

# Explicación Detallada del Proyecto - Backend y Frontend

## 1. Estructura General del Proyecto

El proyecto se compone de dos partes:

- Backend: desarrollado en Spring Boot, contiene la lógica de negocio, modelos, controladores y servicios.
- Frontend: desarrollado en Angular, ofrece la interfaz visual y consume los servicios REST expuestos por el backend.

## 2. Backend - Spring Boot

Estructura de carpetas:

- model / entities: Contiene las clases que representan las tablas de la base de datos (Pedido.java).
- repository: Interfaces que heredan de JpaRepository para acceso a datos.
- service: Lógica de negocio (validaciones, procesamiento antes de guardar).
- controller: Maneja las rutas de API REST.
- application.properties: Configura base de datos y puerto del servidor.

## 3. Módulo de Pedidos - Backend

Entidad Pedido.java:

Representa la tabla 'pedidos' en la base de datos. Incluye campos como idCliente, idVendedor, fecha, direccion, idProducto, nombreProducto, cantidad, valor y totalPedido.

Repositorio PedidoRepository.java:

Extiende JpaRepository<Pedido, Integer> para obtener métodos como findAll, save, deleteById automáticamente.

Servicio PedidoService.java:

Contiene métodos:

- listarPedidos(): devuelve todos los pedidos.
- guardarPedido(): guarda un nuevo pedido.
- actualizarPedido(): actualiza los datos de un pedido existente.
- eliminarPedido(): elimina un pedido por ID.

Controlador PedidoController.java:

Maneja las rutas HTTP:

- GET /api/pedidos
- POST /api/pedidos
- PUT /api/pedidos/{id}
- DELETE /api/pedidos/{id}

## 4. Frontend - Angular

El frontend consume los endpoints expuestos por el backend y presenta la información en la interfaz visual del usuario.

Estructura:

- pedidos.component.ts: Componente que contiene la lógica del formulario y tabla.

## Explicación Detallada del Proyecto - Backend y Frontend

- pedidos.component.html: Contiene el formulario para crear/editar y una tabla para visualizar pedidos.
- pedido.service.ts: Servicio que conecta con la API REST del backend.

Funciones clave:

- listarPedidos(): carga todos los pedidos y los muestra en la tabla.
- guardarPedido(): guarda o actualiza un pedido según el modo (crear o editar).
- editarPedido(): carga los datos en el formulario para editar.
- eliminarPedido(): elimina un pedido seleccionado.
- cancelarEdicion(): resetea el formulario.

### 5. Flujo del módulo de pedidos

1. El usuario abre la ruta /pedidos.
2. Angular llama a GET /api/pedidos y carga los datos en la tabla.
3. Al llenar el formulario y dar clic en 'Guardar', se envía un POST (nuevo) o PUT (editar).
4. Al dar clic en 'Eliminar', se hace DELETE al backend.
5. Todos los cambios se actualizan en tiempo real en la interfaz.