiones y asi a posteriori responderlas.

El Frontend al iniciar esperara hasta que un usuario se conecte y a continuación solicitara los datos al backend y mostrara una página con los datos solicitados.

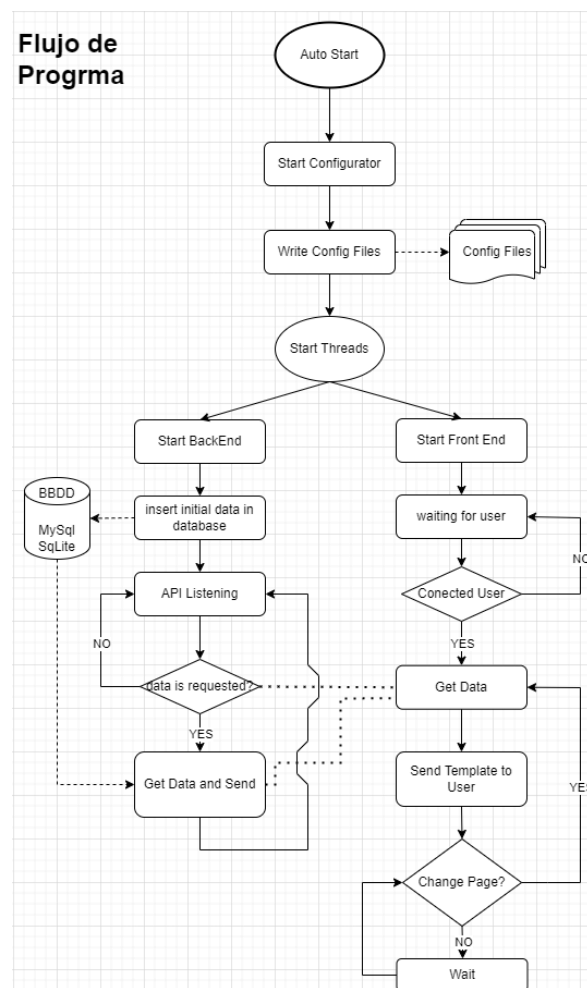


Figura 4.2.1 Diagrama de flujo general de la aplicación.

---

## 4.3. Interfaces de usuario

### Interfaz Publica

La interfaz de usuario publica consiste en las siguientes ventanas: Home, Posts, Tags y los diferentes formularios de inicio de sesión y registro.

#### Home

En la Pantalla **Home** podremos efectuar una búsqueda o navegar por las diferentes pantallas permitidas al usuario como se han indicado anteriormente, A continuación, en la siguiente figura se muestra dicha ventana.

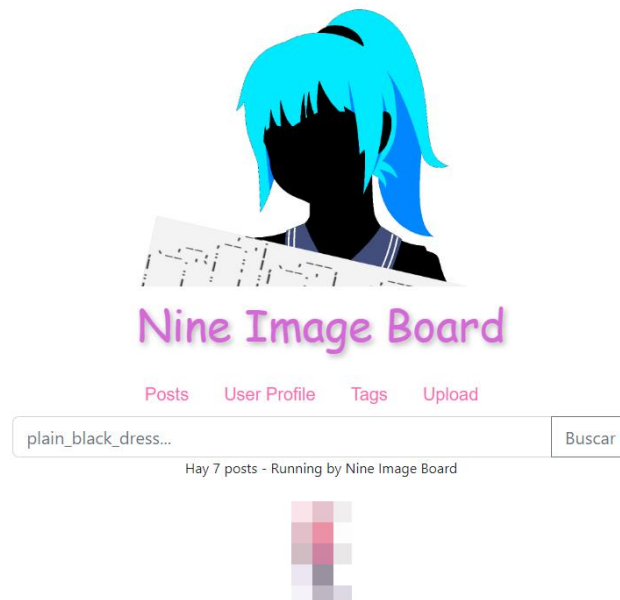


Figura 4.3.1 Ejemplo estructural de la pantalla home.



## Posts

Desde la anterior ventana el usuario (no registrado) podrá acceder a Posts o Realizar una búsqueda en la Search bar tal y como se muestra en la anterior figura, una vez accedido a algunas de las siguientes opciones se mostrara al usuario la ventana de posts que la cual en caso de haber accedido sin realizar una búsqueda pertinente se mostrara al usuario la ventana posts con los últimos agregados, por el contrario de haber realizado una búsqueda se mostrarán dichos posts según la búsqueda tal y como se puede observar en la siguiente figura.

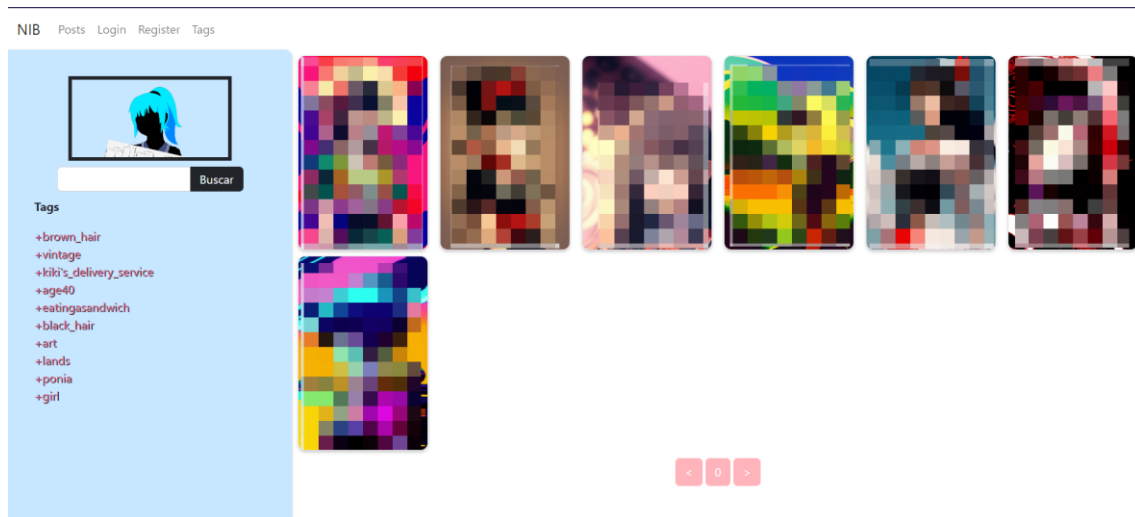


Figura 4.3.2 Ejemplo de la pestaña posts

## Detalles Posts

En la ventana "Detalles de posts", el usuario será capaz de ver información más detallada sobre un post en particular. Esta ventana suele incluir información adicional, como el título del post, la fecha de publicación, el autor, las etiquetas asociadas, el contenido del post y los comentarios relacionados.

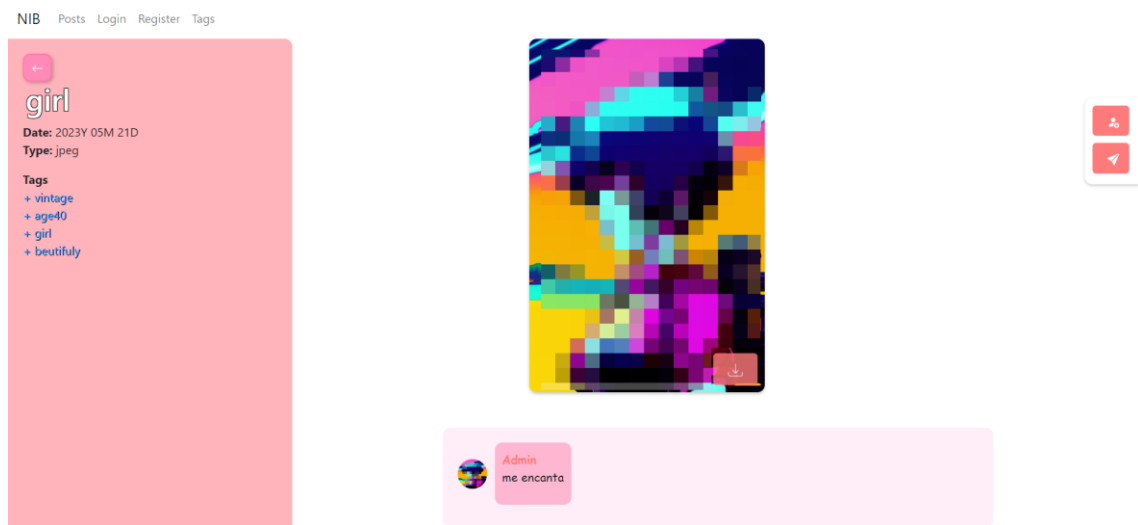


Figura 4.3.3 Ejemplo de los detalles de un post

## Tags

En la pestaña "Tags" del sistema, el usuario será capaz de visualizar las diferentes etiquetas o "tags" asociadas a los posts o publicaciones de la plataforma. Estas etiquetas son palabras o frases que se utilizan para clasificar y agrupar el contenido de la plataforma en función de su temática o contenido.

En la pestaña "Tags", el usuario puede explorar las diferentes etiquetas disponibles y seleccionar aquellas que le interesen para visualizar los posts relacionados.



Figura 4.3.4 Ejemplo de la ventana tags

## Login

En la ventana “**Login**” del sistema, el usuario es capaz de ingresar a su cuenta previamente registrada en la plataforma. Para poder acceder, se le solicita al usuario que ingrese su dirección de correo electrónico y su contraseña, los cuales son validados por el sistema antes de permitir el acceso a la cuenta.

Una vez que el usuario ha ingresado a su cuenta, tiene acceso a todas las funcionalidades y herramientas disponibles en la plataforma.

---

[NIB](#) [Posts](#) [Login](#) [Register](#) [Tags](#)

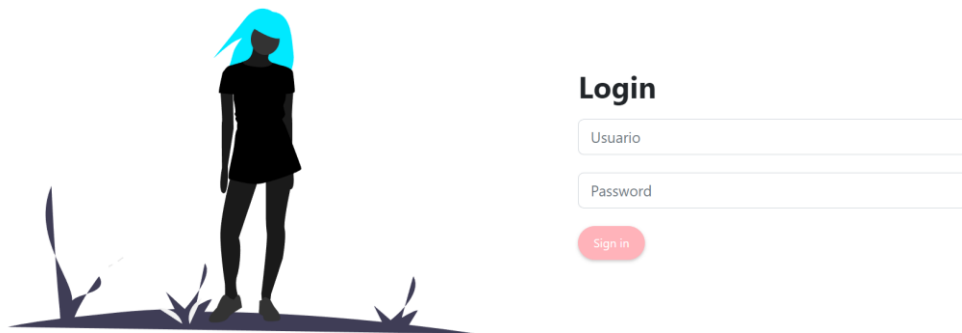



Figura 4.3.5 Login en la aplicación

## Register

En la ventana “**register**” del sistema, el usuario es capaz de realizar el proceso de registro para crear una cuenta en la plataforma. En esta ventana, se le solicita al usuario que ingrese cierta información personal, como su nombre completo (opcional), dirección de correo electrónico, y una contraseña segura.

Una vez que el usuario ha proporcionado esta información, el sistema valida que los datos ingresados cumplan con los requisitos establecidos y, en caso de ser correctos, procede a registrar la cuenta del usuario. Es importante destacar que el proceso de registro se realiza de manera segura, utilizando medidas de seguridad y encriptación para proteger la información personal del usuario.

NIB Posts Login Register Tags



### Register

Otros Datos

Nombre

Apellidos

Usuario

example@mail.com

Password

Register

Figura 4.3.6 Registro en el sistema

## Error 404

La pestaña "Error 404" del sistema es una página de error que se muestra cuando el usuario intenta acceder a una página que no existe en la plataforma. En esta sección, el usuario será capaz de visualizar un mensaje de error indicando que la página solicitada no se encuentra disponible.

Es una herramienta muy útil para mejorar la experiencia del usuario en la plataforma, ya que permite informar al usuario de manera clara y concisa que la página solicitada no existe, y evita confusiones o frustraciones por la falta de información.

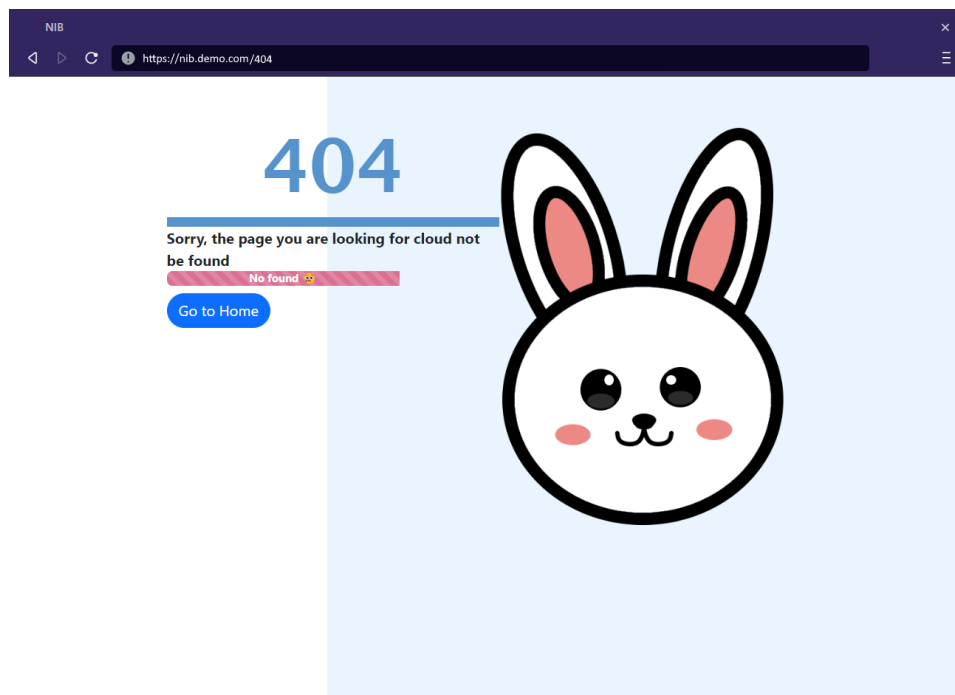


Figura 4.3.7 Página 404

A parte de lo anterior existen algunas páginas ocultas tal y como se describe en anexos apartado 2.2

## Interfaz Usuario Registrado

La interfaz de un usuario es ligeramente diferente debido a que se muestra ligeramente diferente a la interfaz publica, ya que no contiene las opciones de login y registro y en vez de ello contiene las opciones de Upload y panel de usuario además de que en la pantalla de los detalles de posts el usuario es capaz de comentar dichas publicaciones.

### Upload

En la ventana "Upload" el usuario tendrá la capacidad de subir posts con diferentes tags asociados, los cuales permitirán categorizar y organizar los posts de manera efectiva. Para esto, se incluirá un formulario donde el usuario podrá ingresar el título del post, su contenido y una lista de tags separados por comas.

---

NIB

Home

Posts


User Profile

Upload

Tags

Log Out

Upload



Seleccionar archivo7aa1...jpeg

Girl

Tags  
Vintage, 80's, anime, style\_vaporwave

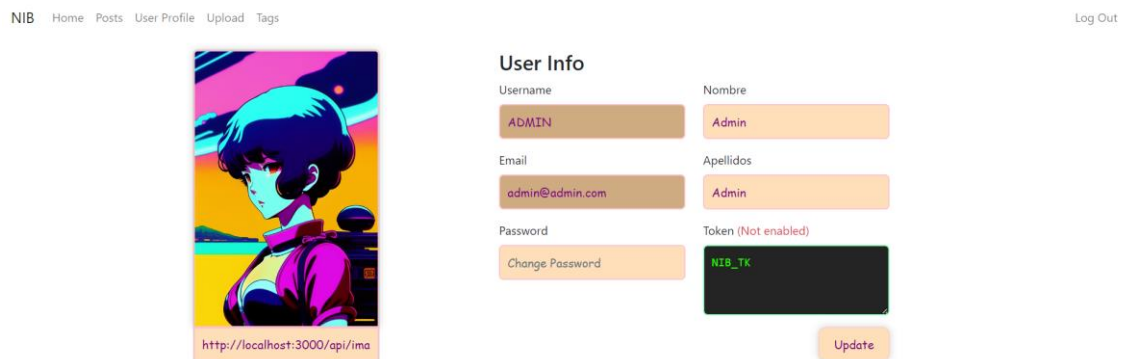
Las tags deberan de ser escritas separdas por comas y sin espacios. Ejemplo:  
Girl,Big\_eyes

Send Post

Figura 4.3.8 Ventana Upload

## Panel de usuario

En la ventana "Panel de usuario" el usuario tendrá acceso a diferentes opciones y herramientas para administrar su cuenta. Una de las opciones principales será la posibilidad de editar y actualizar su perfil, incluyendo información personal como su nombre, correo electrónico, imagen de perfil y contraseña.



The screenshot shows a web application interface for a user profile. At the top, there is a navigation bar with links: NIB, Home, Posts, User Profile, Upload, and Tags. On the right side of the navigation bar is a "Log Out" link. The main content area is titled "User Info" and contains two columns of form fields. The left column includes "Username" (ADMIN), "Email" (admin@admin.com), and "Password" (Change Password). The right column includes "Nombre" (Admin), "Apellidos" (Admin), and "Token (Not enabled)" (WIB\_TK). Below the token field is an "Update" button. On the left side of the main content area, there is a profile picture placeholder showing a stylized anime-style character with the URL http://localhost:3000/api/ima.

Figura 4.3.9 Panel de usuario



## Administrador

La interfaz de un usuario administrador es igual que la de un usuario registrado salvo que tiene un panel en el cual puede modificar los ajustes de la página web así como Panel de usuarios y funciones extras en algunas de las ventanas.

### Panel del Sistema

En el panel del sistema, el administrador tendrá acceso a una serie de configuraciones avanzadas que le permitirán personalizar y ajustar diversos aspectos de la plataforma según las necesidades de la organización.

Entre las configuraciones disponibles, se incluyen opciones para modificar el nombre de la web así como configuraciones internas del sistema.

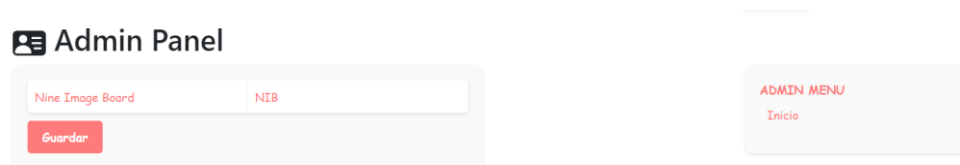


Figura 4.3.10 Panel del sistema administrador

### Panel de usuarios

En la ventana "Panel de usuarios", un administrador tendrá acceso a opciones adicionales de administración que no estarán disponibles para los usuarios regulares. Entre ellas, se incluirá la posibilidad de banear y eliminar a usuarios de la plataforma.

Cuando un usuario es baneado por un administrador, su cuenta se desactivará temporalmente y se le impedirá acceder a la plataforma.

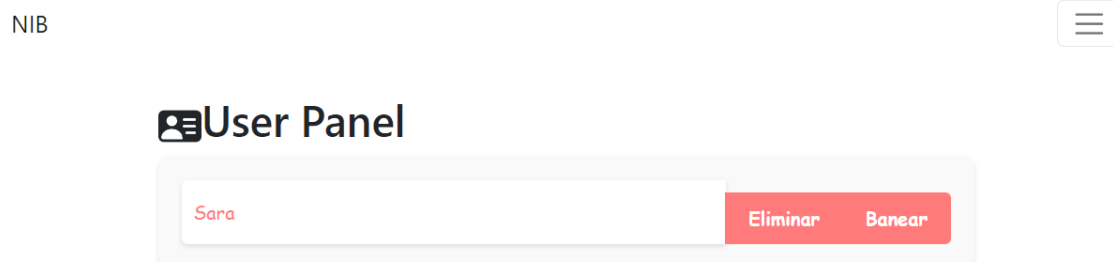


Figura 4.3.11 Panel de usuarios Idea general (no aplicada)

---

## Capítulo 5

### Pruebas de funcionamiento

---

#### 5.1. Explicación

Para garantizar el correcto funcionamiento de la aplicación, se llevarán a cabo dos tipos de pruebas: pruebas unitarias y pruebas visuales.

Las pruebas unitarias se realizarán utilizando `ng Test` y `JUnit`. Estas pruebas se enfocarán en evaluar el funcionamiento individual de cada componente de la aplicación. Mediante `ng Test`, se verificará que cada unidad de código, como funciones o métodos, produce los resultados esperados y maneja correctamente los casos límite y las situaciones de error. Por su parte, `JUnit` permitirá realizar pruebas específicas en el código Java, asegurando la calidad de las funcionalidades implementadas.

Además de las pruebas unitarias, se llevarán a cabo pruebas visuales durante el desarrollo de la aplicación. Estas pruebas se centran en evaluar la interfaz de usuario y la experiencia del usuario en general. Se comprobará que los elementos visuales se muestran correctamente, que la navegación es intuitiva y que las interacciones responden de manera adecuada.

Al combinar las pruebas unitarias y las pruebas visuales, se busca obtener un sistema robusto y confiable. Las pruebas unitarias garantizan la calidad del código subyacente, mientras que las pruebas visuales aseguran una experiencia de usuario fluida y satisfactoria. Ambos tipos de pruebas son fundamentales para detectar y corregir posibles errores antes de que la aplicación sea implementada y puesta en manos de los usuarios finales.

## 5.2. Tabla de verificación

### ► Pruebas Unitarias

Test de Apis	✓
Test de filtrado correcto	✓
Test de Token	✓
Test de Transmisión de binarios	✓
Test de almacenamiento BLOB	✓
Test de autenticación	✓
Test de encriptación de datos	✓
Test de rendimiento	✓
Test de manejo de errores	✓
Test de escalabilidad	✓

Tabla 5.2.1 – Pruebas Unitarias realizadas

### ► Pruebas Visuales

Función	
Api count de posts	✓
Api Tags	✓
Api Posts	✓
Api auth token	✓
Api User	✓
Pagina Home	✓
Pagina User Profile	✓
Pagina Admin Panel	✓
Página 404	✓
Página de créditos (No finalizada)	✗
Página Tags	✓
Pagina Posts	✓
Pagina Upload	✓

Tabla 5.2.2 – Pruebas Visuales

---

## Capítulo 6

### Conclusión

---

En la sección final de esta documentación, se presentarán las conclusiones que se han obtenido durante el desarrollo del Trabajo de Fin de Grado, así como las posibles tareas que se pueden llevar a cabo en el futuro para mejorar el sistema. Además, se proporcionará información sobre la cantidad de tiempo que se ha dedicado al proyecto y se realizará una evaluación personal del trabajo realizado.

#### 6.1.Resultados Obtenidos

Hasta la fecha, los resultados obtenidos en este proyecto han sido realmente prometedores. A pesar de enfrentarnos a limitaciones de tiempo y a la complejidad inherente del proyecto, hemos logrado alcanzar avances significativos en múltiples áreas.

Sin embargo, es importante destacar que debido a las restricciones de tiempo y a la complejidad que implica este proyecto, no hemos podido completar algunas secciones específicas. Estas secciones pendientes representan un desafío adicional que requerirá de una atención cuidadosa en el futuro.

Lo más destacado del proyecto en general es el sistema de binarios, que ha sido implementado de manera eficiente y optimizada para garantizar un rendimiento óptimo. Este sistema de binarios permite el procesamiento rápido y la manipulación eficiente de datos, lo cual resulta fundamental para el correcto funcionamiento de la aplicación.

Además, se ha puesto un gran énfasis en el cuidado del estilo en CSS. Se ha seguido una metodología de desarrollo limpio y modular, utilizando selectores bien estructurados y prácticas de nomenclatura coherentes. Esto no solo facilita la legibilidad y mantenibilidad del código, sino que también mejora la escalabilidad del proyecto, permitiendo la adición de nuevas características y la realización de cambios sin dificultades.

Otro aspecto destacado es el uso interno del back-end. Se ha desarrollado una arquitectura sólida y escalable, que permite manejar eficientemente las solicitudes del cliente y procesar los datos necesarios para brindar una experiencia fluida. Se han utilizado tecnologías y frameworks modernos, lo que ha permitido la implementación de funcionalidades avanzadas y la mejora del rendimiento en general.

En resumen, el proyecto se destaca por su sistema de binarios eficiente, el cuidado del estilo en CSS y el uso interno del back-end. Estas características han sido diseñadas y desarrolladas con el objetivo de lograr un proyecto de alta calidad, que ofrezca un rendimiento óptimo, una experiencia de usuario agradable y una base sólida para futuras expansiones y mejoras.

---

## 6.2. Análisis de trabajo

Tecnologías utilizadas:

- Spring Boot: Se utiliza como servidor Backend. Spring Boot es un framework de desarrollo de aplicaciones Java que proporciona un entorno simplificado para crear aplicaciones robustas y escalables.
- Angular: Se utiliza como servidor Frontend. Angular es un framework de desarrollo de aplicaciones web desarrollado en TypeScript y permite crear interfaces de usuario interactivas y dinámicas.
- MySQL: Se utiliza como base de datos para almacenar la información del programa. MySQL es un sistema de gestión de bases de datos relacional que ofrece un alto rendimiento y una amplia gama de funcionalidades.
- SQLite: Se utiliza como base de datos para almacenar los archivos binarios. SQLite es una base de datos ligera y de alto rendimiento que se integra fácilmente con aplicaciones y ofrece una buena organización y rendimiento en la búsqueda de los binarios.
- Bootstrap: Se utiliza como framework de CSS para el diseño y la presentación de la interfaz de usuario. Bootstrap facilita la creación de interfaces responsivas y atractivas.

---

#### Ventajas de las tecnologías seleccionadas:

- Spring Boot: Ofrece una configuración y puesta en marcha sencillas, así como una integración fluida con otras bibliotecas y frameworks de Spring. Proporciona un soporte sólido para el desarrollo de servicios web y la implementación de API RESTful.
- Angular: Permite crear una interfaz de usuario moderna y receptiva. Su estructura modular y la capacidad de carga veloz de vistas web son beneficiosos para este proyecto de tablón de imágenes.
- MySQL: Es un sistema de gestión de bases de datos ampliamente utilizado y confiable. Ofrece un alto rendimiento y una buena escalabilidad, lo que es importante para un programa que maneja una gran cantidad de datos.
- SQLite: Es una base de datos ligera y de fácil configuración, lo que facilita su integración en el proyecto. Proporciona un buen rendimiento en términos de búsqueda y organización de archivos binarios.
- Bootstrap: Facilita el diseño y la presentación de la interfaz de usuario. Proporciona un conjunto de estilos y componentes predefinidos que agilizan el desarrollo y aseguran una apariencia coherente en diferentes dispositivos y navegadores.

#### Funcionalidades principales del proyecto:

- Registro de usuarios: Permite a los usuarios crear una cuenta en el sistema para acceder a las funcionalidades del tablón de imágenes.
- Visualización de publicaciones: Los usuarios pueden ver las publicaciones existentes en el tablón de imágenes.
- Subida de publicaciones: Los usuarios pueden cargar imágenes y agregar información relevante para crear nuevas publicaciones en el tablón.
- Comentarios en publicaciones: Los usuarios pueden comentar las publicaciones existentes, lo que fomenta la interacción y la participación de la comunidad.
- Etiquetas en publicaciones: Los usuarios pueden agregar etiquetas a las publicaciones para clasificarlas y facilitar su búsqueda y organización.

### 6.3.Mejoras Posibles

Posibilidades	Inconvenientes
Una gran inflexión en el proyecto seria desarrollar su par de frontend en Android y iOS de tal forma que también hubiera soporte para dichas plataformas.	Sería posible de más tiempo de desarrollo.
Sería conveniente montar la aplicación en Docker para mayor facilidad al montarla.	Debido a la gran complejidad del programa sería necesario hacer un estudio sobre las variables conjuntas, esto sería posible realizarlo más adelante.
Lanzador para Linux	Debido a la falta de tiempo no fue posible hacer el lanzador para servidores Linux
Panel de Usuarios y administración	En posteriores versiones se tiene planeado finalizar esta parte
Mejora del sistema y regeneración de tokens	Debido a la complejidad y la falta de tiempo no ha sido posible.

Tabla 6.3.1 – Mejoras posibles

---

## 6.4.Mercado

### Comercialización:

En cuanto al mercado, la comercialización es un aspecto importante a considerar. Aunque el objetivo principal del proyecto no sea la venta ni la comercialización en sí misma, es relevante evaluar su rentabilidad en caso de ser llevado al mercado. Una estrategia que se podría implementar para lograr esto es mediante el uso de anuncios (ADS) en diferentes páginas.

La inclusión de anuncios publicitarios dentro del proyecto permitiría generar ingresos y, por ende, hacerlo rentable. Estos anuncios podrían ser cuidadosamente seleccionados y colocados estratégicamente en las distintas secciones del proyecto, de manera que no afecten negativamente la experiencia del usuario, pero sí brinden una fuente adicional de ingresos.

Es importante destacar que la elección de las páginas donde se mostrarían los anuncios debe ser realizada con un enfoque adecuado, seleccionando sitios web relevantes y afines al tema del proyecto. Esto permitiría llegar a un público objetivo más amplio y aumentar las posibilidades de generar interés y clics en los anuncios.

Además, es recomendable utilizar herramientas de análisis y seguimiento para evaluar el rendimiento de los anuncios y realizar ajustes en caso necesario. Esto permitirá maximizar los resultados y optimizar la rentabilidad del proyecto en caso de ser comercializado.

Es fundamental tener en cuenta que la implementación de anuncios publicitarios debe ser cuidadosa y equilibrada, evitando saturar al usuario con excesiva publicidad. El objetivo principal del proyecto debe seguir siendo brindar un valor significativo a los usuarios, y los anuncios deben ser considerados como una forma complementaria de monetización sin comprometer la calidad y la usabilidad del proyecto en sí.

En resumen, aunque la venta y comercialización no sean el objetivo principal del proyecto, la inclusión de anuncios publicitarios puede ser una estrategia viable para hacerlo rentable en caso de ser comercializado. Mediante la selección adecuada de páginas y el seguimiento del rendimiento de los anuncios, se puede optimizar la rentabilidad sin comprometer la calidad del proyecto ni la experiencia del usuario.



---

### **Público Objetivo:**

Además de las comunidades artísticas o fan arts, el mercado objetivo de esta aplicación se extiende a otros grupos relacionados con el arte y la creatividad. Esto incluye a los diseñadores gráficos, ilustradores, fotógrafos y cualquier persona apasionada por la expresión visual.

Estos grupos suelen encontrar en los tableros de imágenes una forma efectiva de compartir sus creaciones, recibir retroalimentación constructiva y establecer conexiones con otros artistas y entusiastas del arte. La aplicación proporciona un espacio virtual donde pueden exhibir y promocionar su trabajo de manera accesible y atractiva.

Además, el público objetivo de esta aplicación puede abarcar a aquellos que buscan inspiración y entretenimiento visual. Los amantes del arte en general, los aficionados a la moda, los entusiastas del diseño de interiores y las personas interesadas en descubrir nuevas tendencias y estilos también pueden beneficiarse de esta plataforma.

En resumen, el mercado objetivo de esta aplicación abarca a las comunidades artísticas y fan arts, así como a diseñadores, ilustradores, fotógrafos y entusiastas del arte en general. También atrae a personas en busca de inspiración y entretenimiento visual, así como a aquellos interesados en descubrir nuevas tendencias y formas de expresión artística en diversos medios visuales. La aplicación proporciona un espacio interactivo donde todos estos grupos pueden conectarse, compartir y disfrutar del arte en todas sus formas.

## 6.5. Tiempo estimado

<i>Sección</i>	<b>Tiempo estimado</b>
Memoria	38h
Lanzador Grafico	3h
Lanzador de consola	30m
Sistema de auto configuración	2,09h
FrontEnd	32h
BackEnd	28h

(El tiempo se ha calculado de forma aproximada a base de cronometraje del tiempo)

Tabla 6.5.1 – Tiempo estimado

---

## 6.6. Valoración Personal

En el desarrollo de este proyecto he ido observando diversas dificultades así como formas de lograr el objetivo de una forma diferente, ya que el objetivo de este proyecto además de la creación de dicha web con diversas tecnologías más actuales, era el reto de aprender dichas tecnologías y usarlas en conjunto para una gran satisfacción.

En conjunto el proyecto en sí es complejo ya que usamos lenguajes como Java, C#, .bat, Js, Ts, Sql de los cuales algunos se han tenido que aprender desde cero. A parte del aprendizaje de la conexión de diferentes frameworks como SpringBoot o Angular y la creación de ventanas de usuario con Forms.

Además de lo anterior, uno de los desafíos más complejos fue la transmisión de datos binarios entre el backend y el frontend de la aplicación. Para lograrlo, fue necesario implementar un proceso en el que los binarios se convirtieran en formato base64 al ser cargados, luego se enviarían al backend y se formatearían como un tipo de dato BLOB (Binary Large Object), el cual se almacenaba en una base de datos local dinámica por usuario.

Esta tarea requería una cuidadosa manipulación de los datos, ya que la conversión de binarios a base64 y viceversa implicaba transformaciones específicas. En el lado del backend, se debían realizar las operaciones inversas para recuperar los datos binarios originales y garantizar la integridad de la información.

Además de la complejidad técnica de la transmisión y almacenamiento de los binarios, también se enfrentaban desafíos adicionales en la obtención de los datos. El proceso de obtención de datos debía tener en cuenta la estructura de la base de datos local dinámica, que almacenaba información de forma personalizada para cada usuario. Esto requería consultas y filtrado adecuados para obtener los datos relevantes de manera eficiente.

En resumen, la transmisión de binarios entre el backend y el frontend, junto con la manipulación y almacenamiento adecuados de dichos binarios, representó una de las tareas más complicadas en el desarrollo de la aplicación. Requerió una cuidadosa planificación y ejecución para garantizar que los datos se transmitieran correctamente y estuvieran disponibles para su recuperación posterior.

---

## Capítulo 7

### Fuentes de Información

---

#### 7.1. Bibliografía

*Angular Docs.* (s.f.). Obtenido de <https://angular.io/docs>

*DB Browser (SQLite).* (s.f.). Obtenido de <https://sqlitebrowser.org/>

*SQLite Documentation.* (s.f.). Obtenido de <https://www.sqlite.org/docs.html>

*Windows Forms Docs.* (s.f.). Obtenido de <https://learn.microsoft.com/es-es/visualstudio/ide/create-csharp-winform-visual-studio>

*YT latecnologiaavanza.* (s.f.). Obtenido de <https://www.youtube.com/@latecnologiaavanza>

#### 7.2. Recursos

*Counter General Booru.* (s.f.). Obtenido de <https://safebooru.org/counter/{0-9}.gif>

*Icons Bootstrap.* (s.f.). Obtenido de <https://icons.getbootstrap.com/>

#### 7.3. Documentación

*SQLite Documentation.* (s.f.). Obtenido de <https://www.sqlite.org/docs.html>

*Angular Docs.* (s.f.). Obtenido de <https://angular.io/docs>

*Bootstrap Docs.* (s.f.). Obtenido de <https://getbootstrap.com/docs/5.2/getting-started/introduction/>

*SpringBoot Docs.* (s.f.). Obtenido de <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

## Anexos

### 1 - Índice de terminología

Palabra	Definición
<b>Post</b>	Campo con contenido multimedia de tipo imagen (binario) la cual está compuesta por su meta data, nombre, fecha de subida además de sus diferentes tags identificatorios
<b>Binario</b>	Archivo compuesto por un conjunto de Bytes pueden ser cualquier tipo de archivo de no texto
<b>Tag</b>	Etiqueta que identifica un post. Ej.:( Una imagen que aparece una chica se le añadirá la Tag “girl” )
<b>Backend</b>	Servidor que actúa no de forma frontal al usuario
<b>Frontend</b>	Servidor que actúa de forma frontal al usuario
<b>S-A</b>	SpringBoot ↔ Angular
<b>Api</b>	Servicio web que provee datos
<b>Ads</b>	Anuncios
<b>Java</b>	Java es un lenguaje de programación orientado a objetos que se utiliza para desarrollar aplicaciones y programas de software.
<b>C#</b>	C# (pronunciado "C sharp") es un lenguaje de programación orientado a objetos creado por Microsoft.
<b>Js</b>	JS es una abreviatura comúnmente utilizada para referirse a JavaScript, un lenguaje de programación interpretado utilizado principalmente para agregar interactividad y dinamismo a las páginas web.
<b>Ts</b>	TS es una abreviatura comúnmente utilizada para referirse a TypeScript, un lenguaje de programación de código abierto desarrollado por Microsoft. TypeScript es un superset de JavaScript que agrega características adicionales.
<b>.bat</b>	.bat es una extensión de archivo utilizada para archivos de procesamiento por lotes en sistemas operativos de Windows. Un archivo .bat contiene un conjunto de comandos que se ejecutan secuencialmente cuando se inicia el archivo.
<b>Forms</b>	Forms es un sistema de creación de aplicaciones graficas desarrollado por Microsoft, se puede programar tanto en C# o en vb
<b>SpringBoot</b>	Spring Boot es un marco de desarrollo de aplicaciones de código abierto para Java que simplifica el proceso de configuración y creación de aplicaciones basadas en Spring Framework.
<b>Thymeleaf</b>	Thymeleaf es un motor de plantillas de código abierto para Java que se utiliza para crear y renderizar vistas en aplicaciones web.
<b>Angular</b>	Angular es un framework de desarrollo de aplicaciones web de código abierto y basado en JavaScript. Fue desarrollado por Google y se utiliza para crear aplicaciones web de una sola página y aplicaciones móviles híbridas.
<b>NodeJs</b>	Node.js es un entorno de tiempo de ejecución de código abierto basado en JavaScript que permite a los desarrolladores ejecutar código de JavaScript fuera del navegador web.
<b>MIME</b>	MIME (Multipurpose Internet Mail Extensions) es un estándar de Internet que define el formato de los mensajes de correo electrónico y otros contenidos transmitidos a través de la red. MIME se utiliza para describir el tipo de contenido de un archivo o mensaje, y para especificar cómo se deben codificar y decodificar los datos para su transmisión. Resumidamente serian archivos multimedia

<b>BLOB</b>	Los BLOB (Binary Large Objects) son elementos fundamentales en las bases de datos que permiten el almacenamiento de datos de gran tamaño que cambian de forma dinámica.
<b>MVC</b>	MVC es un patrón de arquitectura de software que se utiliza para separar la lógica de presentación de una aplicación web de la lógica de negocio y los datos subyacentes. La abreviatura MVC significa Modelo-Vista-Controlador.
<b>JPA</b>	JPA son las siglas de Java Persistence API, una especificación de Java que define un conjunto de interfaces y clases que se utilizan para persistir objetos Java en una base de datos relacional.
<b>CRUD</b>	CRUD es un acrónimo que se refiere a las operaciones básicas que se pueden realizar en una base de datos o sistema de almacenamiento de datos. CRUD significa Create (crear), Read (leer), Update (actualizar) y Delete (eliminar).
<b>JUnit</b>	JUnit es un framework de pruebas unitarias de código abierto para el lenguaje de programación Java.

Tabla A.1 – Índice de terminología

---

## **2 - Ventanas**

### **2.2 - Paginas ocultas**

- Créditos, dicha ventana muestra la información sobre la creación del programa web a la cual solo se puede acceder desde en la página 404.

## NOTAS DE BORRADOR – SECCION A ELIMINAR EN DOCUMENTO FINAL

**NOTA** Este documento puede sufrir modificaciones

**NOTA** Todo lo marcado en amarillo es temporal y modificable

**NOTA** Hasta la emisión del documento se mostrará ( ¿.?.? ) en las indicaciones contadoras como figuras y tablas.

**NOTA** Hasta la creación de una imagen en caso de no estar disponible debido a la tardía del



desarrollo se usará temporalmente la siguiente de forma temporal

**NOTA** Se han usado indicadores referenciales para mayor comprensión de las figuras

**NOTA** El documento esta incompleto debido a la falta de realización de parte del proyecto.

Notas para clara:

¿Qué Falta?

Usar marcadores para el índice ( <a href="#">LINK</a> ) (Se realizará a la emisión final del documento)	✗
Usar Formato APA para bibliografía	✓
Rehacer imágenes contadores	✗

## WEBS CITADAS

<https://safebooru.org/>

Aplicaciones a usar:

(Creador gifs animados de estadística) <https://datagifmaker.withgoogle.com/>