Pasos a grandes rasgos para diseñar un e-commerce con Angular y Spring Boot

1. Definir los requisitos: Antes de comenzar a diseñar, es importante tener claros los requisitos del e-commerce, como los tipos de productos que se venderán, los métodos de pago y envío, las funcionalidades de búsqueda y filtros, entre otros.
2. Diseñar la base de datos: Una vez que se tienen claros los requisitos, es importante diseñar la base de datos que almacenará toda la información del e-commerce. Se puede utilizar una base de datos relacional como MySQL o una base de datos no relacional como MongoDB.
3. Crear el backend con Spring Boot: Para el backend, se utilizará Spring Boot, un framework de Java que permite crear aplicaciones web de manera rápida y sencilla. En el backend, se crearán los endpoints para realizar las operaciones CRUD (crear, leer, actualizar y eliminar) sobre la base de datos.
4. Crear el frontend con Angular: Para el frontend, se utilizará Angular, un framework de JavaScript que permite crear aplicaciones web modernas y responsivas. En el frontend, se diseñará la interfaz de usuario que permitirá a los usuarios realizar las distintas operaciones del e-commerce, como buscar productos, agregarlos al carrito de compras, realizar pagos, entre otros.
5. Integrar el frontend y el backend: Finalmente, se integrarán el frontend y el backend para que la aplicación funcione de manera completa. Para ello, se consumirán los endpoints del backend desde el frontend utilizando HTTP requests.

Algunas consideraciones adicionales a tener en cuenta durante el diseño incluyen la seguridad de la aplicación, la optimización del rendimiento y la implementación de pruebas automatizadas para asegurar que la aplicación funcione correctamente.

Ir consultando a gustavo todo tipo de dudas.

Se debe crear el front con Angular por separado del back con Springboot?

Para crear una aplicación web con el front-end en Angular y el back-end en Spring Boot, normalmente se crean por separado y luego se conectan a través de una API REST.

Angular es un framework de JavaScript que se utiliza para desarrollar la parte del cliente (front-end) de una aplicación web. Con Angular, se puede crear una interfaz de usuario dinámica e interactiva que permita a los usuarios interactuar con la aplicación.

Por otro lado, Spring Boot es un framework de Java que se utiliza para desarrollar la parte del servidor (back-end) de una aplicación web. Con Spring Boot, se puede crear una API REST que permita a la parte del cliente (Angular) interactuar con la base de datos y realizar operaciones en el servidor.

Para conectar Angular y Spring Boot, normalmente se utiliza HTTP para enviar y recibir datos entre el cliente y el servidor. Por ejemplo, cuando un usuario realiza una solicitud en la aplicación Angular, se envía una solicitud HTTP al servidor Spring Boot, que procesa la solicitud y devuelve una respuesta. Esta respuesta puede incluir datos que Angular utiliza para actualizar la interfaz de usuario.

En resumen, se pueden crear por separado el front-end en Angular y el back-end en Spring Boot y luego conectarlos a través de una API REST utilizando HTTP para enviar y recibir datos.

Paro aunque estén separados deben compartir lógica:

podemos entender cómo el front-end y el back-end se desarrollan por separado pero trabajan juntos para proporcionar una experiencia de usuario satisfactoria.

Imaginemos que estamos desarrollando un e-commerce de ropa en línea. En este caso, el front-end se desarrollaría con Angular, y el back-end con Spring Boot.

En el front-end, se crearían las vistas para que el usuario pueda ver los productos, agregarlos al carrito, realizar el pago, entre otras funcionalidades. Por ejemplo, podríamos tener una vista que muestre una lista de productos disponibles, y que permita filtrarlos por tipo de ropa, talla, color, entre otros criterios.

Cuando el usuario seleccione un producto y lo agregue al carrito, el front-end enviará una solicitud HTTP POST al back-end para que se guarde la información del producto en el servidor. El back-end, que ha sido desarrollado con Spring Boot, recibirá la solicitud, la procesará y guardará la información del producto en la base de datos.

Luego, cuando el usuario decida realizar el pago, el front-end enviará una solicitud HTTP POST al back-end para realizar la transacción. El back-end procesará la solicitud, validará la información del pago y actualizará la base de datos con la información de la transacción.

En este caso, la API REST actúa como un puente que permite que el front-end y el back-end se comuniquen y compartan información. El front-end se encarga de la interfaz de usuario, mientras que el back-end maneja la lógica de negocios y la base de datos.

Además, en un e-commerce, también es importante la autenticación de usuarios, la validación de datos y la seguridad de las transacciones, entre otras funcionalidades que pueden ser compartidas entre el front-end y el back-end.