

# Resumen del Proyecto Angular + Spring Boot

Este documento describe todos los pasos y configuraciones aplicadas para que el sistema de gestión

de productos y usuarios funcione correctamente tanto en el backend (Spring Boot) como en el frontend (Angular).

=== BACKEND (Spring Boot) ===

## 1. Configuración del archivo application.properties:

- Se definió la URL, usuario y contraseña de la base de datos MySQL.
- Se especificó el driver de conexión y el puerto del servidor (8000).

## 2. Producto:

- Se creó la entidad Producto con anotaciones @Entity y @Table.
- Se corrigió el nombre de la columna ID para que coincida con la base de datos: @Column(name = "id\_productos").
- Se implementó un repositorio ProductoRepository que extiende JpaRepository.
- Se desarrolló el controlador ProductoController con los endpoints: GET, POST, PUT y DELETE.

## 3. Usuario:

- Ya estaba implementado UserController, UserService y UserRepository.
- Se detectó que los usuarios no se estaban guardando, por lo que se revisó la conexión con el frontend.

## 4. Verificación de errores:

- Se resolvió un error SQL relacionado con el campo 'id' que no existía en la tabla 'productos'.

- Se validó que la aplicación Spring Boot levantaba correctamente revisando la consola.

### === FRONTEND (Angular) ===

#### 1. ProductosComponent:

- Se creó el componente con un formulario reactivo para crear y editar productos.
- Se implementó una tabla para listar los productos.
- Se agregaron funciones para editar y eliminar productos.

#### 2. ProductoService:

- Se definieron métodos para comunicarse con el backend: `getProductos`, `crearProducto`, `actualizarProducto` y `eliminarProducto`.

#### 3. Manejo de datos:

- Se resolvió el problema de inicialización incorrecta que mostraba ceros en los campos cantidad y precio.
- Se añadió una función `calcularTotal()` para mostrar el valor total en la tabla ( $\text{cantidad} * \text{precio}$ ).

#### 4. Integración de SweetAlert2:

- Se aplicaron alertas visuales en guardado y eliminación tanto para productos como usuarios.

#### 5. Usuarios:

- Se revisó el flujo de eventos emitidos y se verificó la suscripción en `SharingDataService`.
- Se confirmó que el formulario de usuario estaba enviando datos correctamente al backend.

### === Validaciones Realizadas ===

- Se verificó el inicio correcto del backend observando el log de Tomcat.
- Se usó Postman y la consola del navegador para confirmar el flujo de datos.
- Se realizaron pruebas de inserción, actualización y eliminación con resultados satisfactorios.

Este resumen puede servirte como guía de implementación para proyectos similares en el futuro.