

Clara Bujeda Muñoz

Tutora: Rosana Marín Berraondo

Centro: Salesianos Nuestra Señora del Pilar

Grado superior en aplicaciones multiplataforma

NinEIMGBOARD

Borrador Memoria TFG

**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, asi 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 ¿

1.1.Contexto¿

1.2.Objetivos¿

1.3.Motivación ¿

1.4.Tecnologías usadas¿

1.5.Organización de la memoria¿

2. Análisis del sistema ¿

2.1.Sistema Inicial¿

2.2.Requisitos mínimos¿

2.3. Identificación de los actores¿

3. Arquitectura del sistema¿

3.1.Arquitectura Global¿

3.2.Diagrama estructural¿

3.2.1.General¿

3.2.2.Backend¿

3.2.3.Frontend¿

3.2.4.BBDD¿

4. Diseño e implementación del sistema ¿

4.1.Datos y Estructura¿

4.2.Modelo lógico¿

4.3.Interfaces de usuario¿

5. Pruebas de funcionamiento¿

5.1.Explicación¿

5.2.Tabla de verificación¿

6. Conclusión¿

6.1.Resultados Obtenidos¿

6.2.Análisis de trabajo¿

6.3.Mejoras Posibles¿

6.4.Mercado¿

6.5.Tiempo estimado¿

6.6.Valoración Personal¿

7. Fuentes de Información¿

7.1.Bibliografía¿

7.2.Recursos¿

7.3.Documentación¿

8. Anexos¿

8.1. Código fuente y programas realizados¿

8.1. Justificación de las pruebas de funcionamiento¿

Índice de Figuras

--

Índice de Tablas

--

**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 asi 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 asi 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 ¿.¿.¿ - 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 capitulo 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 ideo 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 ¿?

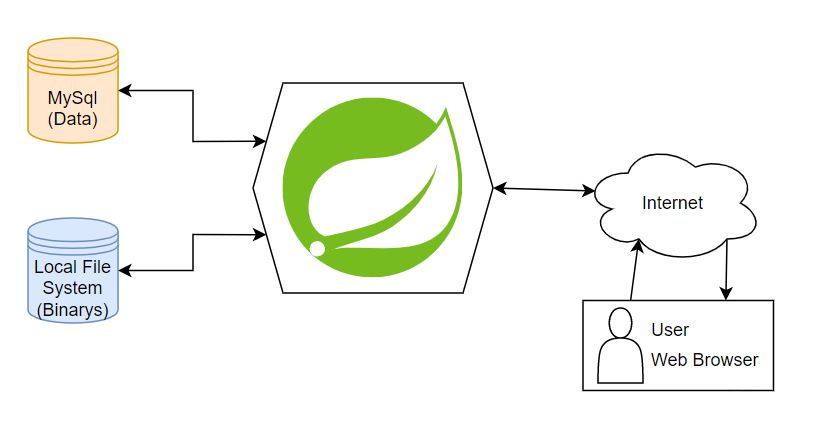
****

Figura ¿.?.? 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 figura ¿?

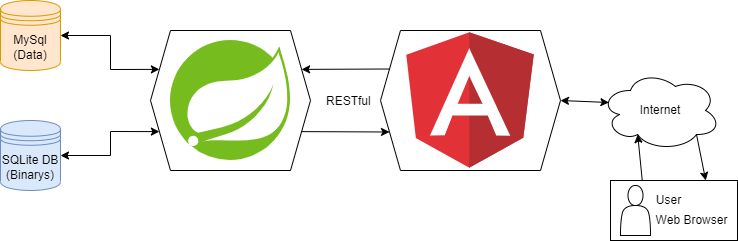


Figura ¿.?.? Diagrama ideático usado para el proyecto final

**2.2. Requisitos mínimos**

En este apartado se especificará la versión mínima de los diferentes programas y diferentes requerimientos que necesitará su servidor o computadora para la correcta ejecución de la aplicación web.

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

**2.3. Identificación de los actores**

|  |
| --- |
| Usuario |
| Un usuario en la aplicación podrá registrarse o no, en las siguientes tablas se muestra la identificación de los actores dependiendo de dichas causas.   |  |  | | --- | --- | | No registrado |  | | |  |  | | --- | --- | | ¿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. | | | |   Tabla ¿.?.? – Identificación de los actores según un usuario no registrado   |  |  | | --- | --- | | 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 asi como personalizar sus datos.   Upload   * En dicha ventana un usuario registrado podrá ser capaz de crear y subir un nuevo post a la aplicación web.   Tags   * En dicha ventana se le mostrará al usuario los tags disponibles para ejercer la búsqueda. | | | |   Tabla ¿.?.? – Identificación de los actores según un usuario registrado |

Tabla ¿.?.? – Identificación de los actores dependiendo de los usuarios.

|  |
| --- |
| Administrador |
| |  |  | | --- | --- | | ¿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 asi 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. | |   Tabla ¿.?.? – Identificación de los actores según un administrador |

Tabla ¿.?.? – Identificación de los actores dependiendo del Administrador.

**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 requiere 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.

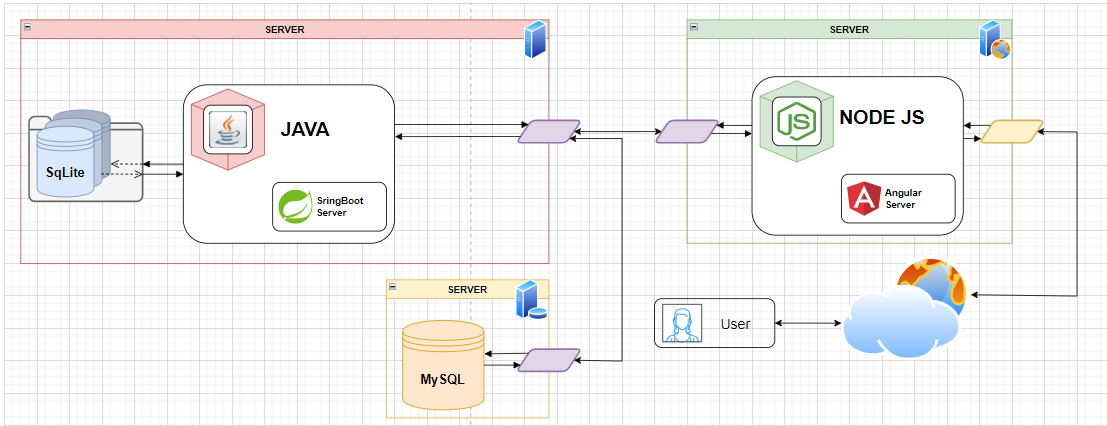


Figura ¿.?.? Arquitectura global del sistema

**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

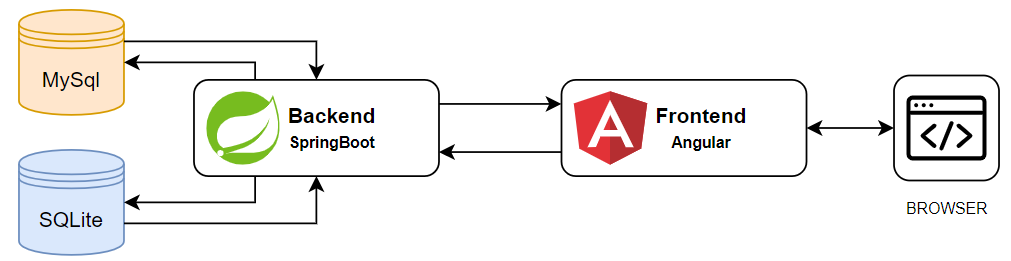
****

Figura ¿.?.? – 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 ultimo la dao la cual usa JPA (CRUD) para comunicarse con las diferentes bases de datos usadas.

****

Figura ¿.?.? 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 ¿.?.? Diagrama de obtención de datos y muestreo.

**3.2.4. BBDD**

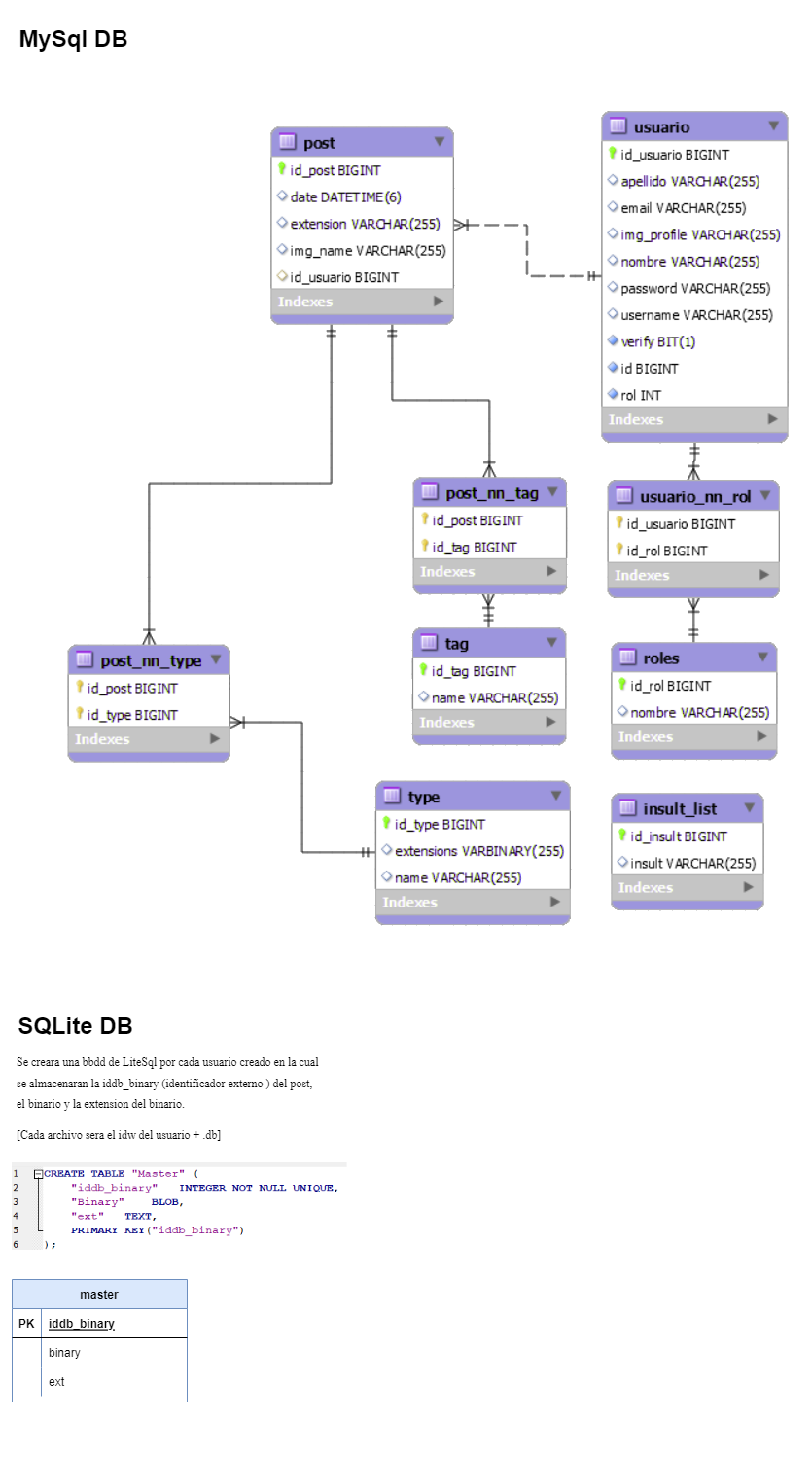
****

Figura ¿.?.? Diagrama de la base de datos

NOTA: El diagrama puede sufrir modificaciones posteriores (Diagrama ideático/ En futuras versiones se modificará por un diagrama ER)

**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**

**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 figura ¿.?.? se muestra dicha ventana.

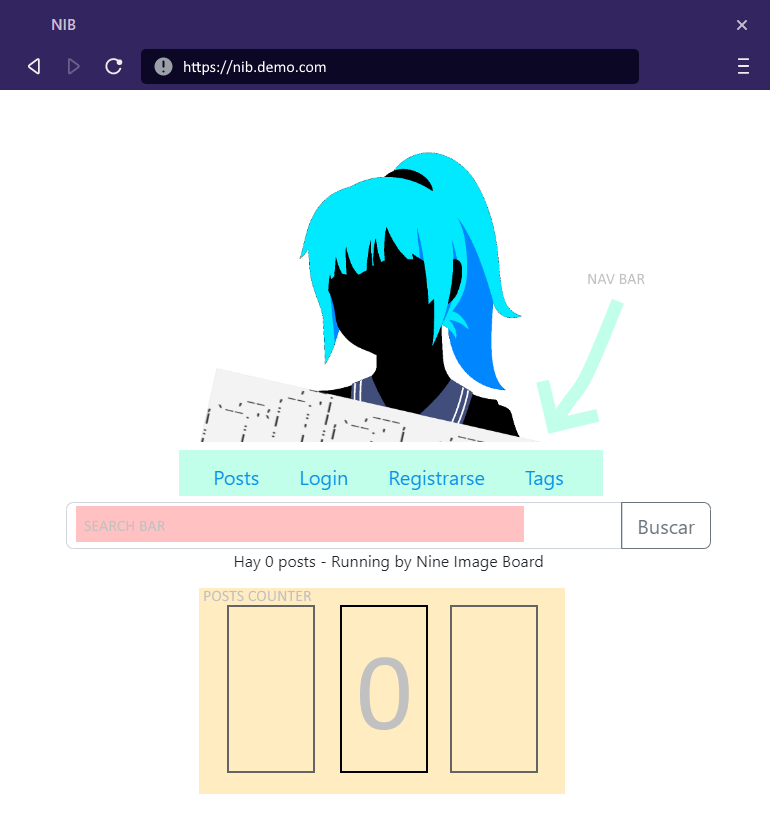


Figura ¿.?.? Plantilla estructural de la pantalla home.

**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 ¿.?.? Plantilla estructural 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 ¿.?.? Plantilla estructural 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 ¿.?.? Plantilla estructural 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.

****

Figura ¿.?.? 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.

****

Figura ¿.?.? 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.

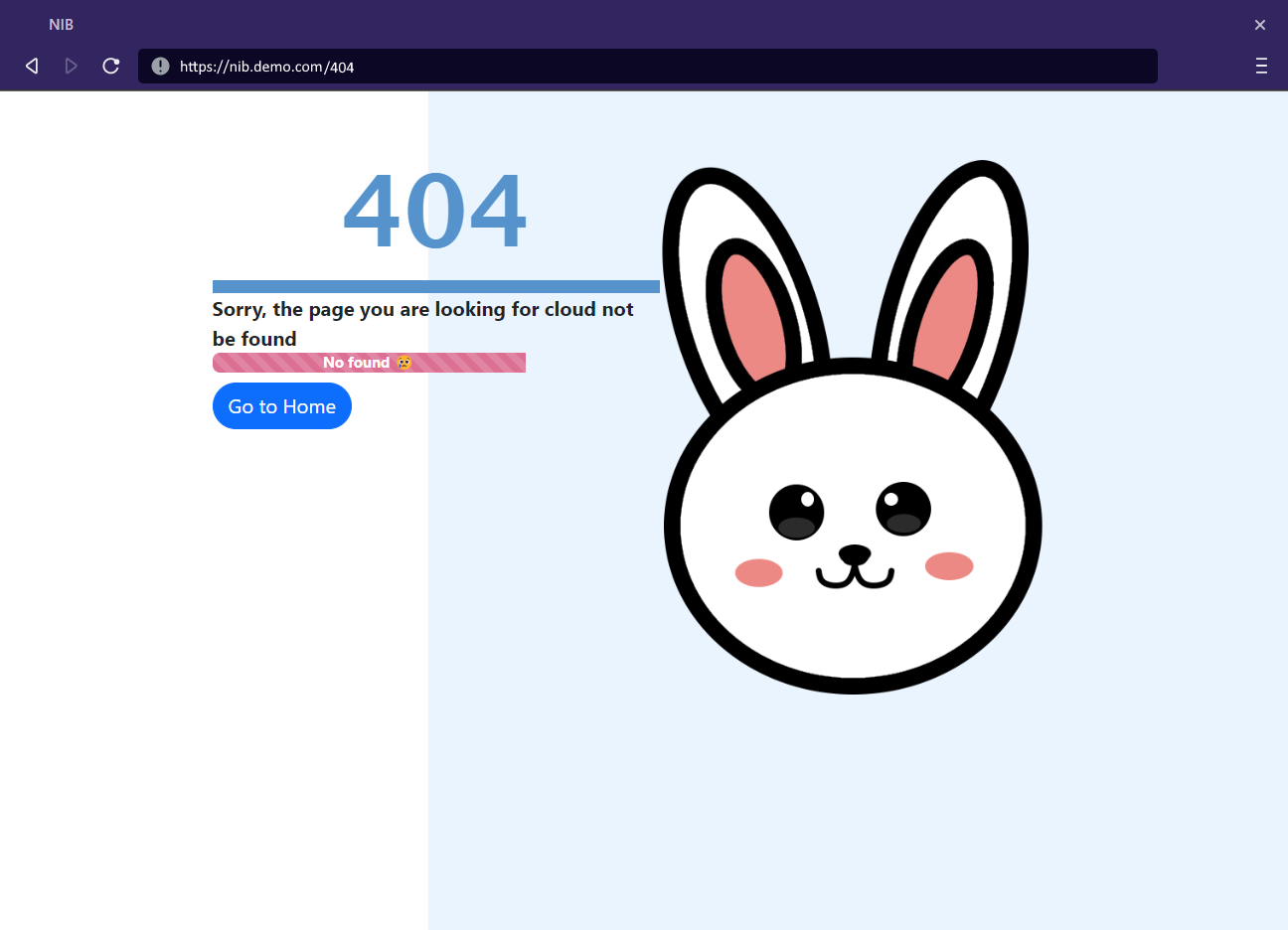
****

Figura ¿.?.? 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.

****

Figura ¿.?.? 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.

****

Figura ¿.?.? 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 pagina web asi 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 asi como configuraciones internas del sistema.

****

Figura ¿.?.? 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 ¿.?.? Panel de usuarios

**Capítulo 5**

**Pruebas de funcionamiento**

**5.1. Explicación**

Para las pruebas de funcionamiento se realizarán dos tipos de pruebas. Pruebas unitarias con ng Test además de JUnit y pruebas visuales testeadas al desarrollar la aplicación

**5.2. Tabla de verificación**

►Pruebas Unitarias

►Pruebas Visuales

|  |  |  |  |
| --- | --- | --- | --- |
| **Función** | **Estado** | **Progreso** |  |
|  |  |  |  |
| Muestreo de posts | **Pendiente** | **50%** | ✖️ |
| Muestreo de tags | **Pendiente** | **80%** | ✖️ |
| Animaciones del frontend | **Pendiente** | **70%** | ✖️ |
|  |  |  |  |
| Api cantidad de posts | **HECHO** | **100%** | ✔️ |
| Api Tags | **HECHO** | **100%** | ✔️ |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Pagina Home | **HECHO** | **100%** | ✔️ |
| Página 404 | **HECHO** | **100%** | ✔️ |
| Página de créditos | **Pendiente** | **50%** | ✖️ |
|  |  |  |  |
|  |  |  |  |

**Capítulo 6**

**Conclusión**

**6.1.Resultados Obtenidos**

(No realizable hasta finalización del proyecto)

**6.2.Análisis de trabajo**

**6.3.Mejoras Posibles**

|  |  |
| --- | --- |
| Posibilidades | Inconvenientes |
| Se podría desarrollar una aplicación en React y establecer una comunicación con el servidor SpringBoot | Sería posible de más tiempo de desarrollo |
|  |  |

**6.4.Mercado**

**Comercialización**

* El objetivo principal del proyecto no es la venta del proyecto ni su comercialización, pero la rentabilidad de este en caso de ser comercializado constaría en el uso de anuncios (ADS) entre las diferentes páginas lo cual permitiría hacer el proyecto rentable.

**Público Objetivo:**

* Las diferentes comunidades interesadas en este tipo de aplicaciones son las artísticas o fan arts. Estos grupos suelen recurrir a tableros de imágenes para publicar sus obras, así como sus representaciones o fotos.

**6.5.Tiempo estimado**

|  |  |
| --- | --- |
| Sección | Tiempo estimado |
| Memoria | 28h |
| Lanzador Grafico | 3h |
| Lanzador de consola | 30m |
| Sistema de auto configuración | 2h |
| FrontEnd | 25h |
| BackEnd | 10h |

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

**6.6.Valoración Personal**

En el desarrollo de este proyecto he ido observando diversas dificultades asi 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 si 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.

//Mencionar seguridad y dificultades

**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**

**7.3.Documentación**

**Anexos**

**1- Índice de terminología**

|  |  |
| --- | --- |
| Palabra | Definición |
| Post | 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 |
| Binario | Archivo compuesto por un conjunto de Bytes pueden ser cualquier tipo de archivo de no texto |
| Tag | Etiqueta que identifica un post. Ej.:( Una imagen que aparece una chica se le añadirá la Tag “girl” ) |
| Backend | Servidor que actúa no de forma frontal al usuario |
| Frontend | Servidor que actúa de forma frontal al usuario |
| S-A | SpringBoot ↔ Angular |
| Api | Servicio web que provee datos |
| Ads | Anuncios |
| Java | Java es un lenguaje de programación orientado a objetos que se utiliza para desarrollar aplicaciones y programas de software. |
| C# | C# (pronunciado "C sharp") es un lenguaje de programación orientado a objetos creado por Microsoft. |
| Js | 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. |
| Ts | 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. |
| .bat | .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. |
| Forms | Forms es un sistema de creación de aplicaciones graficas desarrollado por Microsoft, se puede programar tanto en C# o en vb |
| SpringBoot | 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. |
| Thymeleaf | Thymeleaf es un motor de plantillas de código abierto para Java que se utiliza para crear y renderizar vistas en aplicaciones web. |
| Angular | 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. |
| NodeJs | 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. |
| MIME | 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 |
| BLOB | 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. |
| MVC | 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. |
| JPA | 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. |
| CRUD | 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). |
| JUnit | JUnit es un framework de pruebas unitarias de código abierto para el lenguaje de programación Java. |

**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 ([LINK](https://support.microsoft.com/es-es/office/agregar-o-eliminar-marcadores-en-un-documento-de-word-o-mensaje-de-outlook-f68d781f-0150-4583-a90e-a4009d99c2a0)) (Se realizará a la emisión final del documento) | ✖️ |
| Usar Formato APA para bibliografía | ✔️ |
| Rehacer imágenes contadores | ✖️ |

|  |
| --- |
| WEBS CITADAS |
| https://safebooru.org/ |
|  |
|  |

Aplicaciones a usar:

<https://datagifmaker.withgoogle.com/>