BACKEND – SPRINGBOOT

Comenzaremos creando proyecto en SpringBoot para diseñar la gestión de la base de datos.

Versión de spring: 3.5.6
JDK: 21 o superior
Agregar dependencias:

- Spring web
- Spring data jpa
- Spring devtools
- MySql Driver
- Lombok

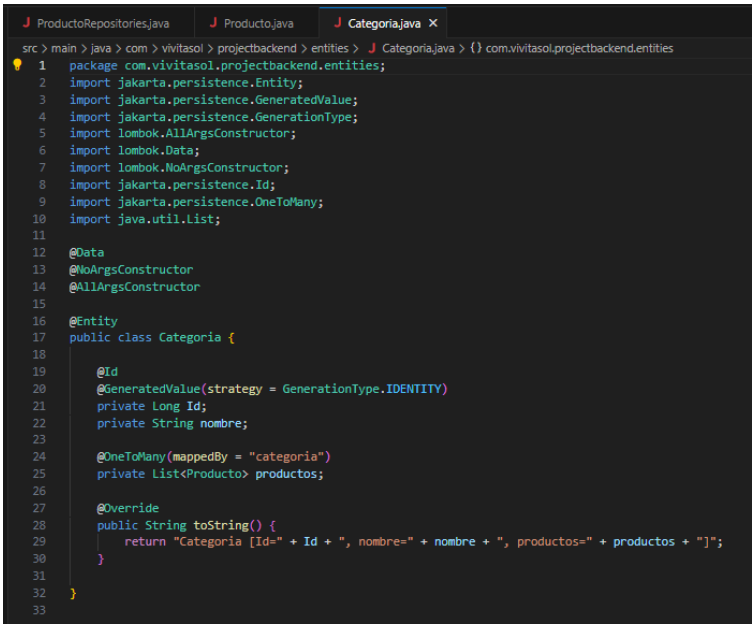
Crear esquema de base de datos en WorkBench: base
Configurar conexión en application.properties

```
# parámetros de la conexión
spring.datasource.url=jdbc:mysql://localhost:3306/base?serverTimezone=UTC&useSSL=false
spring.datasource.username=root
spring.datasource.password=sasa

# Configuración de JPA
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

# Configuración de logs (opcional)
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
```

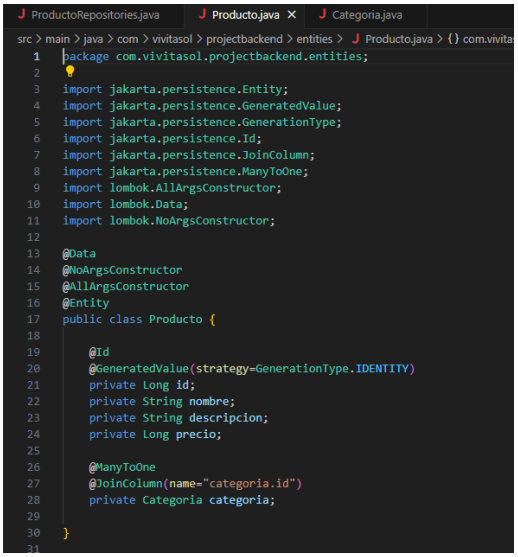
Configuramos la estructura del proyecto a través de los packages: entities, repository, services, restcontrollers

ENTIDADES:

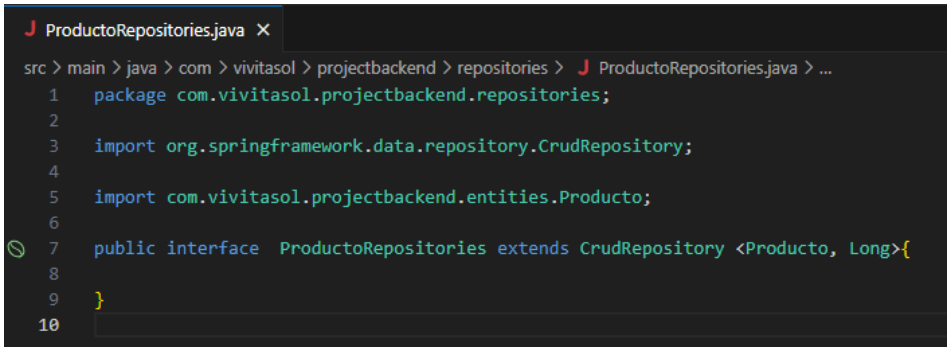
Comenzamos por la creación de las clases Categoría, Producto:

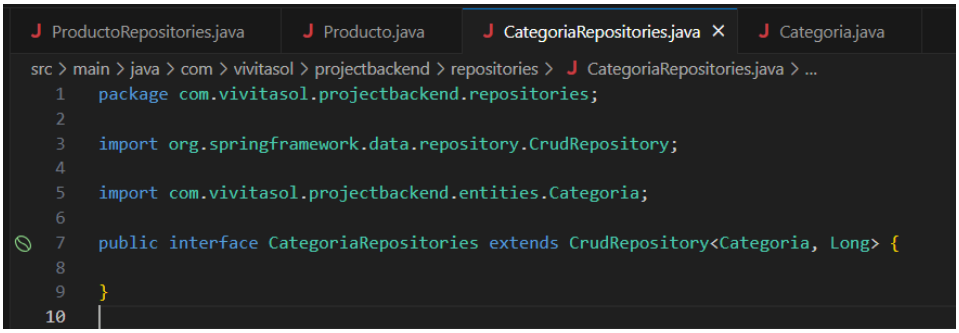


Clase Producto:



REPOSITORIOS:





SERVICIOS

Comenzamos con el servicio de categoría: implementamos interface y clase;
Interface CategoriaServices

Producto.java | CategoriaRepositories.java | CategoriaServices.java

main > java > com > vivitasol > projectbackend > services > CategoriaServices.java > ...

```java
package com.vivitasol.projectbackend.services;

import com.vivitasol.projectbackend.entities.Categoria;
import java.util.List;

public interface CategoriaServices {

    Categoria crear(Categoria categoria);
    Categoria obtenerId(Long id);
    List<Categoria> listarTodas();
    void eliminar(Long id);
    Categoria actualizar(Long id, Categoria categoriaActualizada);

}
```

Clase CategoriaServicesImpl:

CategoriaServicesImpl.java

src > main > java > com > vivitasol > projectbackend > services > CategoriaServicesImpl.java > Language Support for Java

```java
@Service
public class CategoriaServicesImpl implements CategoriaServices {

    @Autowired
    private CategoriaRepositories categoriaRepositories;

    @Override
    public Categoria crear(Categoria categoria){
        return categoriaRepositories.save(categoria);
    }


    @Override
    public Categoria obtenerId(Long id) {
        return categoriaRepositories.findById(id)
                .orElseThrow(() -> new RuntimeException("Categoría no encontrada"));
    }

    @Override
    public List<Categoria> listarTodas() {
        return (List<Categoria>) categoriaRepositories.findAll();
    }

```

CategoriaServicesImpl.java

src > main > java > com > vivitasol > projectbackend > services > CategoriaServicesImpl.java > Language Sup

```java
public class CategoriaServicesImpl implements CategoriaServices {

    @Override
    public List<Categoria> listarTodas() {
        return (List<Categoria>) categoriaRepositories.findAll();
    }

    @Override
    public void eliminar(Long id) {
        if (!categoriaRepositories.existsById(id)) {
            throw new RuntimeException("Categoría no encontrada");
        }
        categoriaRepositories.deleteById(id);
    }

    @Override
    public Categoria actualizar(Long id, Categoria categoriaActualizada) {
        Categoria existente = obtenerId(id);
        existente.setNombre(categoriaActualizada.getNombre());
        return categoriaRepositories.save(existente);
    }


}
```

Nota: implementar servicio para Productos

RESTCONTROLLERS

```java
@RestController
@RequestMapping("/api/categorias")
public class CategoriaRestControllers {

    @Autowired
    private CategoriaServices categoriaServices;

    @PostMapping
    public ResponseEntity<Categoria> crearCategoria(@RequestBody Categoria categoria) {
        Categoria nuevaCategoria = categoriaServices.crear(categoria);
        return ResponseEntity.ok(nuevaCategoria);
    }

    @GetMapping("/{id}")
    public ResponseEntity<Categoria> obtenerCategoriaPorId(@PathVariable Long id) {
        Categoria categoria = categoriaServices.obtenerId(id);
        return ResponseEntity.ok(categoria);
    }


    @GetMapping
    public ResponseEntity<List<Categoria>> listarCategorias() {
        List<Categoria> categorias = categoriaServices.listarTodas();
        return ResponseEntity.ok(categorias);
    }
```

```java
    @GetMapping
    public ResponseEntity<List<Categoria>> listarCategorias() {
        List<Categoria> categorias = categoriaServices.listarTodas();
        return ResponseEntity.ok(categorias);
    }


    @DeleteMapping("/{id}")
    public ResponseEntity<Void> eliminarCategoria(@PathVariable Long id) {
        categoriaServices.eliminar(id);
        return ResponseEntity.noContent().build();
    }


    @PutMapping("/{id}")
    public ResponseEntity<Categoria> actualizarCategoria(@PathVariable Long id, @RequestBody Categoria categoriaActualizada) {
        Categoria categoria = categoriaServices.actualizar(id, categoriaActualizada);
        return ResponseEntity.ok(categoria);
    }
}
```

Implementar servicios y controller para Productos