

Caso Agregar con Spring Boot



Creación del caso de agregar producto en Spring Boot

En esta guía, se explicará cómo se implementó la funcionalidad para agregar un producto en el backend con **Spring Boot**. Se modificó el **controlador** y el **servicio** para retornar el producto recién agregado, permitiendo que Angular pueda desplegar la respuesta correctamente.

Paso 1: Modificación del controlador `ProductoControlador.java`

Código actualizado

```
package gm.inventarios.controlador;

import gm.inventarios.modelo.Producto;
import gm.inventarios.servicio.ProductoServicio;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;
```

```

@RestController
//http://localhost:8080/inventario-app
@RequestMapping("inventario-app")
@CrossOrigin(value = "http://localhost:4200")
public class ProductoControlador {

    private static final Logger logger =
        LoggerFactory.getLogger(ProductoControlador.class);

    @Autowired
    private ProductoServicio productoServicio;

    //http://localhost:8080/inventario-app/productos
    @GetMapping("/productos")
    public List<Producto> obtenerProductos(){
        List<Producto> productos = this.productoServicio.listarProductos();
        logger.info("Productos obtenidos:");
        productos.forEach((producto -> logger.info(producto.toString())));
        return productos;
    }

    @PostMapping("/productos")
    public Producto agregarProducto(@RequestBody Producto producto){
        logger.info("Producto a agregar: " + producto);
        return this.productoServicio.guardarProducto(producto);
    }
}

```

Explicación de los cambios

1. Se agregó el método `agregarProducto()`:
 - o Se define con `@PostMapping("/productos")`, lo que indica que manejará peticiones HTTP POST.
 - o Usa `@RequestBody Producto producto` para recibir un objeto JSON desde Angular.
 - o Llama a `productoServicio.guardarProducto(producto)` y retorna el producto recién guardado.
2. ¿Por qué se retorna el producto agregado?
 - o Permite que Angular muestre los datos inmediatamente después de agregarlos.
 - o Si no se retornara, Angular no recibiría confirmación sobre los datos guardados.

Paso 2: Modificación del servicio `ProductoServicio.java`

Código actualizado

```

package gm.inventarios.servicio;

import gm.inventarios.modelo.Producto;

```

```

import gm.inventarios.repositorio.ProductoRepositorio;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class ProductoServicio implements IProductoServicio{

    @Autowired
    private ProductoRepositorio productoRepositorio;

    @Override
    public List<Producto> listarProductos() {
        return this.productoRepositorio.findAll();
    }

    @Override
    public Producto buscarProductoPorId(Integer idProducto) {
        Producto producto =
            this.productoRepositorio.findById(idProducto).orElse(null);
        return producto;
    }

    @Override
    public Producto guardarProducto(Producto producto) {
        return this.productoRepositorio.save(producto);
    }

    @Override
    public void eliminarProductoPorId(Integer idProducto) {
        this.productoRepositorio.deleteById(idProducto);
    }
}

```

Explicación de los cambios

- Se modificó `guardarProducto()` para retornar el producto guardado:

```

public Producto guardarProducto(Producto producto) {
    return this.productoRepositorio.save(producto);
}

```

- Antes, el método solo guardaba el producto sin devolverlo. Ahora retorna el **producto con su ID generado**, lo que es útil en Angular para actualizar la vista.

Paso 3: Implementación en la interfaz `IProductoServicio.java`

Código actualizado

```

package gm.inventarios.servicio;

import gm.inventarios.modelo.Producto;
import java.util.List;

```

```
public interface IProductoServicio {  
    List<Producto> listarProductos();  
  
    Producto buscarProductoPorId(Integer idProducto);  
  
    Producto guardarProducto(Producto producto);  
  
    void eliminarProductoPorId(Integer idProducto);  
}
```

Explicación

- Se agregó la firma del método **Producto guardarProducto (Producto producto)** en la interfaz.
 - Esto garantiza que cualquier implementación de **IProductoServicio** debe incluir este método regresando el objeto **Producto** en su firma.
-

Paso 4: Pruebas con Postman

Para verificar que el nuevo método funciona correctamente:

1. Abrir Postman.
2. Hacer una petición **POST** a:

`http://localhost:8080/inventario-app/productos`

3. Enviar un **JSON** en el cuerpo de la solicitud (Seleccionar Body y raw para enviar la petición):

```
{  
    "descripcion": "Vaso Tequilero",  
    "precio": 200.00,  
    "existencia": 30  
}
```

4. La respuesta debería ser el objeto recién guardado, incluyendo su ID generado:

```
{  
    "idProducto": 3, // Este valor depende de las pruebas hechas  
    "descripcion": "Vaso Tequilero",  
    "precio": 200.00,  
    "existencia": 30  
}
```

Ejecutamos la aplicación

<https://www.globalmentoring.com.mx>

- Ejecutamos la aplicación de Spring Boot: Click derecho y Run al archivo `InventariosApplication.java` (Backend)
- Ejecutamos la aplicación de Angular (FrontEnd) con el comando: `ng serve -o`
- Entramos a la URL: <http://localhost:4200/> y se refleja el nuevo producto agregado desde el BackEnd.

Id	Descripción del Producto	Precio	Existencias
1	Pantalon Rojo M	500	100
2	Blusa Amarilla L	300	50
3	Vaso Tequilero	200	30

Conclusión

- Se agregó el método `agregarProducto()` en el controlador para manejar solicitudes **POST**.
- Se modificó `guardarProducto()` en el servicio para retornar el producto recién guardado.
- Se verificó que Angular pueda recibir y mostrar el producto nuevo inmediatamente.
- Se probó la funcionalidad con Postman para asegurar que el backend funciona correctamente.

Con estos cambios, ahora la aplicación permite agregar productos desde Spring Boot (Backend) y verlos reflejados en tiempo real con Angular (FrontEnd).

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)