Desarrollo web en entorno servidor

Plataforma Digital: NetGonVideo

Utilizando Spring Tool y Postman

I.E.S. Jacarandá 2ºDAW

Gonzalo García Mateos

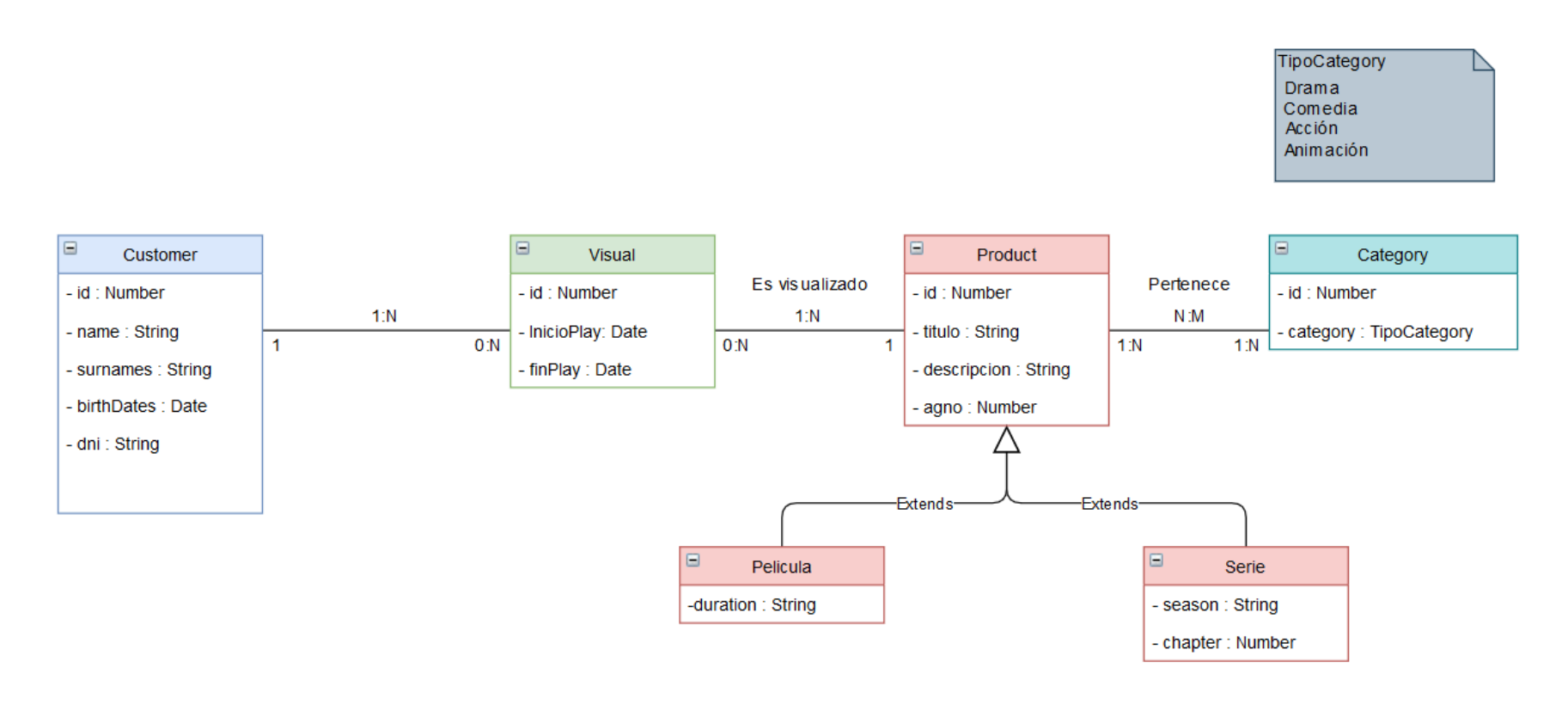
1.- Introducción a la propuesta

NetGonVideo

- Es una plataforma digital que permite a los clientes visualizar series y películas.
- Cada uno de estos productos tiene una o varias categorías.
- He planteado dividirlo en una serie de Entidades: Cliente, Producto, Película, Serie, Categoría y Visual.

2.- UML para el diagrama de clases

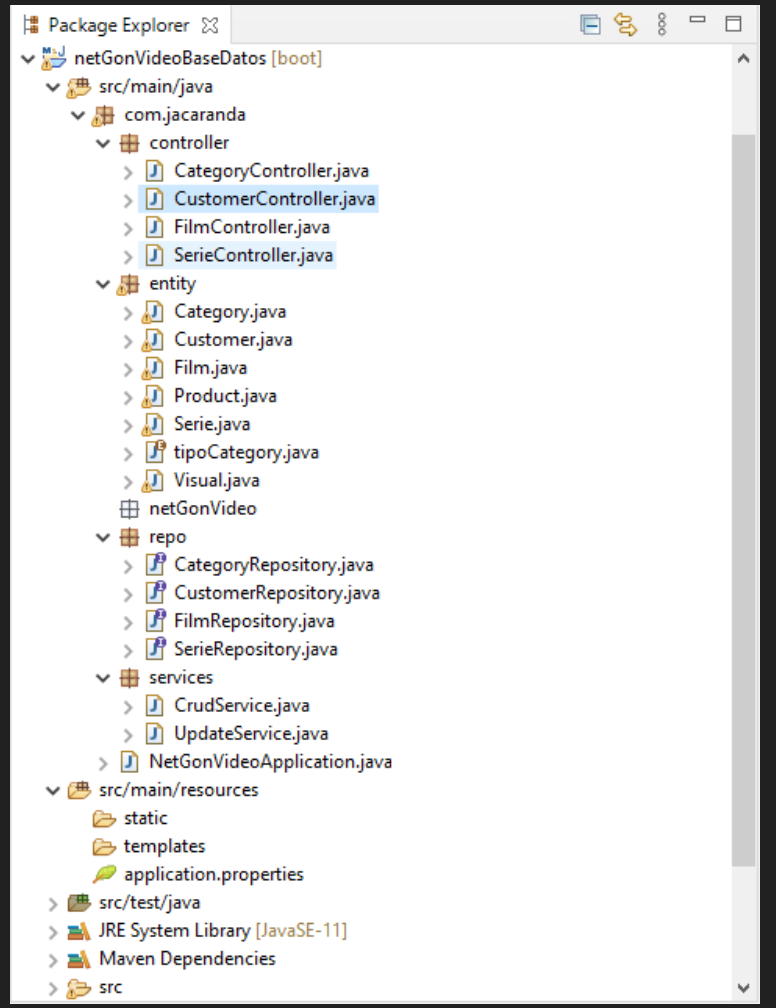
NetGonVideo UML



3.- Estructura de NetGonVideo

Paquetes:

- Controller → Todos los controladores.
- Entity → Todas las entidades.
- netGonVideo → ScanBasePackages (controller y servicies).
- Repo → Todas interfaces CrudRepository
- Services → Todos los servicios.



3.- Estructura de NetGonVideo

Entity:

Customer

```java
package com.jacaranda.entity;


import java.io.Serializable;

@Entity
public class Customer implements Comparable<Customer>,Serializable {
    //Variables

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;

    private String name;
    private String surnames;

    @NonNull
    @DateTimeFormat(pattern = "ddMMyyyy")
    private LocalDate birthDate;

    private String dni;

    // Relaciones con la entidad Visual
    @OneToMany(mappedBy="customer")
    private List<Visual> visualizaciones;


    //Constructores
    public Customer() {
        super();
        visualizaciones = new ArrayList<Visual>();
    }


    public Customer(String name, String surnames, LocalDate birthDate, String dni) {
        super();
        this.name = name;
        this.surnames = surnames;
        this.birthDate = birthDate;
        this.dni = dni;
        visualizaciones = new ArrayList<Visual>();
    }

//GET y SET

    public long getId() {
        return id;
    }

    public String getName() {
        return name;
```

```java

    public void setBirthDate(LocalDate birthDate) {
        this.birthDate = birthDate;
    }


    public String getDni() {
        return dni;
    }



    public List<Visual> getVisualizaciones() {
        return visualizaciones;
    }

    //ComparaTo
    @Override
    public int compareTo(Customer other) {

        return Integer.valueOf(getId()).compareTo(getId());
    }

}
```

3.- Estructura de NetGonVideo

Entity:

Product

```java
package com.jacaranda.entity;

import java.io.Serializable;

@MappedSuperclass
public class Product implements Serializable{
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;
    private String title;
    private String description;
    private int agno;

    // Relación con la entidad Categoría
    @ManyToOne(cascade=CascadeType.ALL)
    @JoinColumn(name="categorias_id", foreignKey = @ForeignKey(name="categorias_id_fk"), nullable = false)

    //Objeto categoría para la relación
    private Category categorias;


    //Constructores
    public Product() {
        super();

    }

    public Product(String title, String description, int agno,Category categorias) {
        super();
        this.title = title;
        this.description = description;
        this.agno = agno;
        this.categorias = categorias;

    }

    //GET Y SET

    public String getTitle() {
        return title;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public int getAgno() {
```

```java
public void setAgno(int agno) {
    this.agno = agno;
}

public void setId(int id) {
    this.id = id;
}
public long getId() {
    return id;
}

public Category getCategorias() {
    return categorias;
}
```

4.- Controladores de NetGonVideo

Realizamos el Servicio para el crud.

En rutamos los Métodos:

Get

Post

Put

Delete

```java
package com.jacaranda.controller;


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

@RestController
@RequestMapping(path = "/api")

public class CustomerController {

    // Creamos el servicio
    @Autowired
    private CrudService crudService;

    // Método GET
    @GetMapping("/customers")
    public ResponseEntity<?> getCustomer() {
        return crudService.getCustomer();
    }

    // Método Get por parámetro id
    @GetMapping("/customers/{id}")
    public ResponseEntity<?> getCustomerId(@PathVariable long id) {
        return crudService.getCustomerId(id);
    }

    // Método POST
    @PostMapping("/customers")
    public ResponseEntity<?> createCustomer(@RequestBody Customer sent) {
        return crudService.createCustomer(sent);
    }

    // Método PUT
    @PutMapping("/customers")
    public ResponseEntity<?> updateCustomer(@RequestBody Customer sent) {
        return crudService.updateCustomer(sent);
    }

    // Método DELETE
    @DeleteMapping("/customers/{id}")
    public ResponseEntity<?> deleteCustomer(@PathVariable long id) {
        return crudService.deleteCustomer(id);
    }

}
```

Desarrollo web en entorno servidor

5.- Servicios de NetGonVideo

CrudService:

Realizamos los cruds de las entidades.

UpDateService

Servicio para actualizar las entidades en el crud

```java
package com.jacaranda.services;

import java.util.Collection;

@Service
public class CrudService {

    // Añadiendo los repositorios
    @Autowired
    private CrudRepository<Customer, Long> crudRepositoryCustomer;

//  @Autowired
//  private CrudRepository<Product, Long> crudRepositoryProduct;

    @Autowired
    private CrudRepository<Category, Long> crudRepositoryCategory;

    @Autowired
    private CrudRepository<Film, Long> crudRepositoryFilm;

    @Autowired
    private CrudRepository<Serie, Long> crudRepositorySerie;

    @Autowired
    private CustomerRepository customerRepository;

//  @Autowired
//  private ProductRepository productRepository;

    @Autowired
    private CategoryRepository categoryRepository;

    @Autowired
    private FilmRepository filmRepository;

    @Autowired
    private SerieRepository serieRepository;

    //Añadiendo Servicios
    @Autowired
    private UpdateService updateService;

    // ENTIDAD CUSTOMER

    // Método Get Customer
    public ResponseEntity<?>getCustomer(){
        Iterable<Customer> resultado = crudRepositoryCustomer.findAll();
        ResponseEntity<?> response;
        if(((Collection<?>)resultado).size() == 0) {
            response= ResponseEntity
                      .status(HttpStatus.NOT_FOUND)
```

```java
// ENTIDAD CUSTOMER

// Método Get Customer
public ResponseEntity<?>getCustomer(){
    Iterable<Customer> resultado = crudRepositoryCustomer.findAll();
    ResponseEntity<?> response;
    if(((Collection<?>)resultado).size() == 0) {
        response= ResponseEntity
                  .status(HttpStatus.NOT_FOUND)
                  .body("La base de datos está vacía");
    }else {
         response= ResponseEntity
                  .status(HttpStatus.OK)
                  .body(crudRepositoryCustomer.findAll());
    }
    return response;
}

// Método Get con parámetro id
public ResponseEntity<?> getCustomerId(@PathVariable long id){
    Customer customer = customerRepository.findCustomerById(id);
    ResponseEntity<?> response;
    if(customer==null) {
        response = ResponseEntity
                  .status(HttpStatus.NOT_FOUND)
                  .body("El cliente no existe");
    }else {
        response = ResponseEntity
                  .status(HttpStatus.OK)
                  .body(crudRepositoryCustomer.findById(id));
    }
    return response;
}
```

6.- Repo de NetGonVideo

Interfaces donde va la query y métodos de algunas entidades.

CategoryRepository

CustomerRepository

FilmRepository

SerieRepository

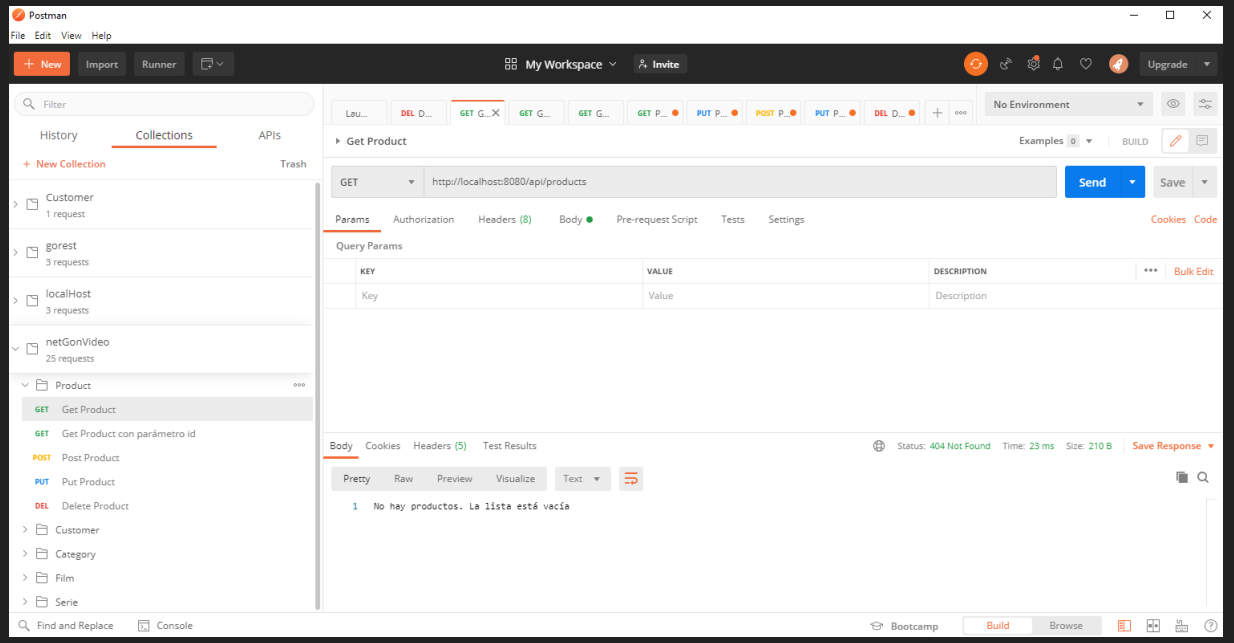
CustomerRepository.java

```java
package com.jacaranda.repo;

import java.util.List;

@Repository
public interface CustomerRepository extends CrudRepository<Customer, Long>{

    @Query(value="select * from Customer order by name", nativeQuery = true)
    public List<Customer> findAllOrdenedByName();

    public Customer findCustomerByDni(String dni);

    public Customer findCustomerById(Long id);

}
```

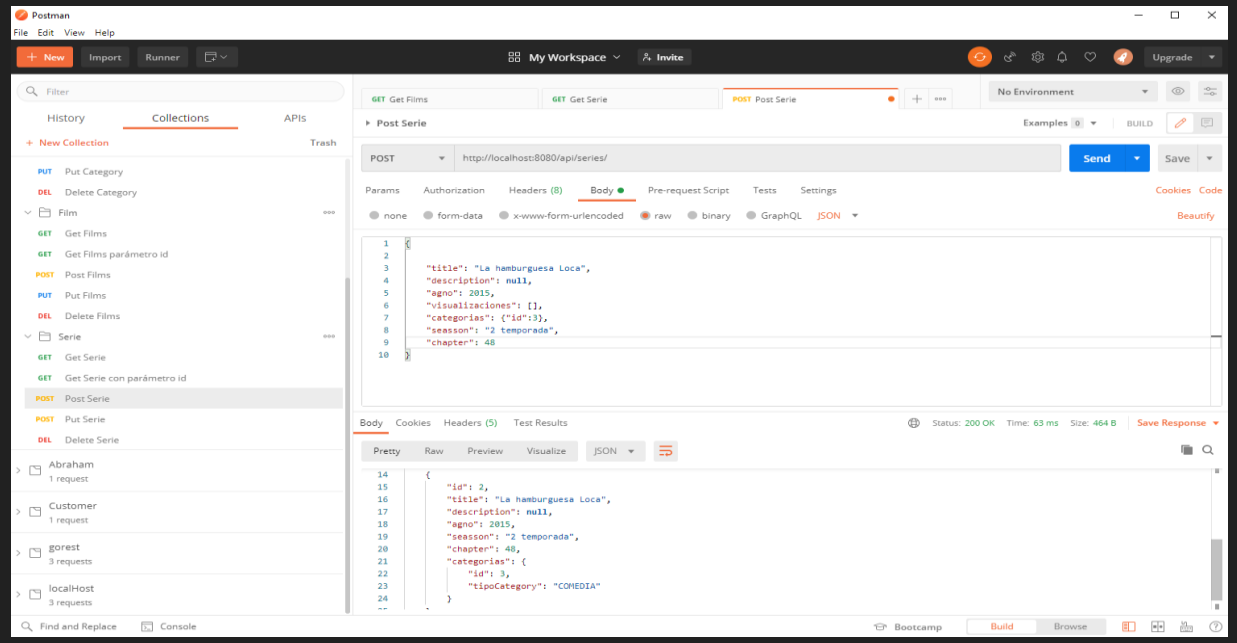
Desarrollo web en entorno servidor

7.- Peticiones de NetGonVideo con Postman

GET



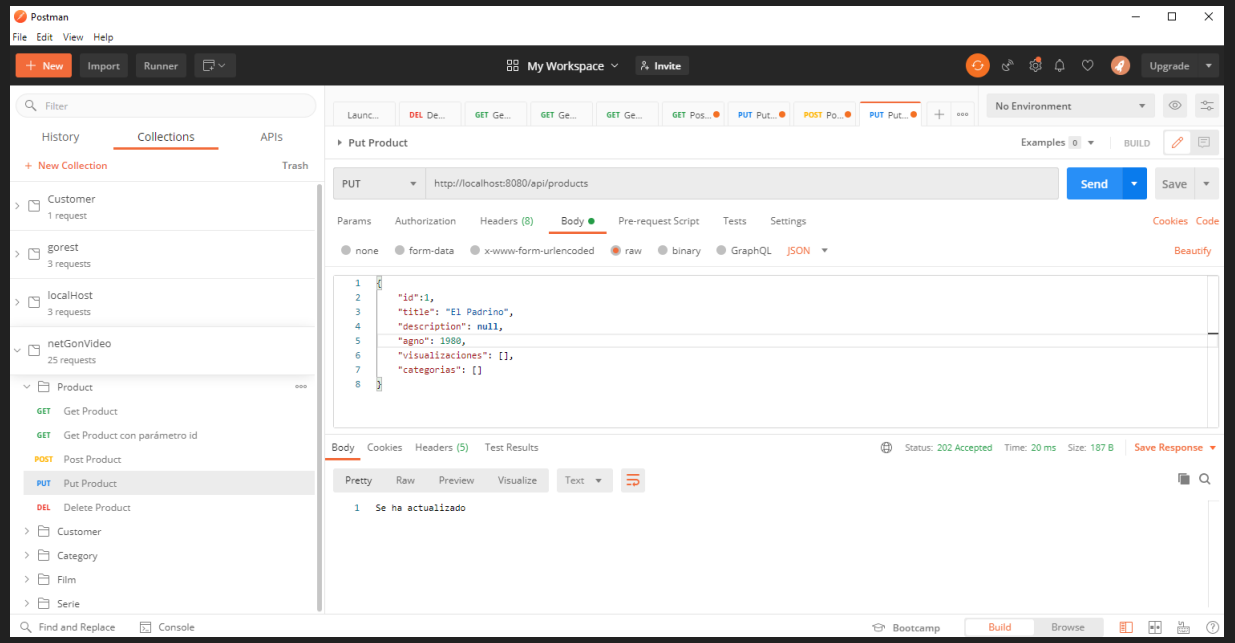
POST



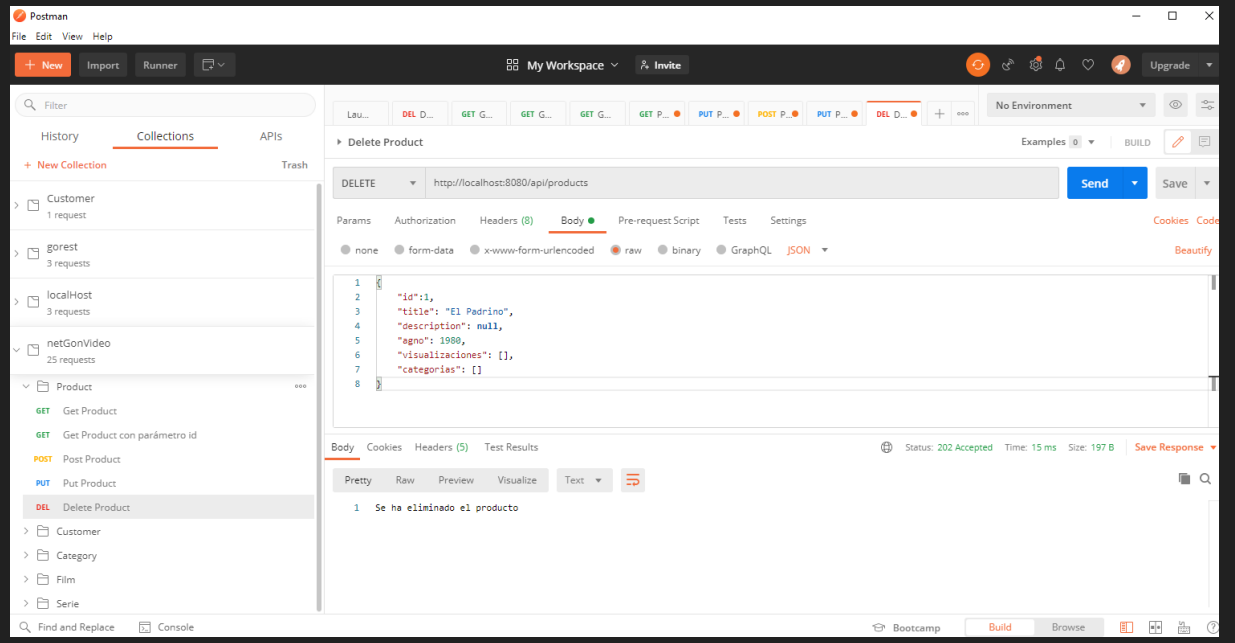
Desarrollo web en entorno servidor

8.- Peticiones de NetGonVideo con Postman

PUT



DELETE



6.- Dificultades de NetGonVideo encontradas

- Problema no reconoce a Product como entidad al llevar la anotación @MappedSuperclass.

  Solución: No hacer crud de la entidad Product y duplicar las relaciones de sus clases hijas (Film y Películas con la entidad Category)

- Problema al crear el método Post tanto de film como de Serie me creaba la categoría sin comprobar si existía.

  Solución: Crear la comprobación en dicho método por el id de categoría y el título de la serie o película.

9.- Propuestas de mejoras de NetGonVideo

- Crear una Query parametrizada.

Desarrollo web en entorno servidor

GRACIAS

https://github.com/Kurtgon/netGonVideo

Gonzalo García Mateos