ACTIVIDAD 3

DESAROLLLO FULL STACK

Miguel Armando Rascon Dominguez

Reglas de negocio en el proyecto de pedidos

Qué reglas de negocio implementé

- *Validación de cantidad y producto: No se permite crear un pedido si la cantidad es menor o igual a cero o si falta el nombre del producto.*
- *Estados de pedido: Cada pedido inicia en estado pendiente. Solo puede cambiar a confirmado o cancelado.*
- *Restricciones de actualización: Un pedido solo puede modificarse si está en estado pendiente. Si ya está confirmado o cancelado, no se puede alterar.*
- *Restricciones de eliminación: Un pedido solo puede eliminarse si está en estado pendiente. Confirmados o cancelados no se eliminan.*

En qué archivo viven esas reglas

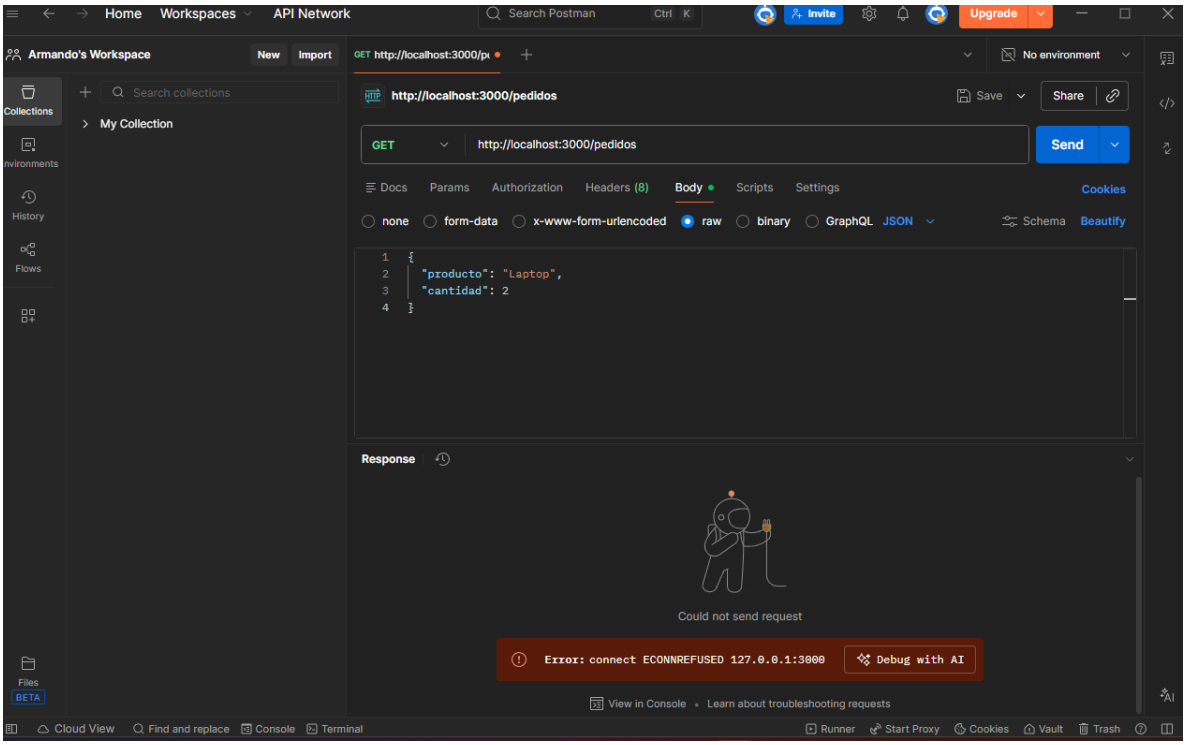
- *src/controllers/pedidos.controller.js: Aquí se implementan las validaciones de negocio antes de llamar al repositorio. Por ejemplo, se valida la cantidad, el estado permitido y las restricciones de actualización/eliminación.*
- *src/repositories/pedidos.repository.js: Aquí se maneja la persistencia en memoria de los pedidos, pero sin lógica de negocio. Solo operaciones CRUD básicas.*

Por qué tomé esas decisiones

- *Separación de responsabilidades: La lógica de negocio vive en el controller porque es el punto donde se procesan las reglas antes de interactuar con los datos. El repository se mantiene limpio para que solo gestione almacenamiento.*
- *Claridad y mantenibilidad: Al separar validaciones y reglas en el controller, el código es más fácil de mantener y extender. Si*

en el futuro se agregan más reglas (por ejemplo, límites de stock), se pueden añadir en el controller sin afectar la capa de datos.

- *Consistencia: Definir estados claros y restricciones evita inconsistencias en el flujo de pedidos, asegurando que solo se modifiquen o eliminen en condiciones válidas.*



PROFE SAE QUE EN LA PARTE DE VISUAL STUDIO SI ME DEJABA CORRER EL CODIGO PERO YA EN POSTMAN ME SALIA ESE ERROR