Lo que deberíamos pasar al backend (Java)

Gestión de transacciones (CRUD)

La lógica de guardar transacciones en la base de datos actualmente está simulada en guardarTransaccion en el componente Calendario. Esto debe trasladarse al backend para que las transacciones se guarden y recuperen desde una base de datos.

Cálculo y filtrado de transacciones (por ejemplo, obtenerTransacciones) debería ser manejado en el backend para optimizar la carga en el cliente, especialmente si hay muchas transacciones.

Tipos de gastos e ingresos

Las categorías de gastos e ingresos (categoriasGastos y categoriasIngresos) deberían ser configurables desde el backend, ya que podrían ser dinámicas en el futuro.

Validaciones importantes

Por ejemplo, validar si la cantidad es válida y si la categoría existe. Actualmente, estas validaciones son simples en el frontend, pero es crítico tenerlas en el backend para evitar datos corruptos.

Autenticación y roles

Aunque no aparece en este fragmento, cualquier gestión de usuarios (registro, inicio de sesión, asignación de roles) debería ser manejada completamente en el backend para garantizar la seguridad.

Estadísticas

El cálculo de estadísticas (como total de ingresos, gastos, o metas alcanzadas) puede ser costoso en el frontend si hay una gran cantidad de datos. Mover estas operaciones al backend simplifica el trabajo en Vue.

Lo que deberíamos mantener en el frontend (Vue.js)

Interactividad del calendario

El cambio de meses (mesAnterior, mesSiguiente) y el renderizado del calendario interactivo pueden quedarse en el frontend, ya que son funcionalidades visuales.

Modales y formularios

La apertura y cierre de modales, así como la lógica para mostrar el formulario de nueva transacción, deben mantenerse en Vue.js.

Validaciones básicas

Validaciones simples como asegurarse de que el campo de cantidad no está vacío o que se selecciona una categoría.

Iconos y estilos dinámicos

La lógica para aplicar clases según el tipo de transacción (como ingreso o gasto) es específica de la interfaz y debería mantenerse aquí.

Renderizado de listas

Renderizar las listas de categorías e iconos directamente desde el backend pero gestionadas en Vue.js (usando datos recibidos por API).

un API REST en el backend para manejar las siguientes operaciones principales:

GET /transacciones - Devuelve todas las transacciones.

POST /transacciones - Crea una nueva transacción.

GET /categorias - Devuelve todas las categorías de gastos e ingresos.

GET /estadisticas - Devuelve estadísticas calculadas (total ingresos/gastos, etc.).