

Universidad de San Carlos de Guatemala.
Centro Universitario de Occidente.
División de Ciencias de la Ingeniería.
Teoría de Sistemas I.
Ing. Pedro Domingo.



“DOCUMENTACIÓN TÉCNICA - OLA KE HACE XELA”

Diego José Maldonado Monterroso.

Carné: 201931811.

Link Repo - Backend: <https://github.com/DiegoMaldonado19/ola-ke-hace-xela-backend>

Link Repo - Frontend: <https://github.com/DiegoMaldonado19/ola-ke-hace-xela-frontend>

Quetzaltenango, Guatemala.

04 de noviembre de 2024.

INTRODUCCIÓN

Este proyecto consiste en desarrollar una aplicación que permite la gestión de Publicaciones dentro del sector de Quetzaltenango, Guatemala. Aplicando conocimientos sobre PHP, Typescript y sus Frameworks, para poder realizar un sistema escalable y sostenible.

El objetivo es experimentar con diferentes tecnologías, como son Laravel y Angular, para poder obtener experiencia sobre el desarrollo de Aplicaciones Server Side y Client Side.

MARCO TEÓRICO

Capítulo 1

1.1 XAMPP

XAMPP es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. Es una herramienta esencial para desarrolladores que buscan un entorno de desarrollo local completo y fácil de usar.

1.2 Componentes de XAMPP

- Apache: El servidor web que maneja las solicitudes HTTP y sirve contenido web.
- MariaDB: Un sistema de gestión de bases de datos relacional, derivado de MySQL.
- PHP: Un lenguaje de scripting del lado del servidor, ampliamente utilizado para el desarrollo web.
- phpMyAdmin: Una herramienta escrita en PHP para la administración de MySQL y MariaDB a través de una interfaz web.

Capítulo 2

2.1 Apache

Apache es uno de los servidores web más populares y utilizados en el mundo. Su flexibilidad y extensibilidad lo hacen ideal para una amplia variedad de aplicaciones web.

2.2 Características de Apache

- Modularidad: Permite añadir funcionalidades mediante módulos.
- Compatibilidad: Soporta múltiples lenguajes de programación y tecnologías web.

Capítulo 3

3.1 MariaDB

MariaDB es un sistema de gestión de bases de datos relacional de código abierto, compatible con MySQL. Es conocido por su rendimiento, estabilidad y características avanzadas.

3.2 Características de MariaDB

- Alto Rendimiento: Optimizado para consultas rápidas y eficientes.
- Seguridad: Incluye características avanzadas de seguridad y cifrado.

Capítulo 4

4.1 PHP

PHP es un lenguaje de scripting del lado del servidor diseñado para el desarrollo web. Es conocido por su facilidad de uso y su capacidad para integrarse con HTML.

4.2 Características de PHP

- Simplicidad: Fácil de aprender y usar.
- Flexibilidad: Compatible con una amplia gama de bases de datos y servidores web.

Capítulo 5

5.1 phpMyAdmin

phpMyAdmin es una herramienta escrita en PHP para la administración de bases de datos MySQL y MariaDB a través de una interfaz web. Facilita la gestión de bases de datos, tablas, columnas y usuarios.

5.2 Funcionalidades de phpMyAdmin

- Interfaz Intuitiva: Permite realizar operaciones complejas de manera sencilla.
- Exportación e Importación: Soporta múltiples formatos de datos.

Capítulo 6

6.1 Introducción a Laravel

Laravel es un framework de PHP para el desarrollo de aplicaciones web con una sintaxis elegante y expresiva. Facilita tareas comunes como la autenticación, el enrutamiento y la gestión de sesiones.

6.2 Características de Laravel

Eloquent ORM: Un mapeador objeto-relacional que facilita la interacción con bases de datos.

Blade: Un motor de plantillas simple y potente.

Capítulo 7

7.1 Eloquent ORM

Eloquent es el ORM incluido en Laravel, que permite interactuar con bases de datos de manera sencilla y eficiente. Facilita la creación, lectura, actualización y eliminación de registros.

7.2 Funcionalidades de Eloquent ORM

- Consultas Dinámicas: Permite construir consultas SQL de manera programática.
- Relaciones: Facilita la definición de relaciones entre tablas.

Capítulo 8

8.1 Composer

Composer es un gestor de dependencias para PHP. Es una herramienta esencial para los desarrolladores que permite gestionar las librerías y paquetes necesarios para un proyecto, asegurando que todas las dependencias estén actualizadas y correctamente instaladas.

8.2 Características de Composer

- Gestión de Dependencias: Composer permite declarar las bibliotecas de las que depende un proyecto y las instala automáticamente.
- Autoloading: Facilita la carga automática de clases, lo que simplifica el desarrollo y mejora la organización del código.
- Versionado: Permite especificar versiones exactas de las dependencias, asegurando la compatibilidad y estabilidad del proyecto.

Capítulo 9

9.1 REST API

Una API REST es una interfaz que permite la comunicación entre sistemas utilizando el protocolo HTTP. Es ampliamente utilizada para la creación de servicios web escalables y eficientes.

9.2 Principios de las API REST

- Stateless: Cada solicitud contiene toda la información necesaria para procesarla.
- Cacheable: Las respuestas pueden ser almacenadas en caché para mejorar el rendimiento.

Capítulo 10

10.1 npm

npm es el gestor de paquetes para Node.js, que permite instalar, compartir y gestionar dependencias de proyectos JavaScript.

10.2 Uso de npm

- **Instalación de Paquetes:** npm install <paquete>
- **Gestión de Dependencias:** Facilita la actualización y eliminación de paquetes.

Capítulo 11

11.1 Angular CLI

Angular CLI es una herramienta de línea de comandos para inicializar, desarrollar y mantener aplicaciones Angular. Simplifica muchas tareas comunes en el desarrollo de aplicaciones Angular.

11.2 Funcionalidades de Angular CLI

- **Generación de Código:** Facilita la creación de componentes, servicios y otros elementos.
- **Automatización de Tareas:** Permite ejecutar pruebas, construir y desplegar aplicaciones.

Capítulo 12

12.1 Angular

Angular es un framework de desarrollo web basado en TypeScript, que permite crear aplicaciones web dinámicas y robustas.

12.2 Características de Angular

- **Componentes:** Facilita la creación de interfaces modulares y reutilizables.
- **Data Binding:** Permite la sincronización automática entre el modelo y la vista.

Capítulo 13

13.1 TypeScript

TypeScript es un superconjunto de JavaScript que añade tipado estático y otras características avanzadas. Es utilizado principalmente en el desarrollo de aplicaciones Angular.

13.2 Ventajas de TypeScript

- Tipado Estático: Ayuda a detectar errores en tiempo de desarrollo.
- Compatibilidad: Se compila a JavaScript, por lo que es compatible con cualquier entorno que soporte JavaScript.

Capítulo 14

14.1 Angular Material

Angular Material es una biblioteca de componentes UI para Angular, basada en las especificaciones de Material Design de Google.

14.2 Características de Angular Material

- Componentes Predefinidos: Ofrece una amplia gama de componentes listos para usar.
- Consistencia: Garantiza una apariencia y comportamiento consistentes en toda la aplicación.

VERSIONES DE SOFTWARE

Sistema Operativo: Microsoft Windows 11.

Backend (Xampp, MariaDB, Composer, Laravel, PHP)

- **Xampp:** v3.3.0
- **MariaDB:** v10.4.32
- **PHP:** v8.2
- **Composer:** v2.8.1
- **Laravel:** v11.9
- **Laravel Sanctum:** v4.0
- **Dependencias:**

```
"require": {  
  "php": "^8.2",  
  "laravel/framework": "^11.9",  
  "laravel/sanctum": "^4.0",  
  "laravel/tinker": "^2.9"  
},
```

Frontend (Typescript, Angular, Angular CLI, Angular Material, NPM)

- **NPM:** v10.8.2
- **Typescript:** v5.5.2
- **Angular:** v18.2
- **Angular CLI:** v18.2.7
- **Angular Material:** v18.2.10
- **Dependencias:**

```
"dependencies": {  
  "@angular/animations": "^18.2.0",  
  "@angular/cdk": "^18.2.10",  
  "@angular/common": "^18.2.0",  
  "@angular/compiler": "^18.2.0",  
  "@angular/core": "^18.2.0",  
  "@angular/forms": "^18.2.0",  
  "@angular/material": "^18.2.10",  
  "@angular/material-moment-adapter": "^18.2.11",  
  "@angular/platform-browser": "^18.2.0",  
  "@angular/platform-browser-dynamic": "^18.2.0",  
  "@angular/platform-server": "^18.2.0",  
  "@angular/router": "^18.2.9",  
  "@angular/ssr": "^18.2.7",  
  "express": "^4.18.2",  
  "moment": "^2.30.1",  
  "rxjs": "~7.8.0",  
  "tslib": "^2.3.0",  
  "zone.js": "~0.14.10"
```


CONFIGURACIÓN DE ENTORNO

1. Debes instalar los siguientes programas: PHP, Xampp, Composer, Laravel, NodeJS, NPM, Angular CLI.

PHP: <https://windows.php.net/download/>

Xampp: <https://www.apachefriends.org/es/download.html>

Composer: <https://getcomposer.org/download/>

Laravel: <https://laravel.com/docs/11.x/installation>

NodeJS: <https://nodejs.org/en/download/package-manager>

NPM: <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>

Angular CLI: <https://angular.dev/installation>

2. Debes clonar los siguientes repositorios:

Backend: <https://github.com/DiegoMaldonado19/ola-ke-hace-xela-backend>

Frontend: <https://github.com/DiegoMaldonado19/ola-ke-hace-xela-frontend>

3. Debes abrir ambos repositorios en tu IDE de preferencia.
4. Para el repositorio de Backend, deberas eliminar el archivo: **composer.lock**
5. Luego debes correr el siguiente comando desde consola: **composer install**
6. Para el repositorio de Frontend, deberas eliminar los siguientes archivos:
 - a. **package-json.lock**
 - b. carpeta: **node_modules**
7. Luego debes correr el siguiente comando: **npm install**
8. Con los pasos 4, 5, 6 y 7, lo que hacemos es instalar las dependencias de nuestros proyectos en nuestra maquina local.
9. Ahora solo nos queda correr los siguientes comandos en las consolas respectivas de nuestros proyectos:
 - a. **php artisan serve**
 - b. **ng serve**

10. Y listo, tendremos nuestro entorno listo para usar la aplicación:

Backend:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS Backend\ola-ke-hace-xela> php artisan serve
[INFO] Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

Frontend:

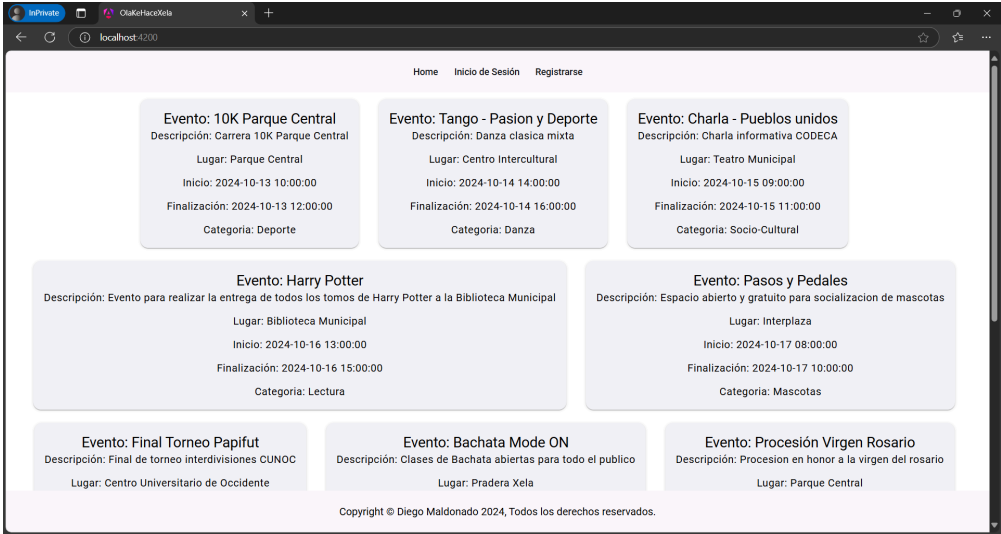
```
PS Frontend\ola-ke-hace-xela> ng serve
Browser bundles
Initial chunk files | Names | Raw size
main.js | main | 142.41 kB |
polyfills.js | polyfills | 90.20 kB |
styles.css | styles | 76.79 kB |
Initial total | 309.40 kB

Server bundles
Initial chunk files | Names | Raw size
polyfills.server.mjs | polyfills.server | 572.91 kB |
main.server.mjs | main.server | 142.86 kB |
render-utils.server.mjs | render-utils.server | 472 bytes |

Application bundle generation complete. [2.263 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
+ Local: http://localhost:4200/
+ press h + enter to show help
```

Aplicación:



ANEXOS

Diagrama Entidad Relación

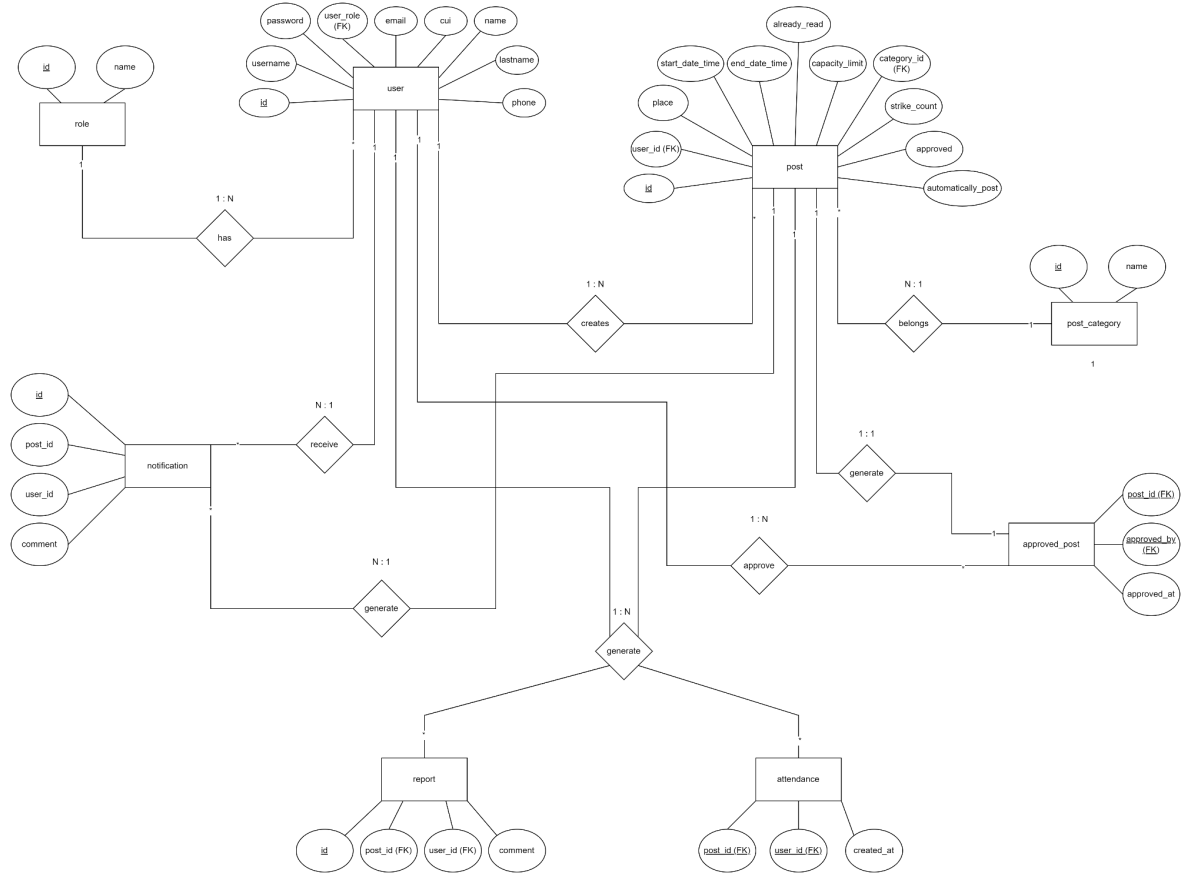


Diagrama de Paquetes Backend.

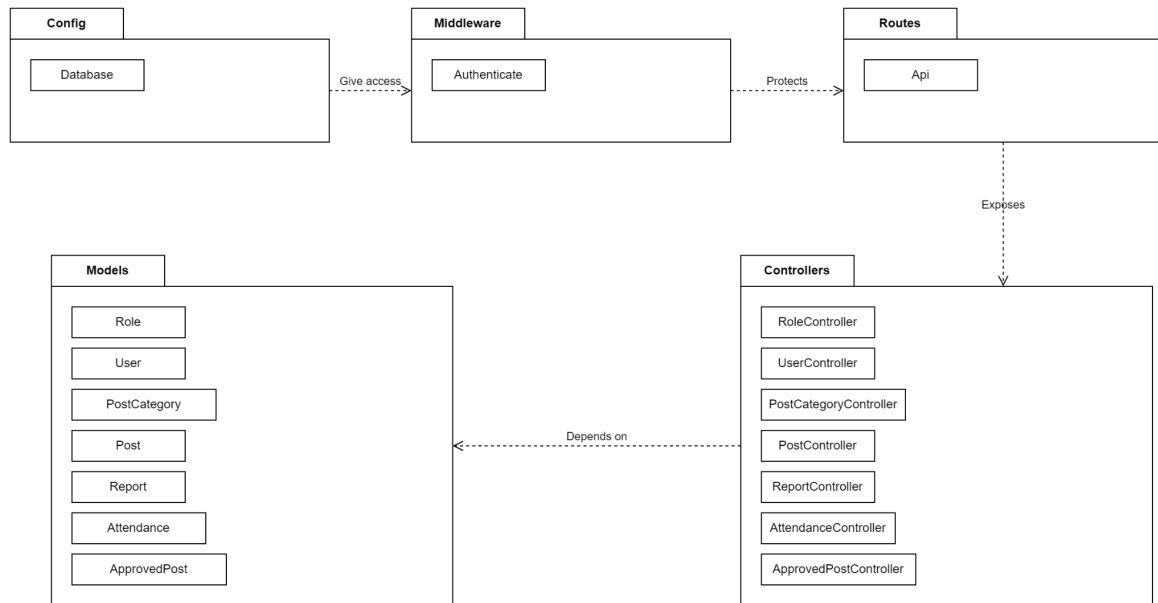


Diagrama de Paquetes Frontend.

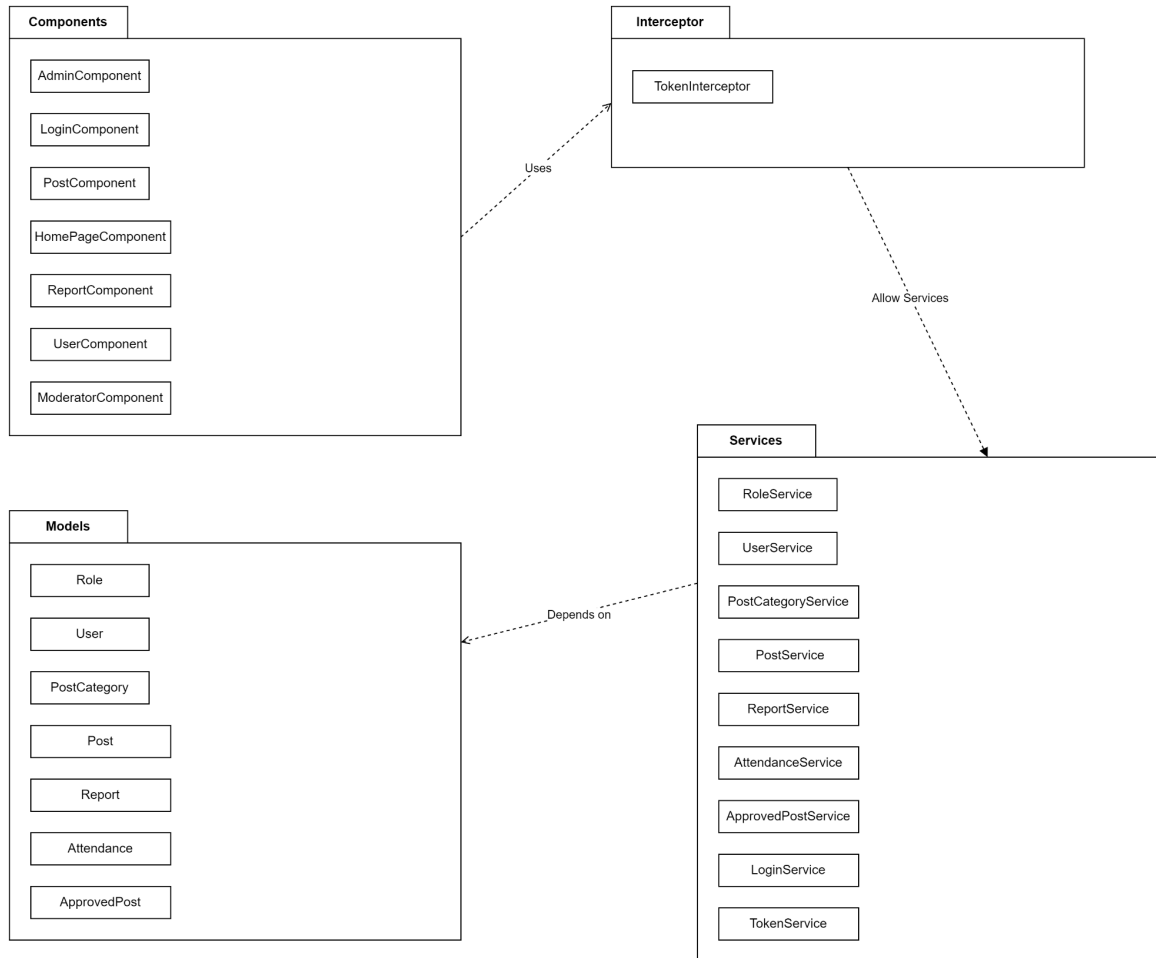


Diagrama de Despliegue.

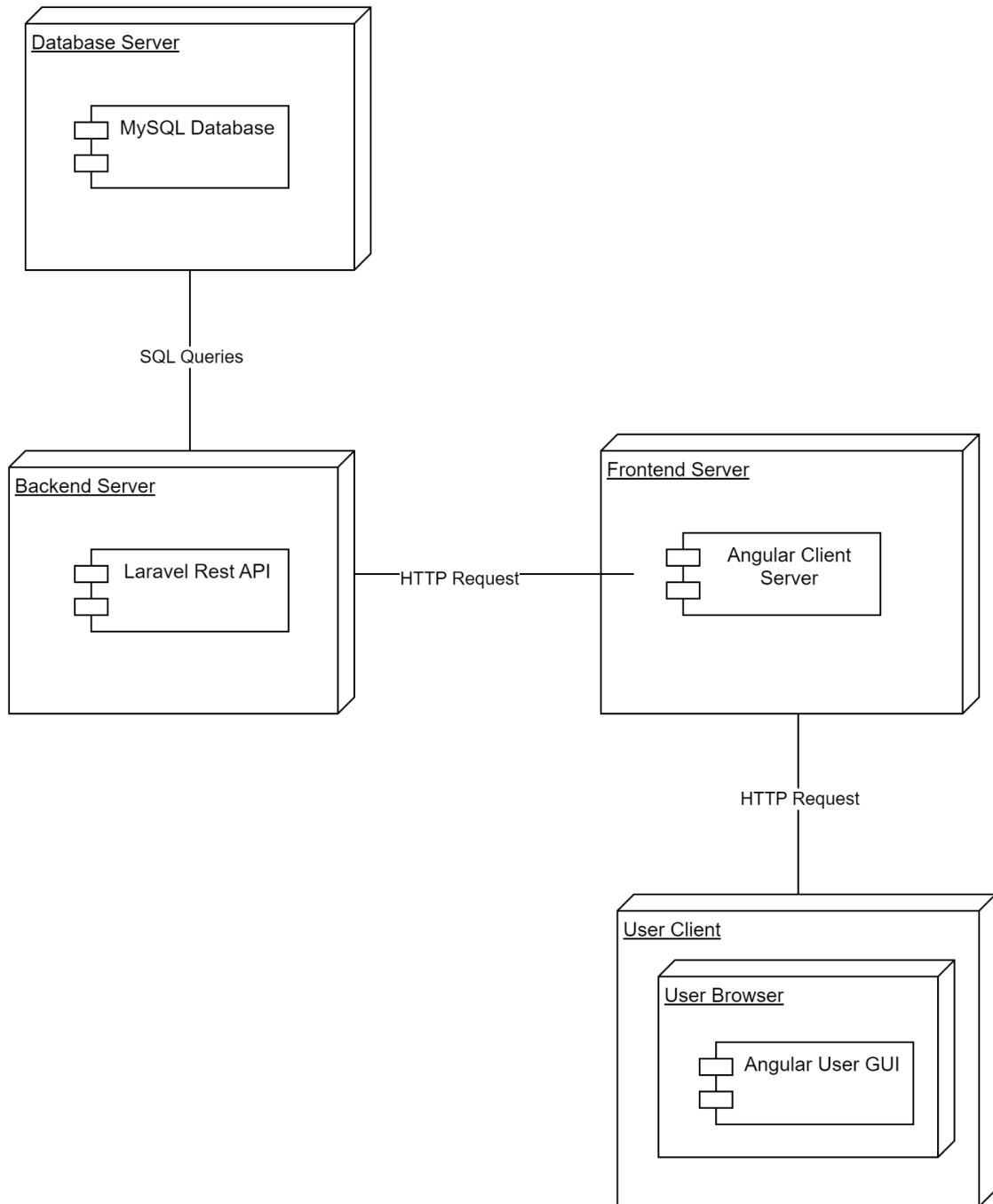


Diagrama de Secuencia de la pagina principal del sistema

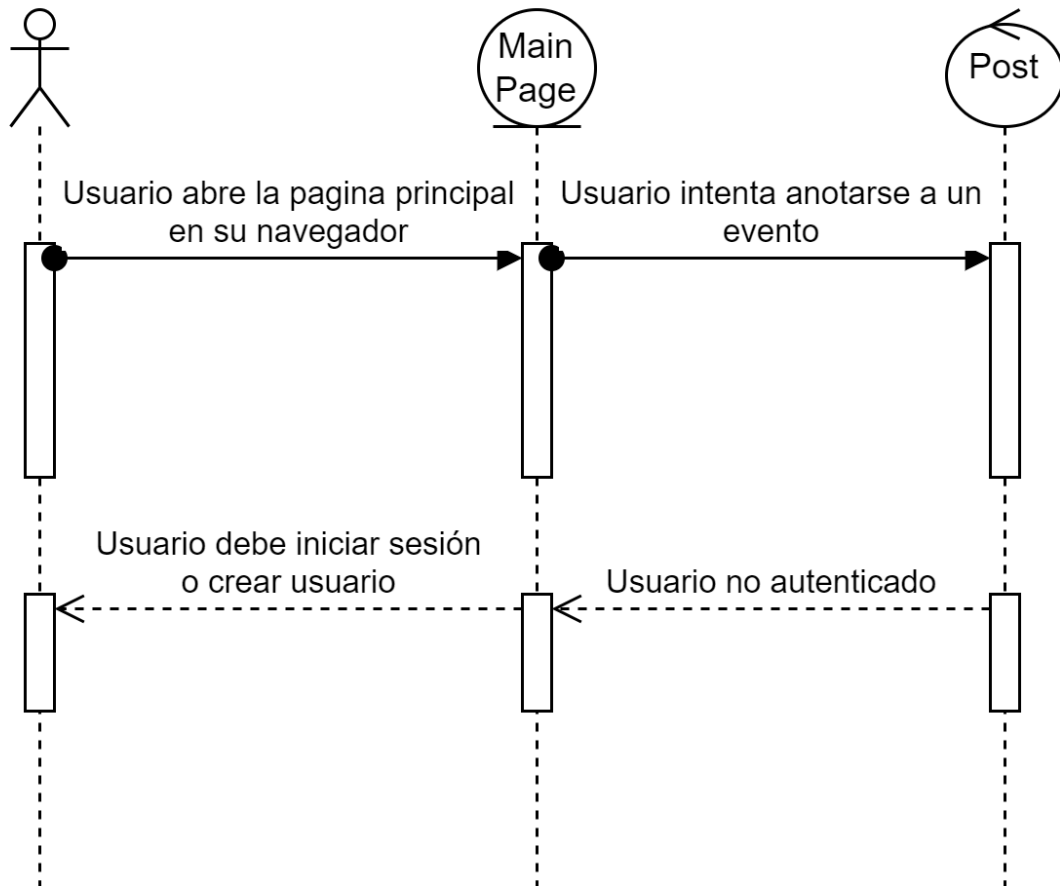


Diagrama de Secuencia de pagina de Login

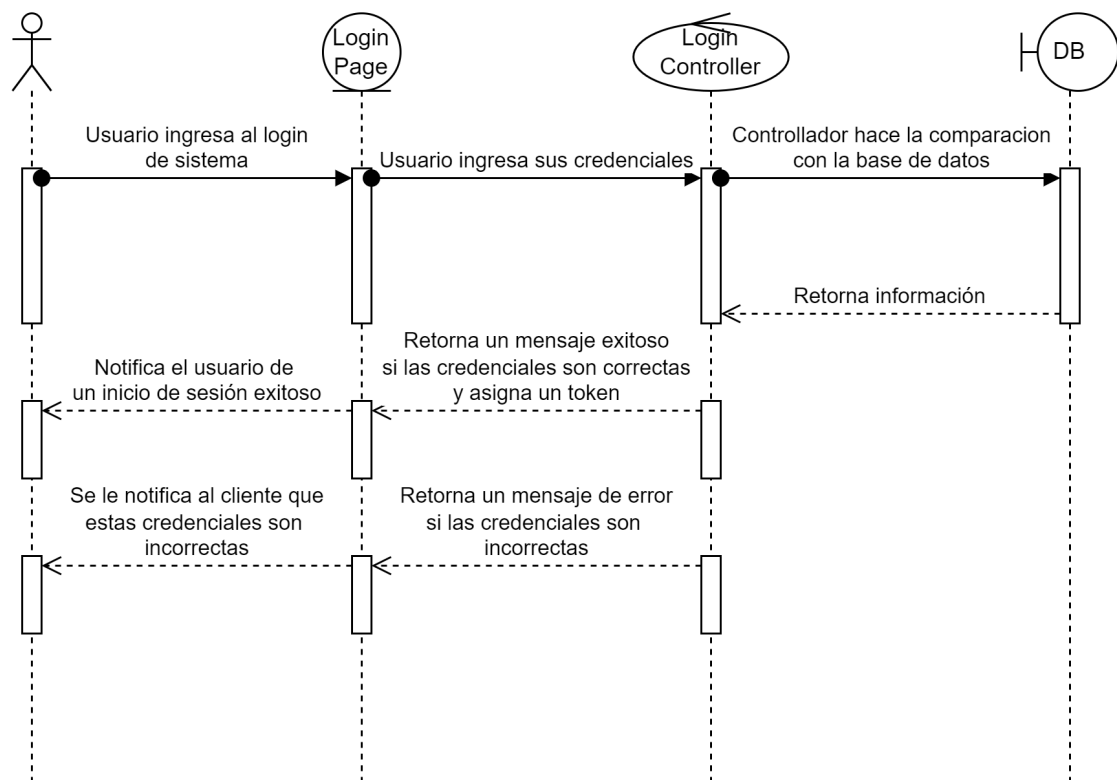


Diagrama de Secuencia apartado de Notificaciones

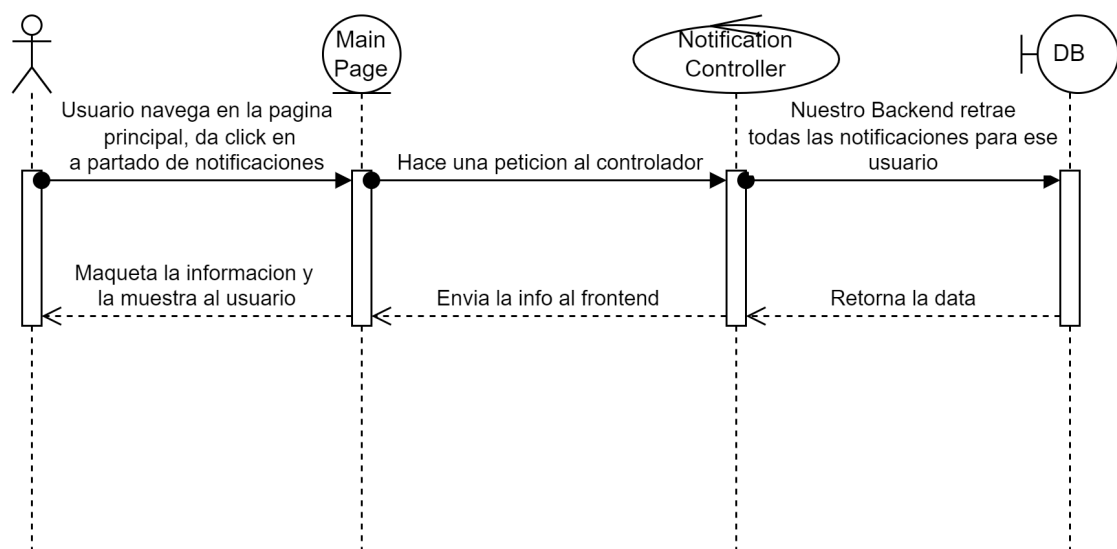


Diagrama de Secuencia Creación de Anuncio

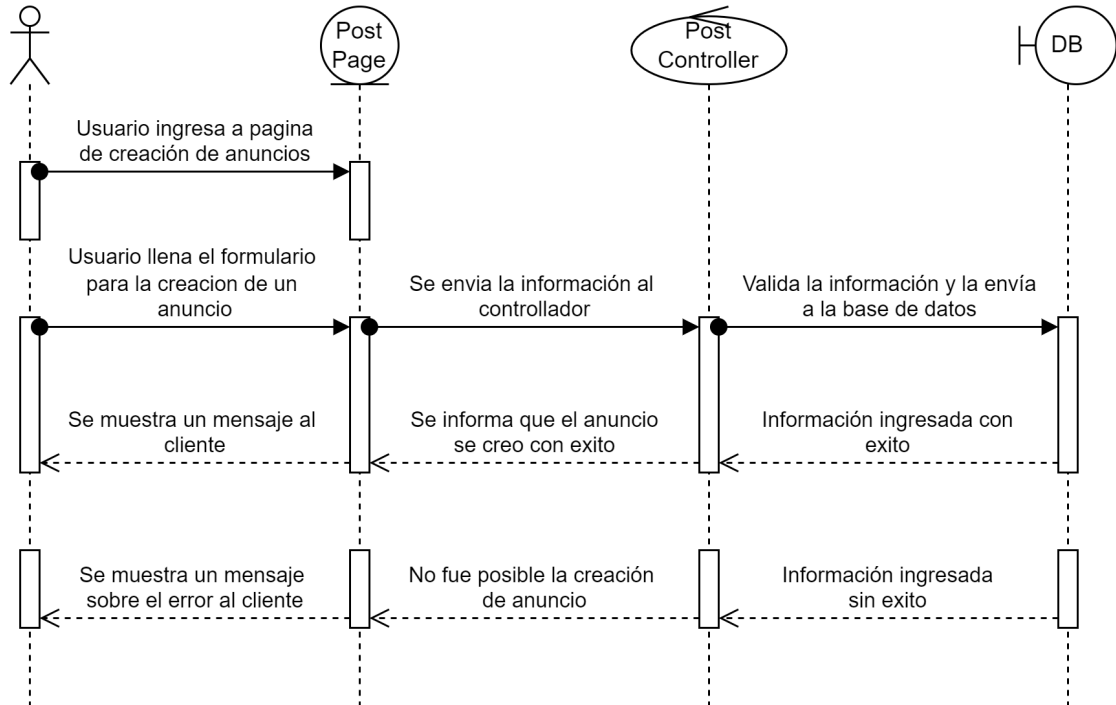


Diagrama de aprobación de Post por parte de los Moderadores

