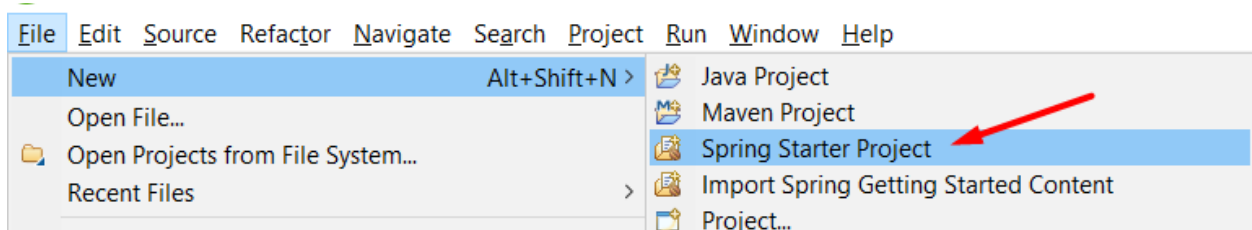


PROYECTO "swagger-rest"

Paso 1: Crear el Proyecto en STS

Abrir Spring Tool Suite (STS).

Ir a **File → New → Spring Starter Project**.



Completar los datos del proyecto:

- **Name:** swagger-api
- **Type:** Maven
- **Java Version:** 17
- **Group:** com.example
- **Artifact:** swagger-api
- **Packaging:** Jar

A screenshot of the 'New Spring Starter Project' dialog box in STS. The dialog contains several fields and options: 'Service URL' is set to 'https://start.spring.io'; 'Name' is 'swagger-api'; 'Use default location' is checked; 'Location' is 'C:\Users\JR\Documents\workspace\swagger-api' with a 'Browse' button; 'Type' is 'Maven', 'Packaging' is 'Jar', 'Java Version' is '17', and 'Language' is 'Java'; 'Group' is 'com.example', 'Artifact' is 'swagger-api', 'Version' is '0.0.1-SNAPSHOT', and 'Description' is 'Demo project for Spring Boot'; 'Package' is 'com.example.swaggerapi'. At the bottom, there is a 'Working sets' section with an unchecked 'Add project to working sets' checkbox, a 'New...' button, and a 'Working sets:' dropdown with a 'Select...' button.

- **Dependencies:**
 - **Spring Web** (Para la API REST)
 - **Spring Boot DevTools** (Para recarga automática)
 - **Springdoc OpenAPI** (Para la documentación de Swagger)
 - **Lombok** (Para reducir código repetitivo)

A screenshot of the 'Add Dependencies' dialog box in STS. It shows a list of dependencies with checkboxes: 'H2 Database' (unchecked), 'Spring Boot DevTools' (checked), 'Spring Web Services' (unchecked), 'Lombok' (checked), 'Spring Data JPA' (unchecked), 'PostgreSQL Driver' (unchecked), and 'Spring Web' (checked).

Hacer clic en **Finish** para generar el proyecto.

Springdoc OpenAPI detecta automáticamente los endpoints REST y genera la documentación.

Abrir pom.xml y agregar la dependencia de Springdoc OpenAPI:



```
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
  <version>2.3.0</version>
</dependency>
```

Configurar el título y descripción de la API en application.properties:



```
spring.application.name=swagger-api

springdoc.api-docs.path=/api-docs
springdoc.swagger-ui.path=/swagger-ui.html
```

Ahora, Swagger estará disponible en:

- **Documentación JSON:** <http://localhost:8080/api-docs>
- **Interfaz Swagger UI:** <http://localhost:8080/swagger-ui.html>

Paso 2: Crear la Entidad Producto

En `com.example.swaggerapi.model`, crear `Producto.java`:



```
package com.example.swaggerapi.model;

import io.swagger.v3.oas.annotations.media.Schema;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Schema(description = "Entidad que representa un producto en la tienda")
public class Producto {

    @Schema(description = "ID único del producto", example = "1")
    private Long id;

    @Schema(description = "Nombre del producto", example = "Laptop Dell XPS 15")
    private String nombre;

    @Schema(description = "Precio del producto en USD", example = "1499.99")
    private double precio;
}
```

Explicación:

- **@Schema** personaliza la descripción y ejemplos en Swagger.

Paso 3: Crear la Capa de Servicio

En `com.example.swaggerapi.service`, crear `ProductoService.java`:



```
package com.example.swaggerapi.service;

import com.example.swaggerapi.model.Producto;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

@Service
public class ProductoService {

    private final List<Producto> productos = new ArrayList<>();

    public List<Producto> obtenerTodos() {
        return productos;
    }

    public Optional<Producto> obtenerPorId(Long id) {
        return productos.stream().filter(p -> p.getId().equals(id)).findFirst();
    }

    public Producto guardar(Producto producto) {
        productos.add(producto);
        return producto;
    }

    public boolean eliminar(Long id) {
        return productos.removeIf(p -> p.getId().equals(id));
    }
}
```

Explicación:

- **obtenerTodos():** Retorna todos los productos.
- **obtenerPorId(Long id):** Busca un producto por ID.
- **guardar(Producto producto):** Guarda un nuevo producto.
- **eliminar(Long id):** Elimina un producto por ID.

Paso 4: Crear el Controlador y Personalizar la Documentación

En `com.example.swaggerapi.controller`, crear **ProductoController.java**:



```
package com.example.swaggerapi.controller;

import com.example.swaggerapi.model.Producto;
import com.example.swaggerapi.service.ProductoService;
import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.media.Content;
import io.swagger.v3.oas.annotations.media.Schema;
import io.swagger.v3.oas.annotations.responses.ApiResponse;
import io.swagger.v3.oas.annotations.responses.ApiResponses;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/productos")
public class ProductoController {

    private final ProductoService productoService;

    public ProductoController(ProductoService productoService) {
        this.productoService = productoService;
    }

    @Operation(summary = "Obtener todos los productos", description = "Retorna una lista de productos disponibles")
    @GetMapping
    public List<Producto> obtenerTodos() {
        return productoService.obtenerTodos();
    }

    @Operation(summary = "Obtener un producto por ID", description = "Busca un producto por su ID y lo devuelve si existe")
    @ApiResponses(value = {
        @ApiResponse(responseCode = "200", description = "Producto encontrado",
            content = @Content(mediaType = "application/json", schema =
                @Schema(implementation = Producto.class))),
        @ApiResponse(responseCode = "404", description = "Producto no encontrado")
    })
    @GetMapping("/{id}")
    public ResponseEntity<Producto> obtenerPorId(@PathVariable Long id) {
        return productoService.obtenerPorId(id)
            .map(ResponseEntity::ok)
            .orElse(ResponseEntity.notFound().build());
    }
}
```

```

    }

    @Operation(summary = "Crear un producto", description = "Guarda un nuevo producto en la
base de datos")
    @ApiResponse(responseCode = "201", description = "Producto creado exitosamente")
    @PostMapping
    public ResponseEntity<Producto> crearProducto(@RequestBody Producto producto) {
        return ResponseEntity.status(201).body(productoService.guardar(producto));
    }

    @Operation(summary = "Eliminar un producto", description = "Elimina un producto por ID si
existe")
    @ApiResponses(value = {
        @ApiResponse(responseCode = "204", description = "Producto eliminado exitosamente"),
        @ApiResponse(responseCode = "404", description = "Producto no encontrado")
    })
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> eliminarProducto(@PathVariable Long id) {
        return productoService.eliminar(id) ?
            ResponseEntity.noContent().build() :
            ResponseEntity.notFound().build();
    }
}

```

Explicación:

- **@Operation(summary, description):** Agrega título y descripción al endpoint.
- **@ApiResponse(responseCode, description):** Define respuestas personalizadas.
- **@ApiResponses({...}):** Lista varias respuestas posibles.

Paso 5: Ejecutar y Probar

Abrir en el navegador:

- **Interfaz Swagger UI:** <http://localhost:8080/swagger-ui.html>

localhost:8080/swagger-ui/index.html

Swagger
powered by SMARTBEAR

/api-docs

Explore

OpenAPI definition

v0 OAS 3.0

/api-docs

Servers

http://localhost:8080 - Generated server url

producto-controller

GET /api/productos Obtener todos los productos

POST /api/productos Crear un producto

GET /api/productos/{id} Obtener un producto por ID

DELETE /api/productos/{id} Eliminar un producto

- Documentación JSON: <http://localhost:8080/api-docs>

localhost:8080/api-docs

Impresión con formato estilístico

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "OpenAPI definition",
    "version": "v0",
    "servers": [
      {
        "url": "http://localhost:8080",
        "description": "Generated server url"
      }
    ],
    "paths": {
      "/api/productos": {
        "get": {
          "tags": [
            "producto-controller"
          ],
          "summary": "Obtener todos los productos",
          "description": "Retorna una lista de productos disponibles",
          "operationId": "obtenerTodos",
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "*/*": {
                  "schema": {
                    "type": "array",
                    "items": {
                      "$ref": "#/components/schemas/Producto"
                    }
                  }
                }
              }
            }
          },
          "post": {
            "tags": [
            "producto-controller"
            ],
            "summary": "Crear un producto",
            "description": "Guarda un nuevo producto en la base de datos",
            "operationId": "crearProducto",
            "requestBody": {
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Producto"
                  }
                }
              }
            },
            "responses": {
              "201": {
                "description": "Producto creado exitosamente",
                "content": {
                  "*/*": {
                    "schema": {
                      "$ref": "#/components/schemas/Producto"
                    }
                  }
                }
              },
              "404": {
                "description": "Producto no encontrado",
                "content": {
                  "*/*": {
                    "schema": {
                      "$ref": "#/components/schemas/Producto"
                    }
                  }
                }
              }
            }
          },
          "delete": {
            "tags": [
            "producto-controller"
            ],
            "summary": "Eliminar un producto",
            "description": "Elimina un producto por ID si existe",
            "operationId": "eliminarProducto",
            "parameters": [
              {
                "name": "id",
                "in": "path",
                "required": true,
                "schema": {
                  "type": "integer",
                  "format": "int64"
                }
              }
            ],
            "responses": {
              "200": {
                "description": "Producto encontrado",
                "content": {
                  "application/json": {
                    "schema": {
                      "$ref": "#/components/schemas/Producto"
                    }
                  }
                }
              },
              "404": {
                "description": "Producto no encontrado",
                "content": {
                  "*/*": {
                    "schema": {
                      "$ref": "#/components/schemas/Producto"
                    }
                  }
                }
              }
            }
          }
        }
      },
      "/api/productos/{id}": {
        "get": {
          "tags": [
            "producto-controller"
          ],
          "summary": "Obtener un producto por ID",
          "description": "Busca un producto por su ID y lo devuelve si existe",
          "operationId": "obtenerPorId",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int64"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "Producto encontrado",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Producto"
                  }
                }
              }
            },
            "404": {
              "description": "Producto no encontrado",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/Producto"
                  }
                }
              }
            }
          }
        },
        "delete": {
          "tags": [
            "producto-controller"
          ],
          "summary": "Eliminar un producto",
          "description": "Elimina un producto por ID si existe",
          "operationId": "eliminarProducto",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int64"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "Producto encontrado",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Producto"
                  }
                }
              }
            },
            "404": {
              "description": "Producto no encontrado",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/Producto"
                  }
                }
              }
            }
          }
        }
      }
    }
  },
  "components": {
    "schemas": {
      "Producto": {
        "type": "object",
        "properties": {
          "id": {
            "type": "integer",
            "format": "int64",
            "example": 1,
            "description": "ID único del producto"
          },
          "nombre": {
            "type": "string",
            "description": "Nombre del producto",
            "example": "Laptop Dell XPS 15"
          },
          "precio": {
            "type": "number",
            "description": "Precio del producto en USD",
            "format": "double",
            "example": 1499.99
          }
        },
        "description": "Entidad que representa un producto en la tienda"
      }
    }
  }
}
```