Moises Joaquin Mestas Maque

Crear proyecto: ms-catalogo

# **Preparación de ambiente de desarrollo**

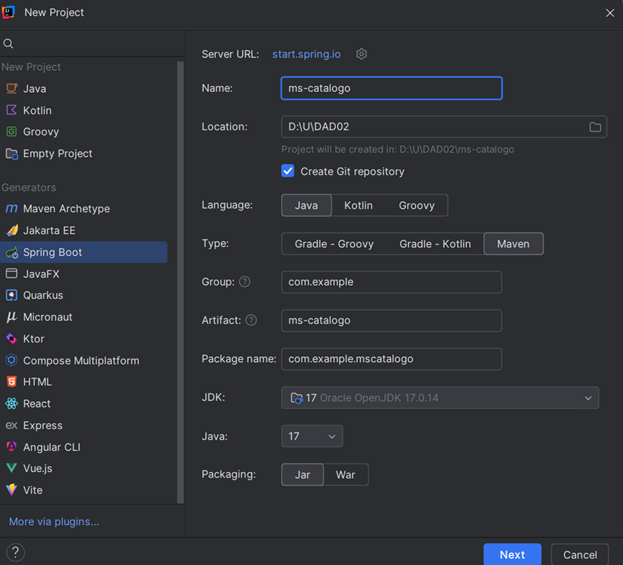
**1.** **Instalación de Laragon**

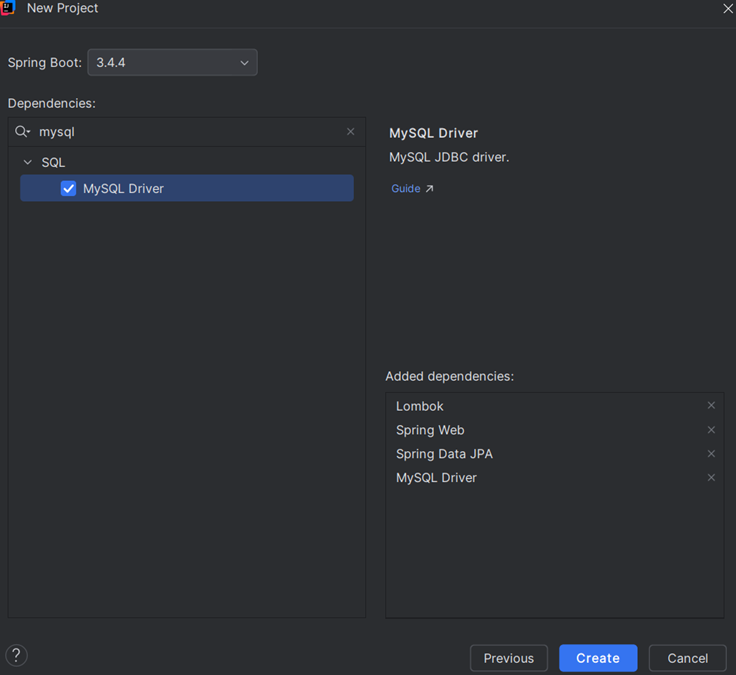
**2.** **Instalación y configuración de Git**

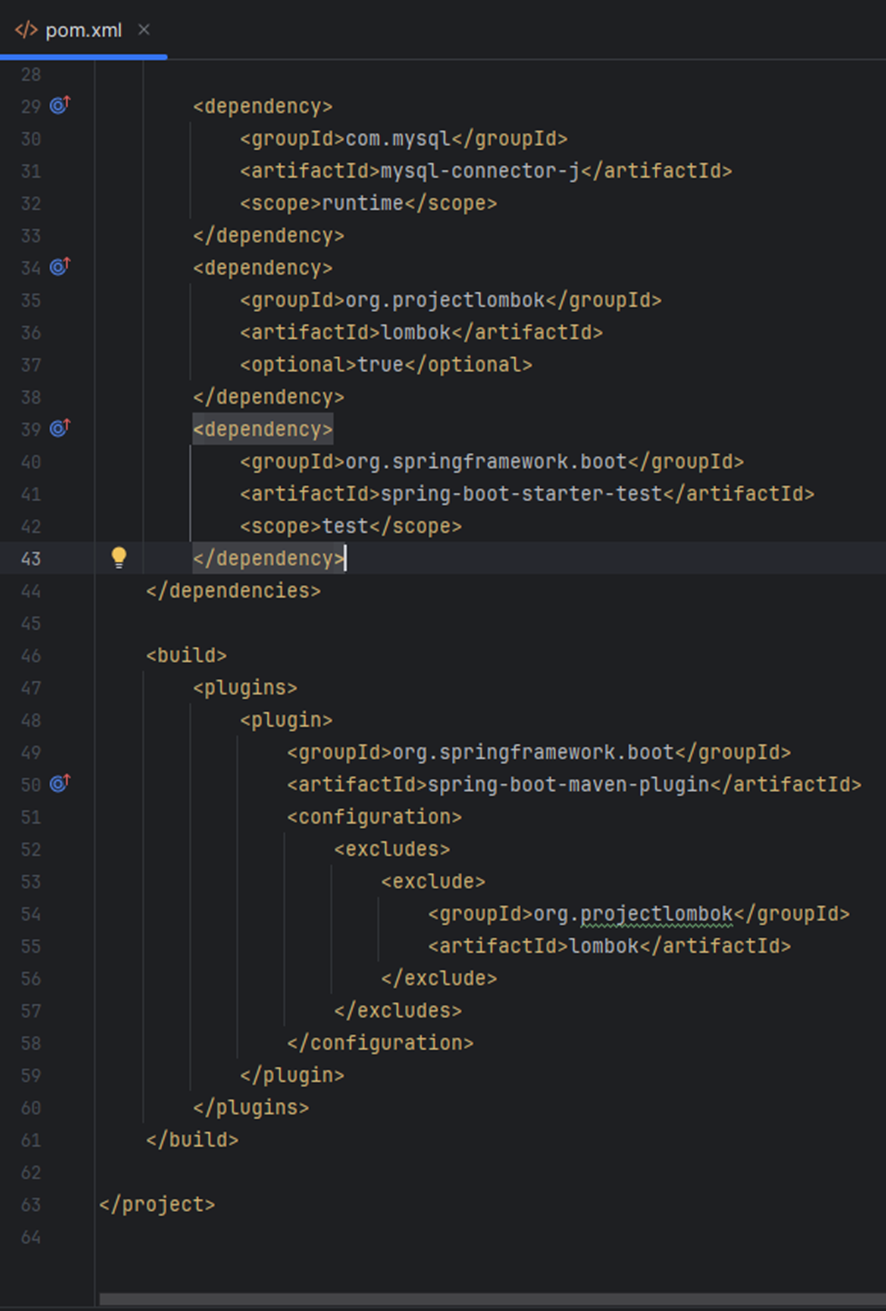
3. **Instalación de Java 17 o 21**

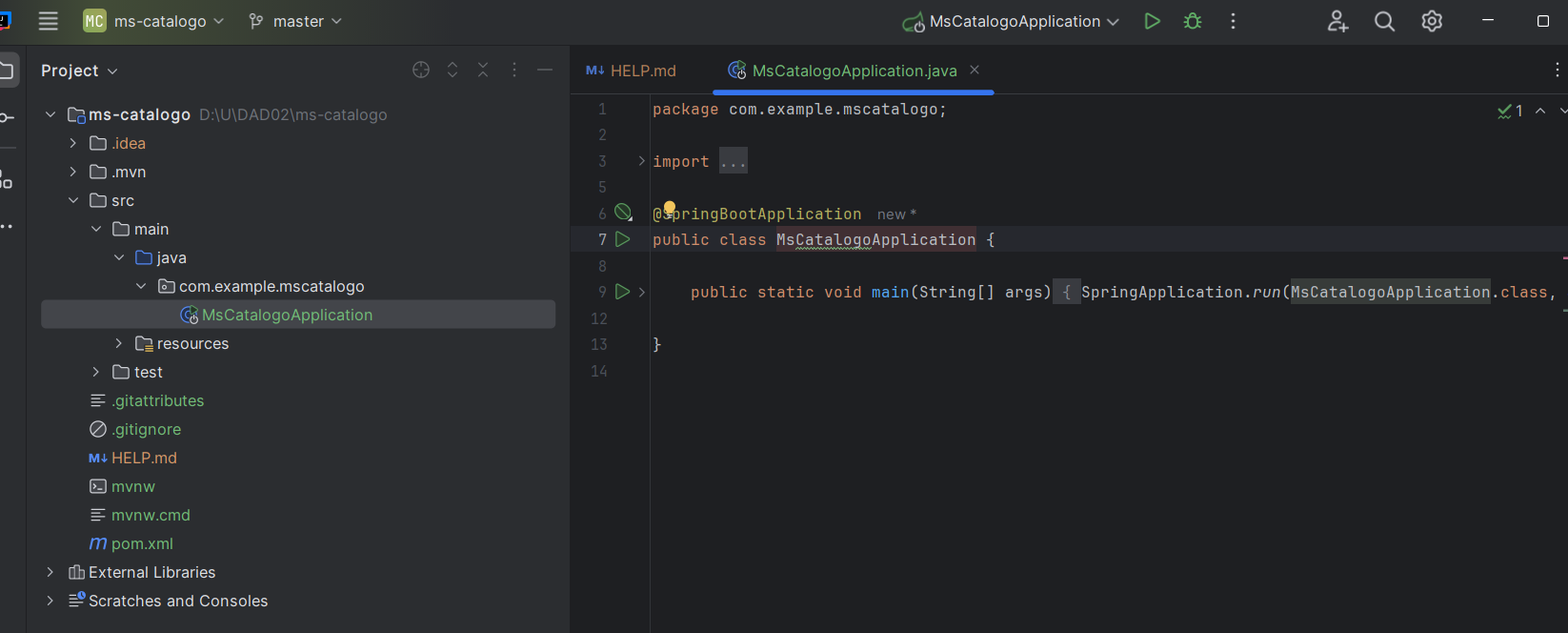
4. **Instalación de Intellij idea**

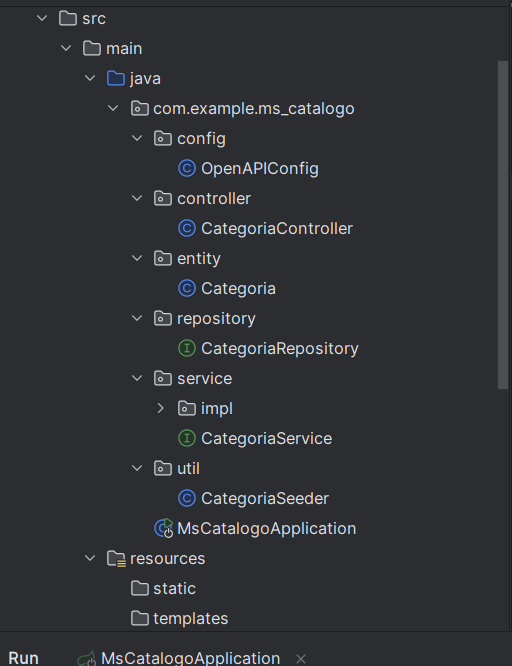
**5.** **Instalación de HTTPie/Postman**



ESCOGER LIBRERÍA PARA DESARROLLAR UN CRUD API REST 

Comprobar las dependencias antes de iniciar en el POM.xml



Crear los respectivos packages 

config - OpenAPICoing.java

package com.example.ms\_catalogo.config;

import io.swagger.v3.oas.models.info.Info;

import io.swagger.v3.oas.models.info.License;

import io.swagger.v3.oas.models.OpenAPI;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class OpenAPIConfig {

@Bean

public OpenAPI customOpenAPI() {

return new OpenAPI().info(new Info()

.title("OPEN API MICROSERVICIO CATÁLOGO")

.version("0.0.1")

.description("Servicios web catálogo")

.termsOfService("http://swagger.io/terms")

.license(new License().name("Apache 2.0").url("http://springdoc.org"))

);

}

}

controller - CategoriaController

package com.example.ms\_catalogo.controller;

import com.example.ms\_catalogo.entity.Categoria;

import com.example.ms\_catalogo.service.CategoriaService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/categorias")

public class CategoriaController {

@Autowired

private CategoriaService categoriaService;

@RequestMapping

public List<Categoria> listar() {

return categoriaService.listar();

}

@RequestMapping("/{id}")

public Optional<Categoria> buscar(@PathVariable Integer id) {

return categoriaService.buscar(id);

}

@PostMapping

public Categoria guardar(@RequestBody Categoria categoria) {

System.*out*.println(categoria);

return categoriaService.guardar(categoria);

}

@PutMapping("/{id}")

public Categoria actualizar(@PathVariable Integer id, @RequestBody Categoria categoria) {

return categoriaService.actualizar(id, categoria);

}

@DeleteMapping("/{id}")

public void eliminar(@PathVariable Integer id) {

categoriaService.eliminar(id);

}

}

entity-Categoria

package com.example.ms\_catalogo.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import lombok.Data;

@Entity

@Data

public class Categoria {

@Id

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

private Integer id;

private String nombre;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public Categoria(Integer id, String nombre) {

this.id = id;

this.nombre = nombre;

}

public Categoria() {

}

@Override

public String toString() {

return "Categoria{" +

"id=" + id +

", nombre='" + nombre + '\'' +

'}';

}

}

repository - CategoriaRepository

package com.example.ms\_catalogo.repository;

import com.example.ms\_catalogo.entity.Categoria;

import org.springframework.data.jpa.repository.JpaRepository;

public interface CategoriaRepository extends JpaRepository<Categoria, Integer> {

}

service -impl (CategoriaServiceImpl)

package com.example.ms\_catalogo.service.impl;

import com.example.ms\_catalogo.entity.Categoria;

import com.example.ms\_catalogo.repository.CategoriaRepository;

import com.example.ms\_catalogo.service.CategoriaService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class CategoriaServiceImpl implements CategoriaService {

@Autowired

private CategoriaRepository categoriaRepository;

@Override

public List<Categoria> listar() {

return categoriaRepository.findAll();

}

@Override

public Optional<Categoria> buscar(Integer id) {

return categoriaRepository.findById(id);

}

@Override

public Categoria guardar(Categoria categoria) {

return categoriaRepository.save(categoria);

}

@Override

public Categoria actualizar(Integer id, Categoria categoria) {

categoria.setId(id);

return categoriaRepository.save(categoria);

}

@Override

public void eliminar(Integer id) {

categoriaRepository.deleteById(id);

}

}

CategoriaService

package com.example.ms\_catalogo.service;

import com.example.ms\_catalogo.entity.Categoria;

import java.util.List;

import java.util.Optional;

public interface CategoriaService {

List<Categoria> listar();

Optional<Categoria> buscar(Integer id);

Categoria guardar(Categoria categoria);

Categoria actualizar(Integer id, Categoria categoria);

void eliminar(Integer id);

}

util -CategoriaSeeder (Datos sin laragon )

package com.example.ms\_catalogo.util;

import com.example.ms\_catalogo.entity.Categoria;

import com.example.ms\_catalogo.repository.CategoriaRepository;

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

@Component

public class CategoriaSeeder implements CommandLineRunner {

private final CategoriaRepository categoriaRepository;

public CategoriaSeeder(CategoriaRepository categoriaRepository) {

this.categoriaRepository = categoriaRepository;

}

@Override

public void run(String... args) {

// Verificamos si ya existen datos para no duplicar

if (categoriaRepository.count() == 0) {

Categoria cat1 = new Categoria(null, "Electrónica");

Categoria cat2 = new Categoria(null, "Ropa");

Categoria cat3 = new Categoria(null, "Hogar");

Categoria cat4 = new Categoria(null, "Juguetes");

Categoria cat5 = new Categoria(null, "Libros");

Categoria cat6 = new Categoria(null, "Deportes");

Categoria cat7 = new Categoria(null, "Salud");

Categoria cat8 = new Categoria(null, "Belleza");

Categoria cat9 = new Categoria(null, "Automotriz");

Categoria cat10 = new Categoria(null, "Mascotas");

Categoria cat11 = new Categoria(null, "Herramientas");

Categoria cat12 = new Categoria(null, "Computadoras");

Categoria cat13 = new Categoria(null, "Celulares");

Categoria cat14 = new Categoria(null, "Música");

Categoria cat15 = new Categoria(null, "Jardín");

Categoria cat16 = new Categoria(null, "Alimentos");

Categoria cat17 = new Categoria(null, "Bebidas");

Categoria cat18 = new Categoria(null, "Electrodomésticos");

Categoria cat19 = new Categoria(null, "Joyería");

Categoria cat20 = new Categoria(null, "Videojuegos");

categoriaRepository.save(cat1);

categoriaRepository.save(cat2);

categoriaRepository.save(cat3);

categoriaRepository.save(cat4);

categoriaRepository.save(cat5);

categoriaRepository.save(cat6);

categoriaRepository.save(cat7);

categoriaRepository.save(cat8);

categoriaRepository.save(cat9);

categoriaRepository.save(cat10);

categoriaRepository.save(cat11);

categoriaRepository.save(cat12);

categoriaRepository.save(cat13);

categoriaRepository.save(cat14);

categoriaRepository.save(cat15);

categoriaRepository.save(cat16);

categoriaRepository.save(cat17);

categoriaRepository.save(cat18);

categoriaRepository.save(cat19);

categoriaRepository.save(cat20);

System.*out*.println("Datos de categorías insertados correctamente.");

} else {

System.*out*.println("Las categorías ya existen, no se insertaron datos.");

}

}

}

cambiar el application.properties a Application.yml

spring:

application:

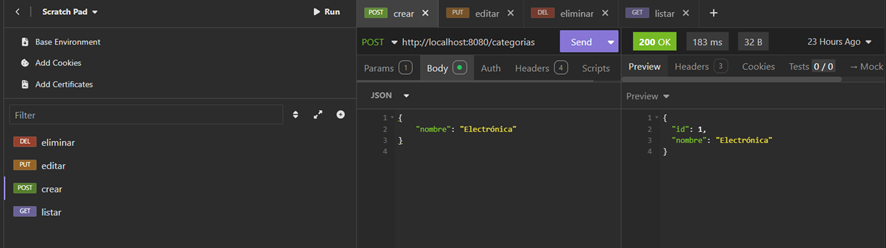
name: ms-catalogo-service

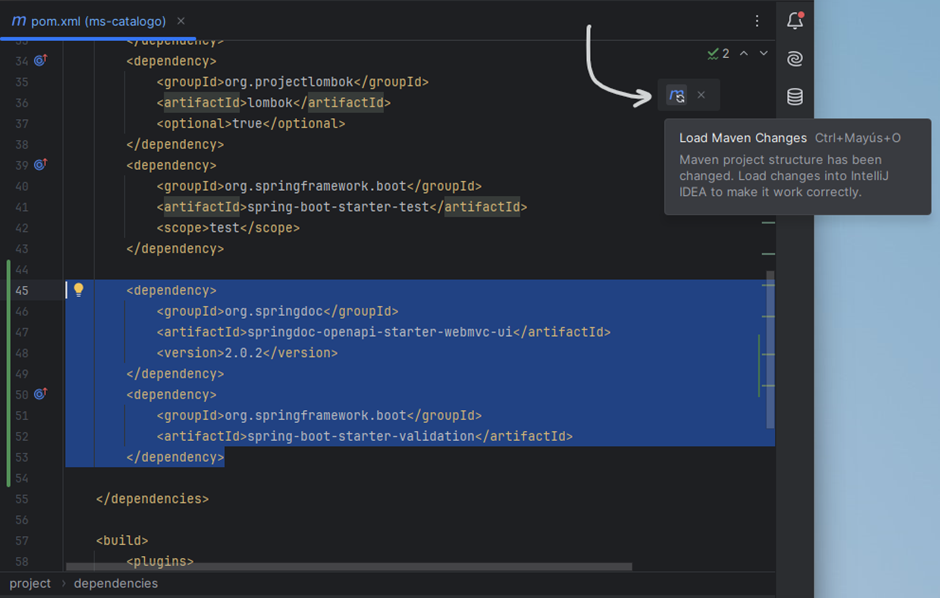
profiles:

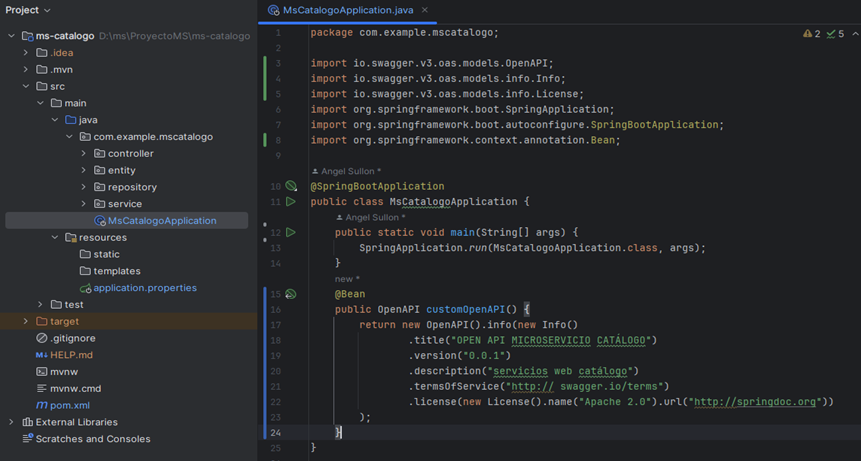
active: development

config:

import: optional:configserver:http://root:123456@localhost:7070

Testeo en insomnia

Ahora, debes actualizar la descarga de las librerías

Configurar Bean: MsCatalogoApplication

package com.example.mscatalogo;

import io.swagger.v3.oas.models.OpenAPI;

import io.swagger.v3.oas.models.info.Info;

import io.swagger.v3.oas.models.info.License;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

@SpringBootApplication

public class MsCatalogoApplication {

public static void main(String[] args) {

SpringApplication.*run*(MsCatalogoApplication.class, args);

}

@Bean

public OpenAPI customOpenAPI() {

return new OpenAPI().info(new Info()

.title("OPEN API MICROSERVICIO CATÁLOGO")

.version("0.0.1")

.description("servicios web catálogo")

.termsOfService("http:// swagger.io/terms")

.license(new License().name("Apache 2.0").url("http://springdoc.org"))

);

}

}

