

SPRING ACME – RED SOCIAL

TRAINER

CARLOS H. RUEDA C.

INGENIERO DE SISTEMAS

ALVARO ANDRES MARTINEZ ALCINA

PROYECTO SKILLS – SPRING BOOT

CAMPUSLANDS

2025

RESUMEN DEL PROYECTO

Este proyecto consiste en una aplicación web responsiva tipo red social, donde los usuarios pueden interactuar con otras personas mediante la creación de publicaciones, la publicaciones se crean con texto e imágenes (a través de enlaces); reacciones (likes), comentarios, y gestión de los usuarios a través de la función de "seguir" y "dejar de seguir". Además, el sistema permite la edición de los datos del usuario y el envío de notificaciones para mantener la interacción en tiempo real.

REQUERIMIENTOS FUNCIONALES

1. El usuario puede Inicio de sesión y registrarse en la aplicación.
2. EL usuario puede hacer publicaciones (crear, editar, eliminar y ordenar).
3. El usuario puede hacer interacciones (comentarios, likes y etiquetado de usuarios).
4. Realizar notificaciones y realizar un sistema de seguimiento de usuarios.
5. EL usuario debe registrar su fecha de nacimiento (se validará que el usuario tenga al menos 14 años para registrarse).
6. El usuario debe crear una contraseña segura (8-12 caracteres con al menos 1 mayúscula, 1 minúscula, 1 número y 1 símbolo especial %\$;&.,#).
7. El usuario registra sus imágenes a través de enlace URL tanto para su imagen de perfil y publicación.
8. La descripción, el contenido de la publicación debe tener mínimo 5 caracteres, máximo 500 y debe ser obligatorio.
9. La Imagen de la publicación es opcional, pero debe guardar enlace URL.
10. Las publicaciones deben tener Tags personalizables (#Ejemplo).
11. La edición y eliminación de publicaciones solo por el autor.
12. EL ordenamiento de las publicaciones, permitiendo que el usuario pueda elegir entre:

- Orden cronológico (ascendente o descendente).
 - Orden por relevancia (según interacciones).
13. Los usuarios podrán comentar publicaciones (máximo 200 caracteres por comentario).
14. Se podrán etiquetar otros usuarios con @<nombre_usuario>.
15. Las publicaciones deben tener "Me gusta" (un usuario solo puede dar un "Me gusta" a cada publicación, y puede quitarlo).
16. Resumen de interacciones en cada publicación, mostrando:
- Número total de reacciones.
 - Número total de comentarios.
17. Un usuario podrá seguir y dejar de seguir a otros.
18. En la sección de inicio, solo se mostrarán publicaciones de usuarios seguidos.
19. Se podrán ver las listas de:
- Seguidores.
 - Seguidos.
20. EL usuario podrá recibir notificaciones a través de:
- Reciban un comentario en su publicación.
 - Reciban un "Me gusta" en su publicación.
 - Sean etiquetados en una publicación.
 - Sean seguidos por otro usuario.

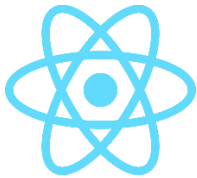
REQUERIMIENTOS NO FUNCIONALES

1. Desarrollo de la aplicación debe ser full stack utilizando Spring Boot para el backend, MySQL o PostgreSQL para la base de datos y ReactJS para el frontend.
2. Inicio y cierre de sesión seguro mediante autenticación con JWT.
3. Cifrado de contraseñas con BCrypt para mayor seguridad.
4. Implementación de Spring Security con JWT, Spring Data JPA con Hibernate, Arquitectura modular con DTOs, Services, Entities, etc.
5. Modo oscuro y claro con persistencia de estado.
6. Diseño responsivo optimizado para móviles y escritorio.

HERRAMIENTAS IMPLEMENTACIÓN

FRONTEND

ReactJS: Biblioteca de JavaScript para construir la interfaz de usuario.



Vite: Herramienta de construcción rápida que mejora la experiencia de desarrollo.

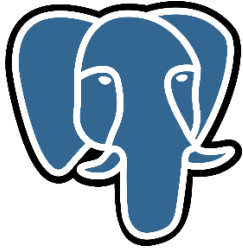


TailwindCSS: Para el diseño responsivo.



BASE DE DATOS

PostgreSQL: Base de datos seleccionada para el desarrollo de la aplicación.



BACKEND

Spring Boot: Framework basado en Java para crear aplicaciones web y servicios RESTful de manera sencilla.



OTRAS HERRAMIENTAS

JWT (JSON Web Tokens): Para la autenticación y autorización del usuario.



Fetch: Para hacer peticiones HTTP desde el frontend.



Spring Security: Para el manejo de la seguridad y generación de permisos.

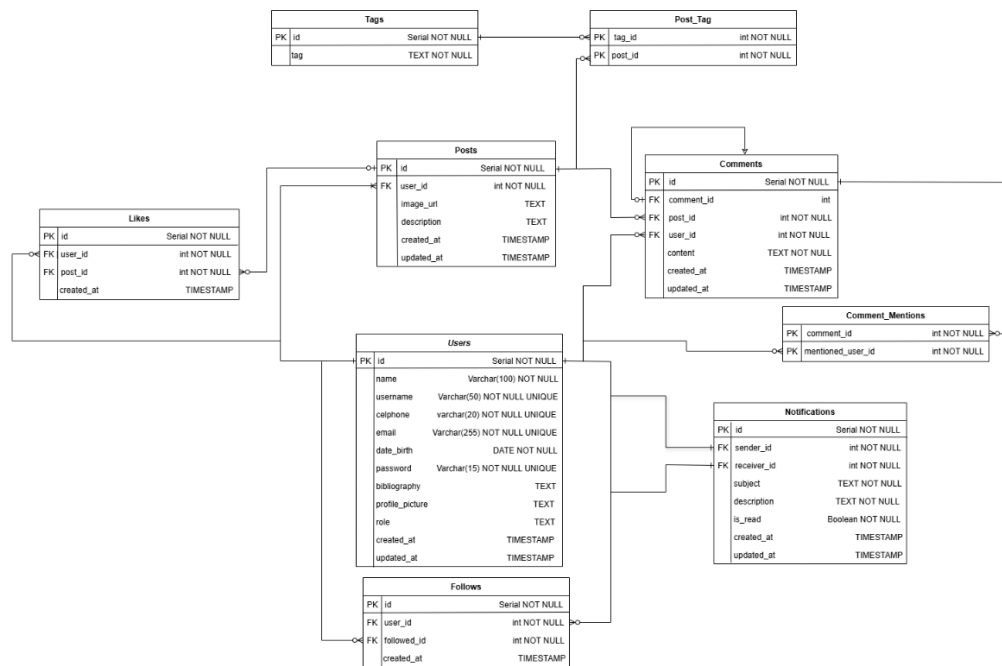


Postman: Permite realizar pruebas API, lo que da la posibilidad de testear 'HTTP REQUEST a través de la plataforma, obteniendo diferentes tipos de respuesta que posteriormente deberán ser validados.

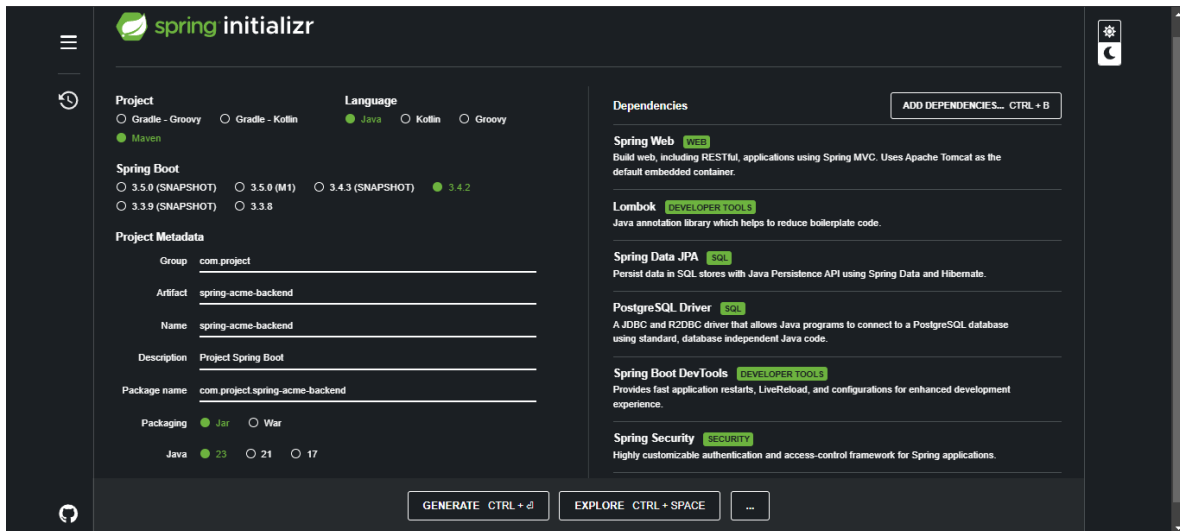


DESARROLLO DEL PROYECTO

En el desarrollo de la aplicación web responsiva tipo Red Social, se comenzó con la creación del Modelo Relacional, definiendo las tablas necesarias para el almacenamiento y registro de los datos.



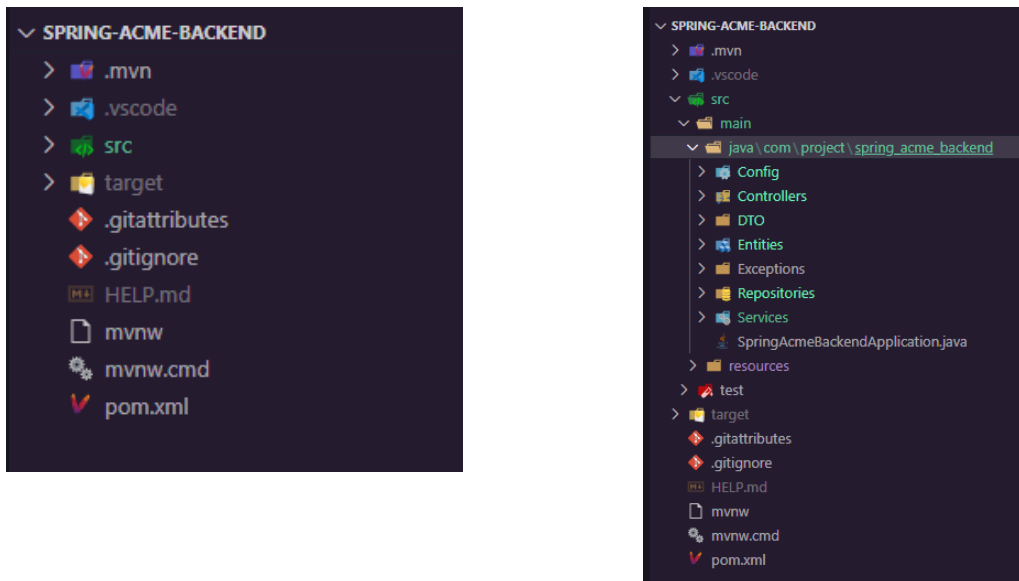
Para el desarrollo del backend, se creó el proyecto en Spring Boot a través de la página <https://start.spring.io/>, configurando los parámetros necesarios. Luego, solo fue necesario abrir el proyecto en Visual Studio Code para comenzar con el desarrollo.



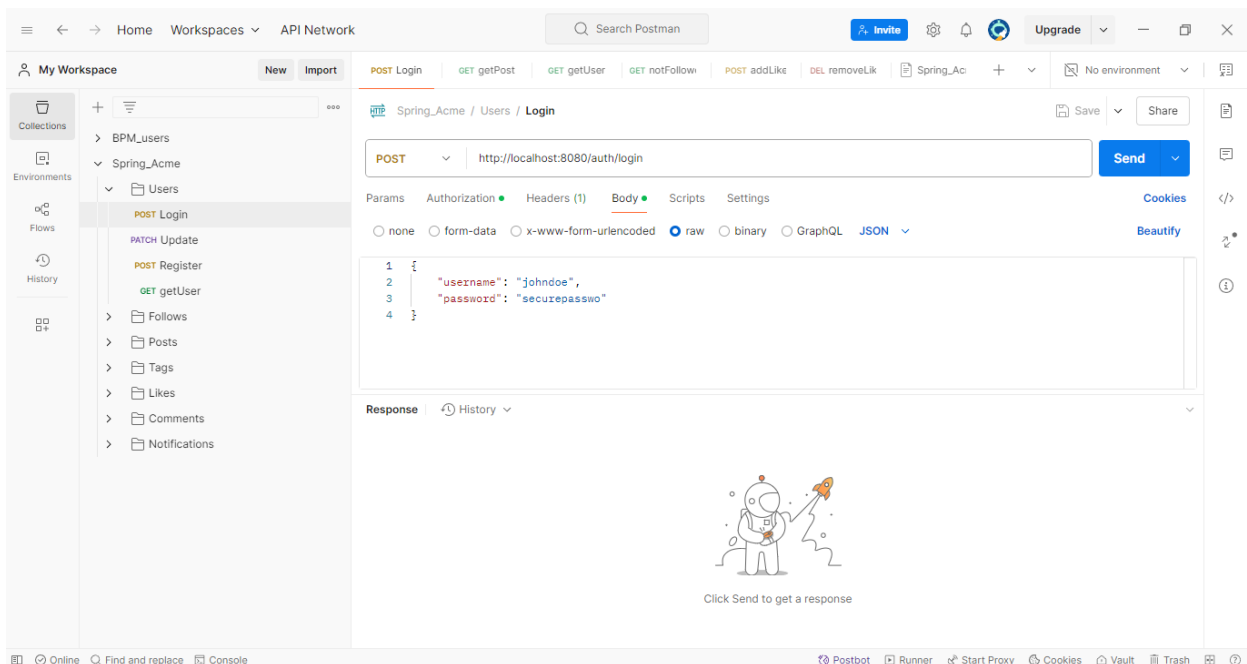
Se insertó las dependencias faltantes para el desarrollo del proyecto. Las dependencias que se establecieron eran parte de JWT.

```
<dependencies>Add Spring Boot Starters...
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-api</artifactId>
    <version>0.11.5</version>
  </dependency>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-impl</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-jackson</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <optional>true</optional>
  </dependency>
</dependencies>
```

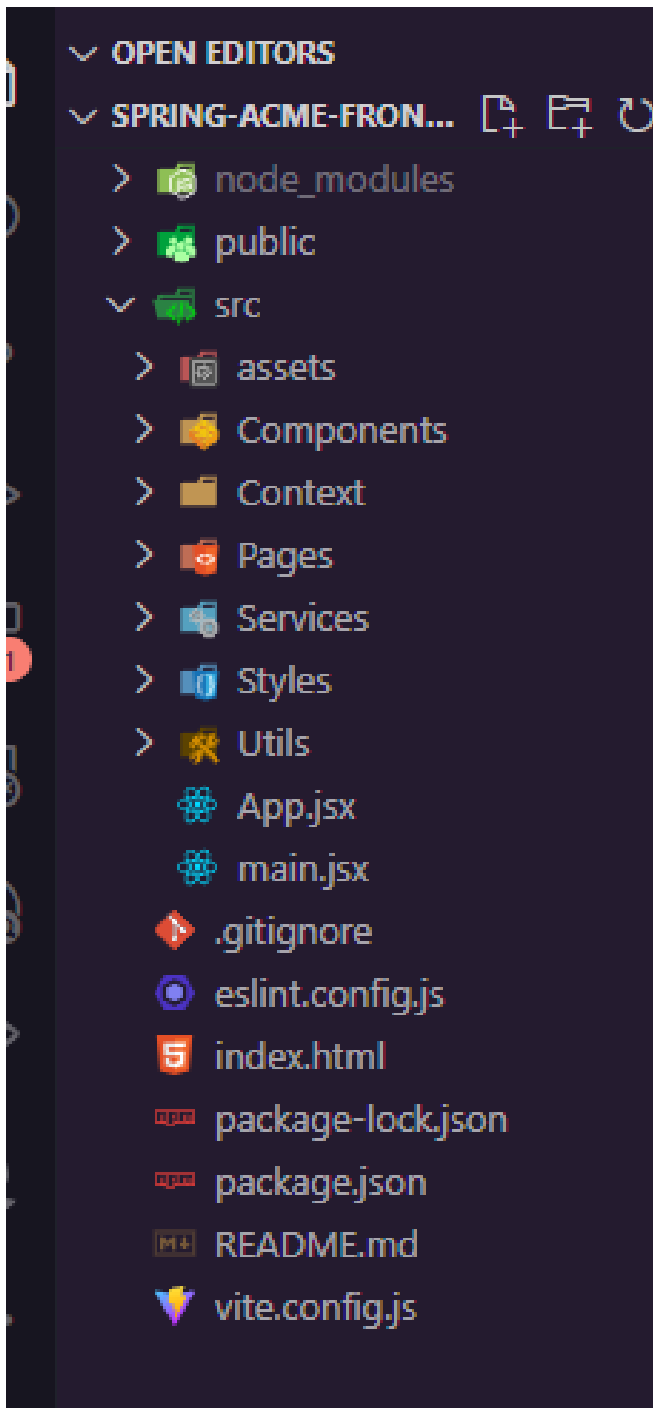
Para el desarrollo del Software en el enfoque del Backend se estableció Estructuración de carpetas para el mejor manejo de los archivos del Backend.



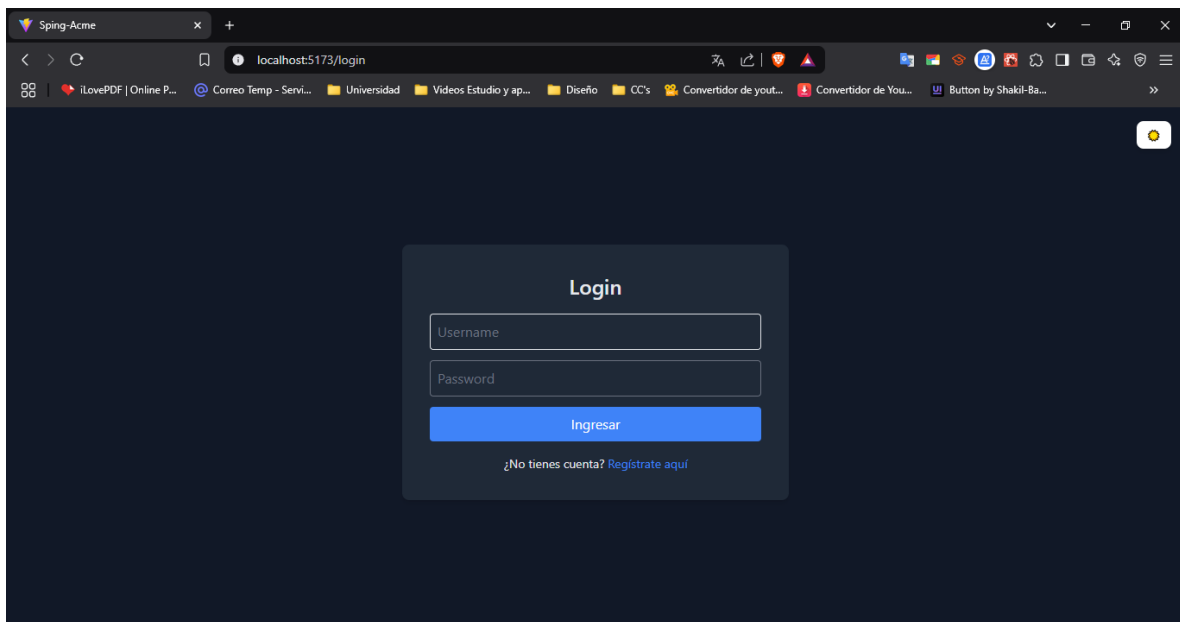
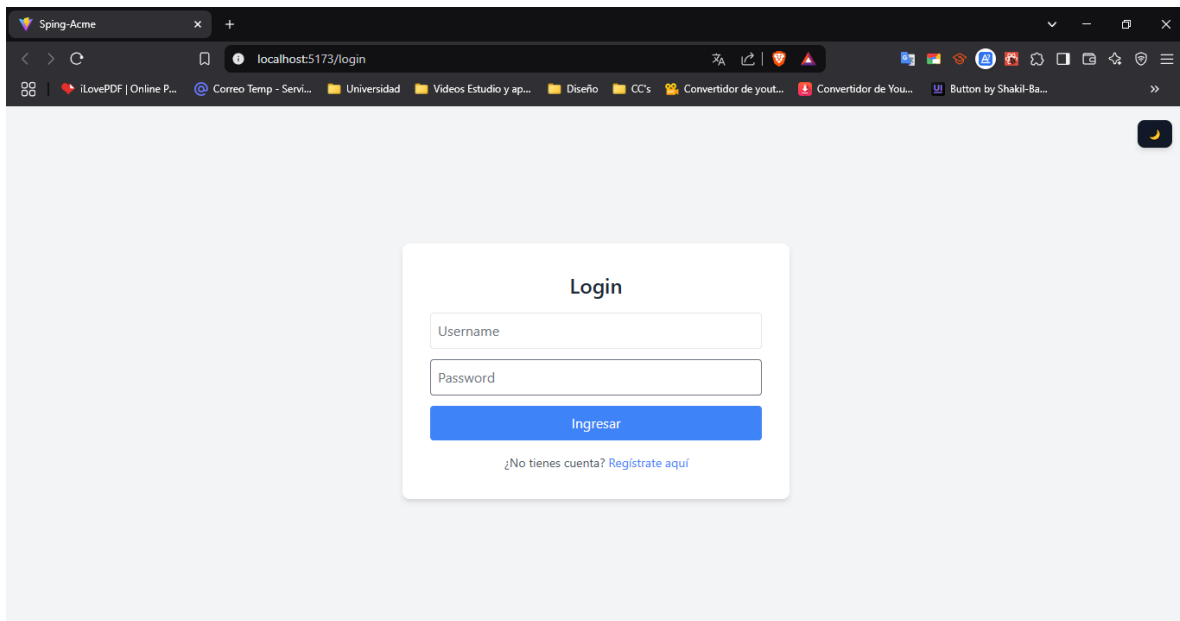
Se realizaron Entidades, Repositorios, Servicios y Controladores de cada entidad y lo que se requirió para el desarrollo de la aplicación. Se crearon respectivos endpoint en cada controlador para ser consumidos en el Frontend. Para las pruebas y validación de los Endpoint se implementó Postman.



En el siguiente paso se instaló y se creó el proyecto Frontend a través de React Vite en donde se diseñó y se creó una estructuración de carpetas del proyecto. Se establecieron los componentes, servicios y las páginas donde se iba a dar el recorrido la Red social. Se establecieron colores ajustes a la vista de los próximos usuarios y una implementación de una aplicación web dinámica.



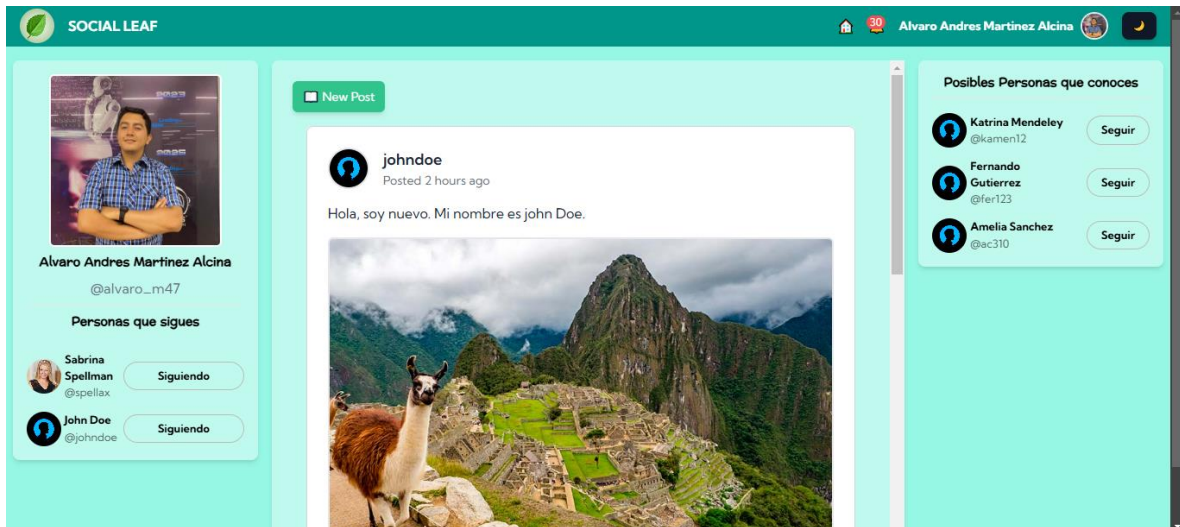
Se tuvo en cuenta los requerimientos como el Dark Mode y Light Mode en cierta parte en la paleta de colores como al mismo tiempo el uso de Reponsive. Por lo tanto se introdujo una opción para que el usuario pudiera usar el modo que mejor le parezca para su vista.



Al iniciar sesión, el usuario es redirigido a la página de inicio (Home), donde se muestra una pantalla principal con un fondo blanco. En el centro, hay un botón sencillo que permite crear la primera publicación. A los lados de esta sección principal, se muestran otros usuarios.

En la columna derecha, se presenta una lista de usuarios recomendados para seguir, mientras que, en la columna izquierda, se detalla una pequeña sección con información del usuario, así

como un apartado que muestra a quiénes está siguiendo. Esta información es visible para otros usuarios, quienes pueden ver la lista de seguidores si hacen clic sobre ellos en la columna derecha.



El usuario también puede ver su perfil y editar su perfil. El usuario puede ver sus posts.

