

Arquitectura de la solución

La arquitectura de la aplicación se dividirá en tres capas:

- Capa de presentación: Esta capa se encargará de mostrar la interfaz de usuario de la aplicación. Esta capa estará formada por componentes React.
- Capa de aplicación: Esta capa se encargará de la lógica de negocio de la aplicación. Esta capa estará formada por servicios y controladores.
- Capa de datos: Esta capa se encargará de la interacción con la base de datos. Esta capa estará formada por modelos y repositorios.

Componentes React

Los componentes React se dividirán en dos categorías:

- Componentes de presentación: Estos componentes se encargarán de mostrar la interfaz de usuario de la aplicación.
- Componentes de estado: Estos componentes se encargarán de manejar el estado de la aplicación.

Servicios y controladores

Los servicios se encargarán de la lógica de negocio de la aplicación. Los controladores se encargarán de la comunicación entre la capa de presentación y la capa de datos.

Modelos y repositorios

Los modelos representarán los datos de la base de datos. Los repositorios se encargarán de la interacción con la base de datos.

MongoDB

La aplicación se conectará a una base de datos MongoDB para persistir datos de sesión y usuarios. La base de datos se utilizará para almacenar los siguientes datos:

- Usuarios: Nombre de usuario, correo electrónico, contraseña, etc.
- Sesiones: ID de sesión, usuario autenticado, etc.

Buenas prácticas

La aplicación utilizará las siguientes buenas prácticas:

- Routing: Se utilizará el routing de React para controlar la navegación entre las diferentes páginas de la aplicación.
- Estado global: Se utilizará un estado global para almacenar datos que sean necesarios para toda la aplicación.
- Contexts: Se utilizarán los contexts de React para compartir datos entre componentes.
- Hooks: Se utilizarán los hooks de React para simplificar el código.
- Módulos: Se utilizarán los módulos de JavaScript para organizar el código.

Pruebas

Los componentes principales de la aplicación se cubrirán con pruebas unitarias y de integración.

Desarrollo

El desarrollo de la aplicación se realizará en las siguientes etapas:

1. Instalación de dependencias: Se instalarán las dependencias necesarias para el desarrollo de la aplicación.
2. Creación de la base de datos: Se creará la base de datos MongoDB.
3. Creación de los modelos: Se crearán los modelos de datos para la base de datos.
4. Creación de los repositorios: Se crearán los repositorios para la base de datos.
5. Creación de los servicios: Se crearán los servicios para la lógica de negocio.
6. Creación de los controladores: Se crearán los controladores para la comunicación con la capa de datos.
7. Creación de los componentes: Se crearán los componentes React.
8. Implementación de la lógica de negocio: Se implementará la lógica de negocio de la aplicación.
9. Implementación del routing: Se implementará el routing de la aplicación.
10. Implementación del estado global: Se implementará el estado global de la aplicación.
11. Implementación de los contexts: Se implementarán los contexts de la aplicación.
12. Implementación de los hooks: Se implementarán los hooks de la aplicación.

13. Implementación de las pruebas: Se implementarán las pruebas unitarias y de integración de la aplicación.
14. Despliegue de la aplicación: Se desplegará la aplicación en un servidor web.

Codigo mermaid, ejecutado en <https://mermaid.live/>

sequenceDiagram

participant Frontend

participant Backend

participant Database

Frontend->>Backend: /api/auth/login

Backend->>Database: Consultar usuario por nombre de usuario y contraseña

Database->>Backend: Devuelve usuario

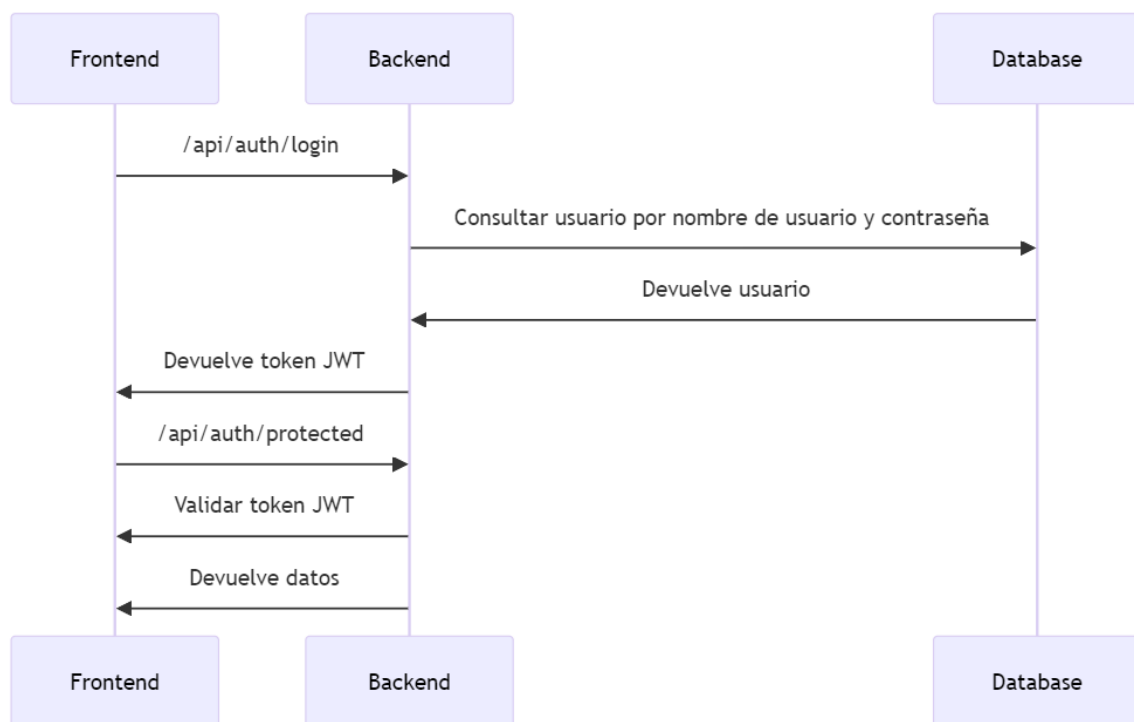
Backend->>Frontend: Devuelve token JWT

Frontend->>Backend: /api/auth/protected

Backend->>Frontend: Validar token JWT

Backend->>Frontend: Devuelve datos

Diagrama de secuencia



Comandos para arrancar los servicios:

Base de Datos

> mongod

Server

> npm start

Client

> npm run dev