

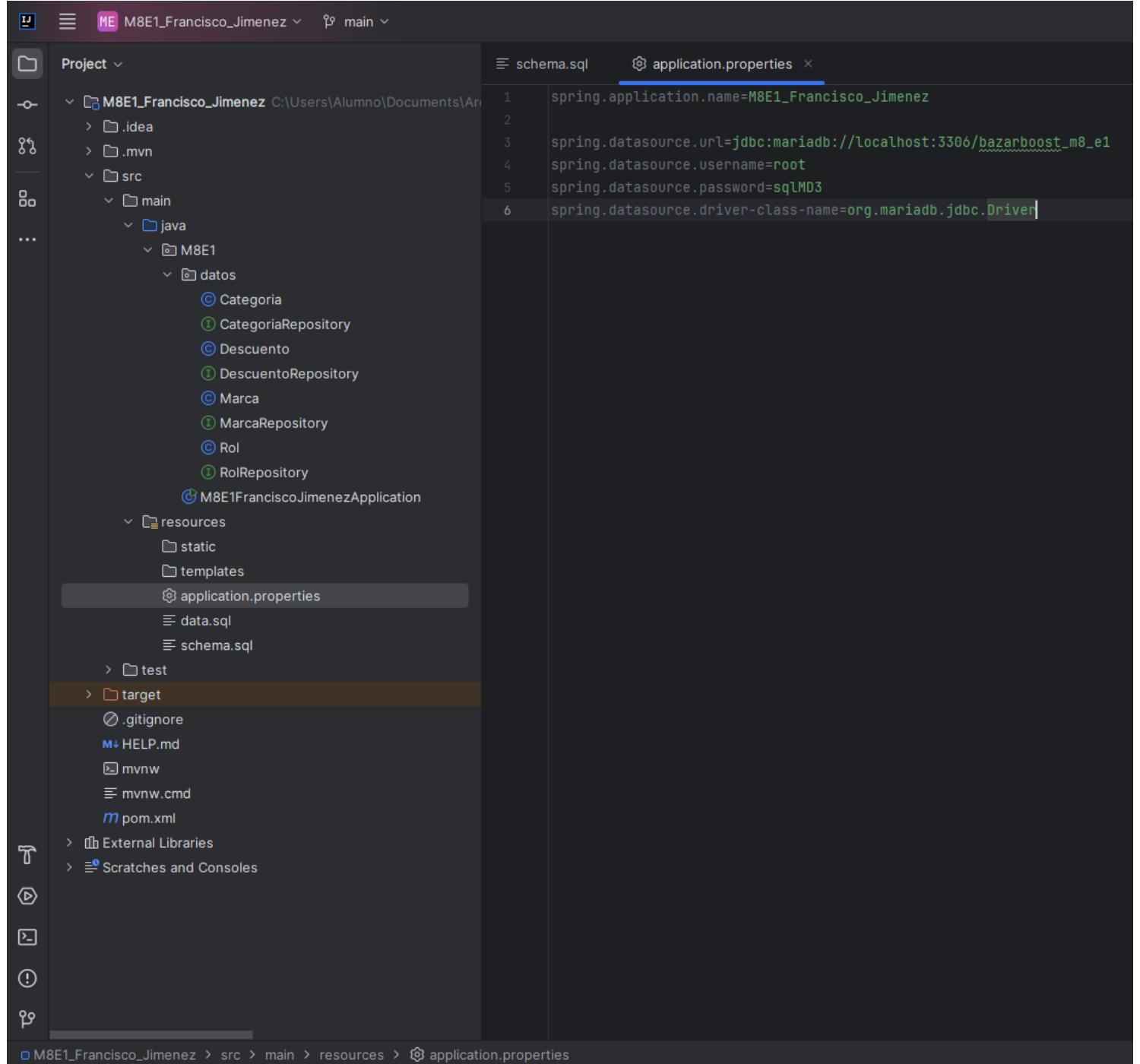
Francisco Williams Jiménez Hernández

Ejercicio 1

1. Configuración inicial

Para la realización de este ejercicio, lo primero que hice fue configurar la conexión a la base de datos desde el archivo **application.properties** como se muestra en la siguiente captura.

Archivo: application.properties



The screenshot shows a Java IDE interface with a dark theme. On the left, the project structure is displayed under the 'Project' tab, showing a package named 'M8E1_Francisco_Jimenez' containing subfolders '.idea', '.mvn', 'src', 'test', 'target', and files like 'HELP.md', 'mvnw', 'mvnw.cmd', and 'pom.xml'. The 'src' folder is expanded, showing a 'main' folder which contains a 'java' folder with a package 'M8E1' and a 'datos' folder. Inside 'datos', there are several repository classes: 'CategoriaRepository', 'DescuentoRepository', 'MarcaRepository', and 'RolRepository', along with their corresponding entity classes 'Categoria', 'Descuento', 'Marca', and 'Rol'. Below 'datos' is the 'M8E1FranciscoJimenezApplication' class. The 'resources' folder contains 'application.properties', 'data.sql', and 'schema.sql'. The 'target' folder is currently selected. On the right, the code editor displays the 'application.properties' file with the following configuration:

```
spring.application.name=M8E1_Francisco_Jimenez
spring.datasource.url=jdbc:mariadb://localhost:3306/bazarboost_m8_e1
spring.datasource.username=root
spring.datasource.password=sqlMD3
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
```

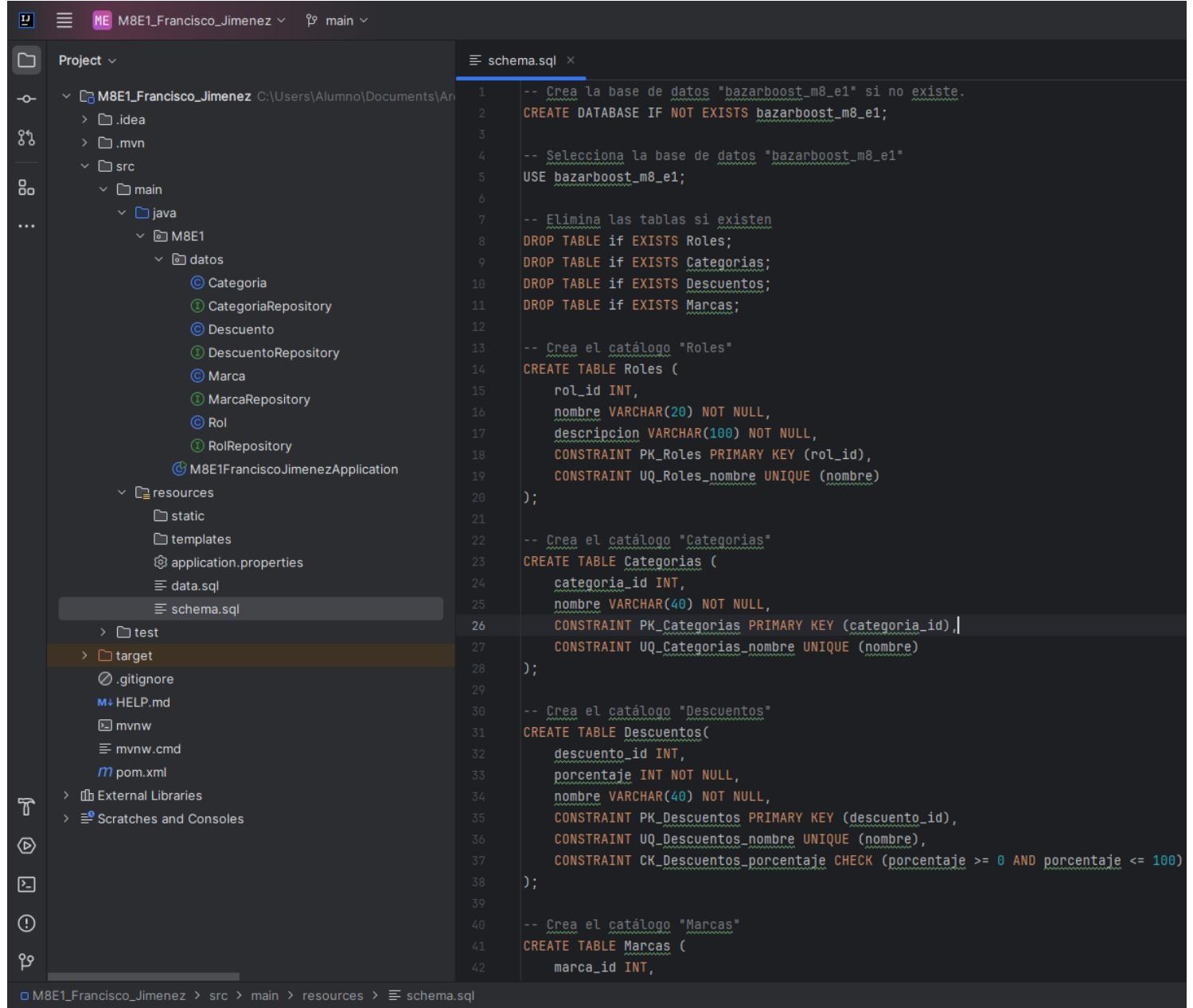
Captura 1. Código del archivo application.properties del proyecto para la configuración de la conexión a la base de datos.

Francisco Williams Jiménez Hernández

Ejercicio 1

Posteriormente, cree los scripts **schema.sql** y **data.sql** para la creación del esquema y la carga de los datos de prueba. Estos archivos los coloqué en la carpeta **resources** del proyecto. El contenido de ambos archivos se muestra a continuación.

Archivo: schema.sql



The screenshot shows the IntelliJ IDEA interface with the project 'M8E1_Francisco_Jimenez' open. The left sidebar displays the project structure, including the 'src' directory containing 'main' and 'resources' (which contains 'static', 'templates', 'application.properties', 'data.sql', and 'schema.sql'). The right pane shows the content of the 'schema.sql' file. The code is a SQL script for creating a database and its tables:

```
-- Crea la base de datos "bazarboost_m8_e1" si no existe.  
CREATE DATABASE IF NOT EXISTS bazarboost_m8_e1;  
  
-- Selecciona la base de datos "bazarboost_m8_e1"  
USE bazarboost_m8_e1;  
  
-- Elimina las tablas si existen  
DROP TABLE IF EXISTS Roles;  
DROP TABLE IF EXISTS Categorias;  
DROP TABLE IF EXISTS Descuentos;  
DROP TABLE IF EXISTS Marcas;  
  
-- Crea el catálogo "Roles"  
CREATE TABLE Roles (  
    rol_id INT,  
    nombre VARCHAR(20) NOT NULL,  
    descripcion VARCHAR(100) NOT NULL,  
    CONSTRAINT PK_Roles PRIMARY KEY (rol_id),  
    CONSTRAINT UQ_Roles_nombre UNIQUE (nombre)  
);  
  
-- Crea el catálogo "Categorias"  
CREATE TABLE Categorias (  
    categoria_id INT,  
    nombre VARCHAR(40) NOT NULL,  
    CONSTRAINT PK_Categorias PRIMARY KEY (categoria_id),  
    CONSTRAINT UQ_Categorias_nombre UNIQUE (nombre)  
);  
  
-- Crea el catálogo "Descuentos"  
CREATE TABLE Descuentos(  
    descuento_id INT,  
    porcentaje INT NOT NULL,  
    nombre VARCHAR(40) NOT NULL,  
    CONSTRAINT PK_Descuentos PRIMARY KEY (descuento_id),  
    CONSTRAINT UQ_Descuentos_nombre UNIQUE (nombre),  
    CONSTRAINT CK_Descuentos_porcentaje CHECK (porcentaje >= 0 AND porcentaje <= 100)  
);  
  
-- Crea el catálogo "Marcas"  
CREATE TABLE Marcas (  
    marca_id INT,
```

Captura 2.1. Primera parte del código del archivo schema.sql para la creación del esquema.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: schema.sql

```
12 -- Crea el catálogo "Roles"
13 CREATE TABLE Roles (
14     rol_id INT,
15     nombre VARCHAR(20) NOT NULL,
16     descripcion VARCHAR(100) NOT NULL,
17     CONSTRAINT PK_Roles PRIMARY KEY (rol_id),
18     CONSTRAINT UQ_Roles_nombre UNIQUE (nombre)
19 );
20
21
22 -- Crea el catálogo "Categorías"
23 CREATE TABLE Categorias (
24     categoria_id INT,
25     nombre VARCHAR(40) NOT NULL,
26     CONSTRAINT PK_Categorías PRIMARY KEY (categoria_id),
27     CONSTRAINT UQ_Categorías_nombre UNIQUE (nombre)
28 );
29
30 -- Crea el catálogo "Descuentos"
31 CREATE TABLE Descuentos(
32     descuento_id INT,
33     porcentaje INT NOT NULL,
34     nombre VARCHAR(40) NOT NULL,
35     CONSTRAINT PK_Descuentos PRIMARY KEY (descuento_id),
36     CONSTRAINT UQ_Descuentos_nombre UNIQUE (nombre),
37     CONSTRAINT CK_Descuentos_porcentaje CHECK (porcentaje >= 0 AND porcentaje <= 100)
38 );
39
40 -- Crea el catálogo "Marcas"
41 CREATE TABLE Marcas (
42     marca_id INT,
43     nombre VARCHAR(40) NOT NULL,
44     descripcion VARCHAR(100),
45     CONSTRAINT PK_Marcas PRIMARY KEY (marca_id),
46     CONSTRAINT UQ_Marcas_nombre UNIQUE (nombre)
47 );
48
```

Captura 2.2. Segunda parte del código del archivo schema.sql para la creación del esquema.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: data.sql

The screenshot shows a code editor interface with a dark theme. On the left is a project navigation sidebar titled 'Project'. It lists a 'M8E1_Francisco_Jimenez' project containing several sub-folders and files. The 'src/main/resources' folder is expanded, showing 'application.properties', 'data.sql' (which is currently selected), and 'schema.sql'. Below this are 'test', 'target', '.gitignore', 'HELP.md', 'mvnw', 'mvnw.cmd', and 'pom.xml'. The main workspace on the right displays the contents of the 'data.sql' file. The file contains SQL code for inserting data into four tables: 'Roles', 'Categorias', 'Descuentos', and 'Marcas'. The 'data.sql' file is located at the bottom of the navigation bar.

```
-- Selecciona la base de datos "bazarboost_m8_e1"
USE bazarboost_m8_e1;

-- Inserción de datos en la tabla "Roles"
INSERT INTO Roles (rol_id, nombre, descripcion) VALUES
(1, 'Administrador', 'Acceso total al sistema'),
(2, 'Cliente', 'Puede realizar compras y dejar reseñas'),
(3, 'Vendedor', 'Puede gestionar productos y ventas');

-- Inserción de datos en la tabla "Categorias"
INSERT INTO Categorias (categoria_id, nombre) VALUES
(1, 'Camisetas'),
(2, 'Pantalones'),
(3, 'Chaquetas'),
(4, 'Accesorios');

-- Inserción de datos en la tabla "Descuentos"
INSERT INTO Descuentos (descuento_id, porcentaje, nombre) VALUES
(1, 10, 'Descuento de Bienvenida'),
(2, 20, 'Descuento de Temporada'),
(3, 5, 'Descuento por Fidelidad'),
(4, 15, 'Descuento Black Friday');

-- Inserción de datos en la tabla "Marcas"
INSERT INTO Marcas (marca_id, nombre, descripcion) VALUES
(1, 'Nike', 'Marca deportiva enfocada en ropa casual y deportiva'),
(2, 'Adidas', 'Marca deportiva reconocida por sus accesorios y calzado'),
(3, 'Levi\'s', 'Marca icónica de pantalones y ropa de mezclilla'),
(4, 'H&M', 'Marca de moda rápida con una amplia gama de ropa y accesorios');
```

Captura 3. Código del archivo data.sql para la inserción de datos de prueba.

Francisco Williams Jiménez Hernández

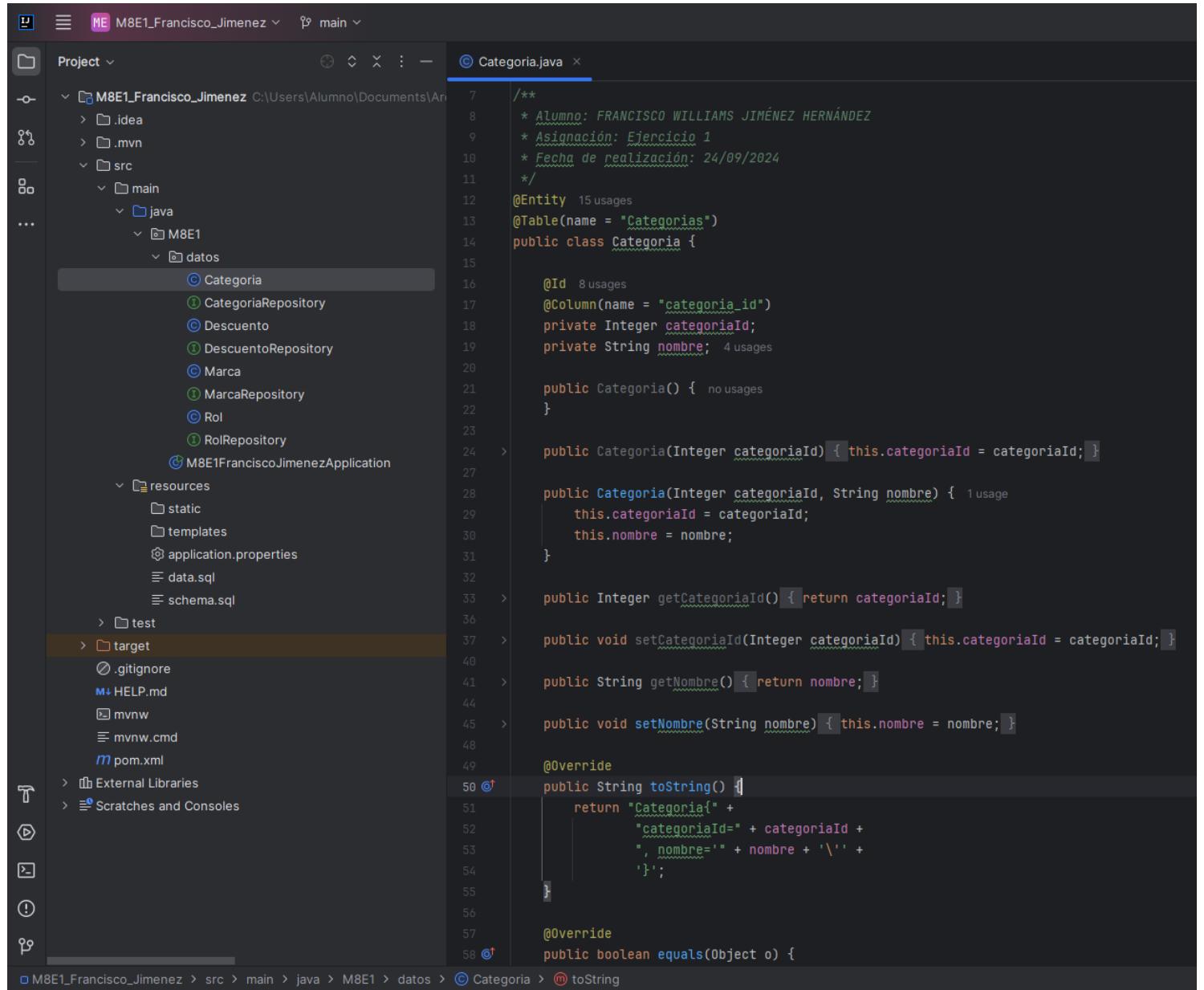
Ejercicio 1

2. Entidades

El siguiente paso consistió en crear las clases entidad utilizando anotaciones de Spring Data JPA para mapear las clases a las tablas en el modelo de datos. Las entidades se muestran a continuación.

Nota importante: Con la finalidad de no sobrecargar el documento de capturas de pantalla más de lo necesario y mantener un orden en el mismo, se muestra únicamente una captura por entidad; cada entidad está constituida por: Anotaciones JPA, atributos, tres métodos constructores (vacío, con id, con todos los atributos), métodos getters, métodos setters, método `toString`, método `hashCode` y método `equals` como se enseñó en clase.

Archivo: `Categoría.java`



The screenshot shows an IDE interface with the project tree on the left and the code editor on the right. The project tree shows a structure with 'src' containing 'main' and 'java' subfolders, which further contain 'M8E1' and 'datos' subfolders. Inside 'datos', there are several Java files: 'Categoría', 'CategoriaRepository', 'Descuento', 'DescuentoRepository', 'Marca', 'MarcaRepository', 'Rol', 'RolRepository', and 'M8E1FranciscoJimenezApplication'. The 'resources' folder contains 'static', 'templates', 'application.properties', 'data.sql', and 'schema.sql'. The 'test' and 'target' folders are also present. The '.gitignore' and 'mvnw' files are in the root. The 'External Libraries' and 'Scratches and Consoles' sections are at the bottom. The code editor displays the 'Categoría.java' file with the following content:

```
7  /**
8  * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
9  * Asignación: Ejercicio 1
10 * Fecha de realización: 24/09/2024
11 */
12 @Entity 15 usages
13 @Table(name = "Categorías")
14 public class Categoría {
15
16     @Id 8 usages
17     @Column(name = "categoria_id")
18     private Integer categoriaId;
19     private String nombre; 4 usages
20
21     public Categoría() { no usages
22     }
23
24     public Categoría(Integer categoriaId) { this.categoriaId = categoriaId; }
25
26     public Categoría(Integer categoriaId, String nombre) { 1 usage
27         this.categoriaId = categoriaId;
28         this.nombre = nombre;
29     }
30
31     public Integer getCategoriaId() { return categoriaId; }
32
33     public void setCategoriaId(Integer categoriaId) { this.categoriaId = categoriaId; }
34
35     public String getNombre() { return nombre; }
36
37     public void setNombre(String nombre) { this.nombre = nombre; }
38
39     @Override
40     public String toString() {
41         return "Categoría{" +
42             "categoriaId=" + categoriaId +
43             ", nombre='" + nombre + '\'' +
44             '}';
45     }
46
47     @Override
48     public boolean equals(Object o) {
49
50     }
51
52
53
54
55
56
57
58 }
```

Captura 4. Clase entidad para el catálogo “Categoría”.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: Descuento.java

The screenshot shows an IDE interface with the following details:

- Project:** M8E1_Francisco_Jimenez
- File:** Descuento.java
- Content:** The code defines a Java entity class named Descuento. It includes annotations for Entity, Table, and various fields like descuentoId, porcentaje, and nombre. The class has constructors, getters, setters, and a toString method.
- Project Structure:** The project structure on the left shows the following hierarchy:
 - M8E1_Francisco_Jimenez
 - .idea
 - .mvn
 - src
 - main
 - java
 - M8E1
 - datos
 - Categoría
 - CategoríaRepository
 - Descuento
 - DescuentoRepository
 - Marca
 - MarcaRepository
 - Rol
 - RolRepository
 - M8E1FranciscoJimenezApplication
 - resources
 - static
 - templates
 - application.properties
 - data.sql
 - schema.sql
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - External Libraries
 - Scratches and Consoles

Captura 5. Clase entidad para el catálogo “Descuento”.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: Marca.java

The screenshot shows a Java code editor within an IDE. The project navigation bar at the top indicates the current project is "M8E1_Francisco_Jimenez" and the active file is "main". The left sidebar displays the project structure under "Project". The "src" folder contains "main", which has "java" and "M8E1" subfolders. "M8E1" contains "datos" and "Marca.java", which is currently selected. Other files in "M8E1" include "Categoria", "CategoriaRepository", "Descuento", "DescuentoRepository", "Rol", "RolRepository", and "M8E1FranciscoJimenezApplication". The "resources" folder contains "static", "templates", "application.properties", "data.sql", and "schema.sql". The "test" and "target" folders are also listed. The bottom status bar shows the full path: "M8E1_Francisco_Jimenez > src > main > java > M8E1 > datos > Marca > setNombre".

```
7  /**
8   * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
9   * Asignación: Ejercicio 1
10  * Fecha de realización: 24/09/2024
11  */
12 @Entity 15 usages
13 @Table(name = "Marcas")
14 public class Marca {
15
16     @Id 8 usages
17     @Column(name = "marca_id")
18     private Integer marcaId;
19     private String nombre; 4 usages
20
21     public Marca() { no usages
22     }
23
24     public Marca(Integer marcaId) { this.marcaId = marcaId; }
25
26     public Marca(Integer marcaId, String nombre) { 1 usage
27         this.marcaId = marcaId;
28         this.nombre = nombre;
29     }
30
31     public Integer getMarcaId() { return marcaId; }
32
33     public void setMarcaId(Integer marcaId) { this.marcaId = marcaId; }
34
35     public String getNombre() { return nombre; }
36
37     public void setNombre(String nombre) { 1 usage
38         this.nombre = nombre;
39     }
40
41     @Override
42     public String toString() {
43         return "Marca{" +
44             "marcaId=" + marcaId +
45             ", nombre='" + nombre + '\'' +
46         '}';
47     }
48
49
50     @+
51
52
53
54
55 }
```

Captura 6. Clase entidad para el catálogo “Marca”.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: Rol.java

The screenshot shows the IntelliJ IDEA interface with the project 'M8E1_Francisco_Jimenez' open. The left sidebar displays the project structure, and the right pane shows the code editor for 'Rol.java'. The code defines a Java entity class 'Rol' with annotations for JPA and a constructor, getters, and setters for fields 'rolId', 'nombre', and 'descripcion'.

```
7  /**
8   * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
9   * Asignación: Ejercicio 1
10  * Fecha de realización: 24/09/2024
11  */
12 @Entity 15 usages
13 @Table(name = "Roles")
14 public class Rol {
15
16     @Id 8 usages
17     @Column(name = "rol_id")
18     private Integer rolId;
19
20     private String nombre; 4 usages
21
22     private String descripcion; 4 usages
23
24     public Rol() { no usages
25
26     }
27
28
29     public Rol(Integer rolId) { no usages
30         this.rolId = rolId;
31     }
32
33
34     public Rol(Integer rolId, String nombre, String descripcion) { 1 usage
35         this.rolId = rolId;
36         this.nombre = nombre;
37         this.descripcion = descripcion;
38     }
39
40     public Integer getRolId() { no usages
41         return rolId;
42     }
43
44     public void setRolId(Integer rolId) { no usages
45         this.rolId = rolId;
46     }
47
48     public String getNombre() { no usages
49         return nombre;
50     }
51
52     public void setNombre(String nombre) { 1 usage
53         this.nombre = nombre;
54     }
55 }
```

Captura 7. Clase entidad para el catálogo “Rol”.

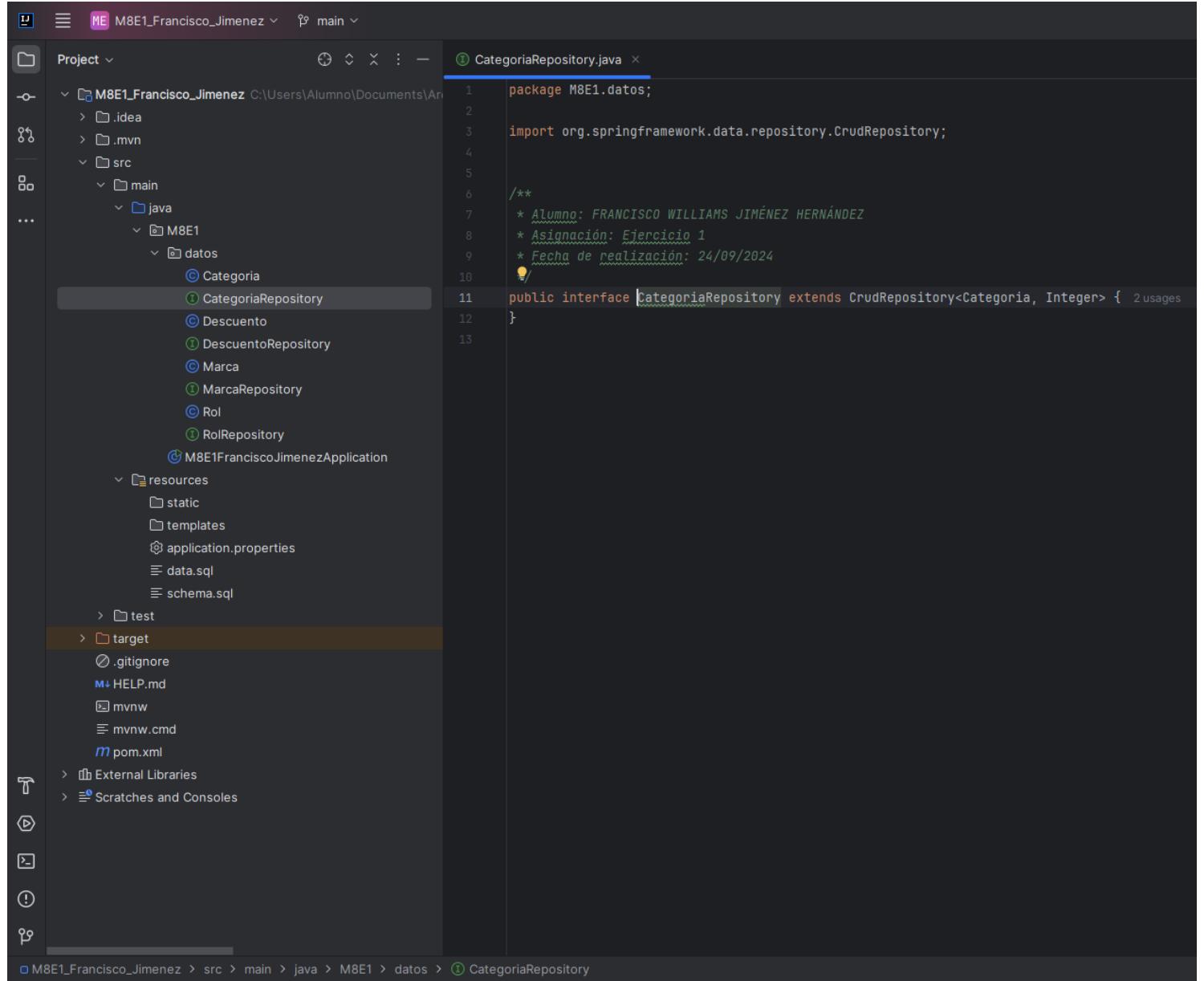
Francisco Williams Jiménez Hernández

Ejercicio 1

3. Repositorios

Después de crear las entidades, cree los repositorios. Para la creación de los repositorios utilicé la interfaz CrudRepository, indicando para cada repositorio la entidad y el tipo de dato del identificador con el que se trabaja mediante el uso de genéricos. A continuación, se muestran los repositorios.

Archivo: CategoriaRepository.java



The screenshot shows a Java code editor with the file `CategoriaRepository.java` open. The code defines a repository interface for the `Categoría` entity, extending the `CrudRepository` interface from Spring Data. The code includes a header with student information and a copyright notice. The project structure on the left shows the `M8E1_Francisco_Jimenez` project with its `src/main/java` directory containing the repository interface and other entity classes like `Descuento`, `Marca`, and `Rol`. The `target` directory is also visible.

```
package M8E1.datos;

import org.springframework.data.repository.CrudRepository;

/**
 * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
 * Asignación: Ejercicio 1
 * Fecha de realización: 24/09/2024
 */
public interface CategoriaRepository extends CrudRepository<Categoria, Integer> { }
```

Captura 8. Interfaz para el repositorio de la entidad “Categoría” usando CrudRepository.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: DescuentoRepository.java

The screenshot shows a Java project structure in an IDE. The project is named 'M8E1_Francisco_Jimenez' and contains a 'main' module. The 'src/main/java/M8E1/datos' package contains several repository interfaces: CategoríaRepository, DescuentoRepository, MarcaRepository, RolRepository, and M8E1FranciscoJimenezApplication. The 'DescuentoRepository' interface is currently selected. The code for 'DescuentoRepository' is as follows:

```
package M8E1.datos;  
import org.springframework.data.repository.CrudRepository;  
  
/**  
 * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ  
 * Asignación: Ejercicio 1  
 * Fecha de realización: 24/09/2024  
 */  
public interface DescuentoRepository extends CrudRepository<Descuento, Integer> { 2 usages }
```

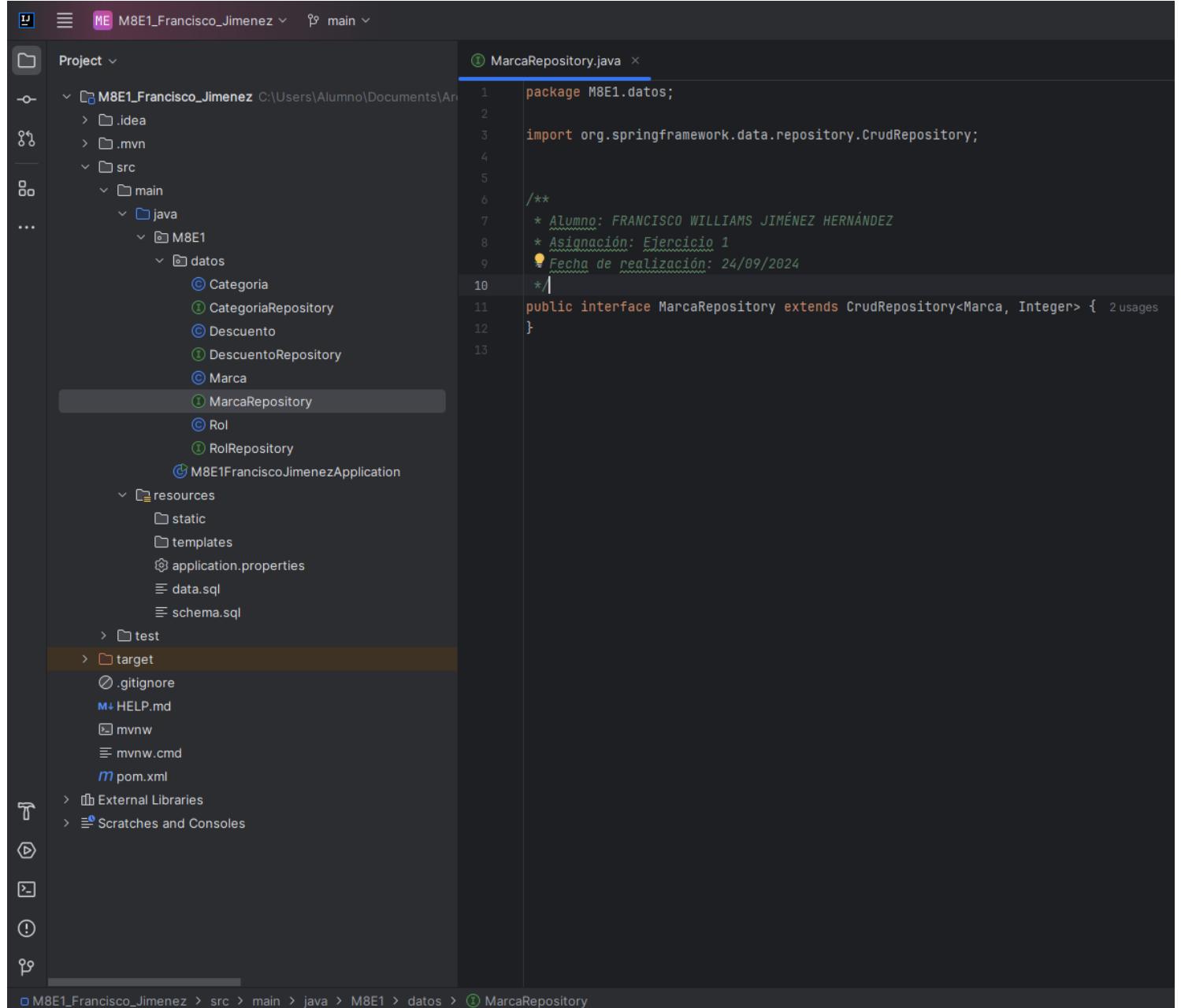
The 'target' folder is highlighted in the project tree. The status bar at the bottom shows the full path: 'M8E1_Francisco_Jimenez > src > main > java > M8E1 > datos > DescuentoRepository'. The status bar also displays the message 'Captura 9. Interfaz para el repositorio de la entidad “Descuento” usando CrudRepository.'

Captura 9. Interfaz para el repositorio de la entidad “Descuento” usando CrudRepository.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: MarcaRepository.java



```
package M8E1.datos;  
import org.springframework.data.repository.CrudRepository;  
  
/**  
 * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ  
 * Asignación: Ejercicio 1  
 * Fecha de realización: 24/09/2024  
 */  
public interface MarcaRepository extends CrudRepository<Marca, Integer> { 2 usages  
}
```

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `MarcaRepository.java` file. The file contains Java code for a repository interface that extends the `CrudRepository` interface from Spring Data. The code is annotated with a copyright notice and a date. The project structure on the left shows the directory `M8E1_Francisco_Jimenez` containing `.idea`, `.mvn`, `src`, `resources`, `target`, and other files like `pom.xml` and `application.properties`.

Captura 10. Interfaz para el repositorio de la entidad “Marca” usando CrudRepository.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: RolRepository.java

The screenshot shows a Java project structure in an IDE. The project is named 'M8E1_Francisco_Jimenez' located at 'C:\Users\Alumno\Documents\Archivos'. The 'src' directory contains a 'main' package with a 'java' folder. Inside 'java' is a 'M8E1' package, which contains a 'datos' package. The 'datos' package contains several repository interfaces: 'Categoria', 'CategoriaRepository', 'Descuento', 'DescuentoRepository', 'Marca', 'MarcaRepository', 'Rol', and 'RolRepository'. The 'RolRepository' interface is currently selected and shown in the code editor. The code implements the 'CrudRepository' interface for the 'Rol' entity. The code includes a header with student information and a copyright notice.

```
package M8E1.datos;

import org.springframework.data.repository.CrudRepository;

/**
 * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
 * Asignación: Ejercicio 1
 * Fecha de realización: 24/09/2024
 */
public interface RolRepository extends CrudRepository<Rol, Integer> { }
```

Captura 11. Interfaz para el repositorio de la entidad “Rol” usando CrudRepository.

4. Pruebas CRUD

El último paso consistió en crear una clase de pruebas para las operaciones CRUD para cada entidad:

- M8E1_CRUD_Tests_Categoría
- M8E1_CRUD_Tests_Descuento
- M8E1_CRUD_Tests_Marca
- M8E1_CRUD_Tests_Rol

Para cada una de estas clases se utilizó la anotación `@Sql` para asegurar la ejecución de los scripts `schema.sql` y `data.sql` antes de la ejecución de cada método de prueba y así asegurar la carga del esquema y datos iniciales antes de cada prueba. Adicionalmente se hace uso de las anotaciones `@TestMethodOrder` y `@Order` con el único fin de que las pruebas se ejecuten en un orden específico en caso de ejecutarse la clase completa y no una por una, para así mantener una salida limpia.

Cada clase de prueba incluye ocho métodos de prueba, cada uno para una operación CRUD vista en clase:

- Insertar un registro
- Leer un registro por ID
- Leer todos los registros
- Actualizar un registro
- Eliminar un registro
- Buscar varios registros por IDs
- Insertar varios registros
- Eliminar varios registros

Francisco Williams Jiménez Hernández

Ejercicio 1

4.1. CRUD para Categoría

Archivo: M8E1_CRUD_Tests_Categoría.java (parte 1)

The screenshot shows an IDE interface with the following details:

- Project:** M8E1_Francisco_Jimenez
- File:** M8E1_CRUD_Tests_Categoría.java
- Content Summary:** The code is a Java test class for the Category entity. It includes imports for Spring Boot, JUnit, and various repository interfaces. The class contains a single test method, `insertarUnaCategoriaTest`, which prints category lists before and after insertion and checks if the new category exists.
- Code Snippet:** Below is a snippet of the code from line 17 to line 58, highlighting the test method and its logic.

```
17  /**
18   * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
19   * Asignación: Ejercicio 1
20   * Fecha de realización: 24/09/2024
21   * Descripción: Pruebas para el CRUD de la entidad Categoría
22  */
23
24 @SpringBootTest
25 @Sql({{"schema.sql", "data.sql"}}) // Carga esquema y datos de inicio antes de cada prueba
26 @TestMethodOrder(MethodOrderer.OrderAnnotation.class) // Ejecuta las pruebas en orden para un resultado más predecible
27 public class M8E1_CRUD_Tests_Categoría {
28
29     private final String ALUMNO = "FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ"; 8 usages
30     private final int CATEGORÍA_ID_INSERTADA = 5; // ID para la nueva categoría 3 usages
31     private final int CATEGORÍA_ID_ELIMINADA = 1; // ID para la categoría a eliminar 4 usages
32     private final int CATEGORÍA_ID_BUSCAR = 3; // ID para la categoría a buscar 3 usages
33     private final String CATEGORÍA_NUEVA_NOMBRE = "Calzado"; 1 usage
34     private final String CATEGORÍA_ACTUALIZADA_NOMBRE = "Playeras"; 1 usage
35
36     @Autowired 20 usages
37     CategoriaRepository categoriaRepository;
38
39     @Test
40     @Order(1)
41     void insertarUnaCategoriaTest() {
42         System.out.println("\n*****");
43         System.out.println(ALUMNO);
44         System.out.println("Prueba 1: Insertar una categoría");
45         System.out.println("*****");
46
47         // Lista de categorías antes de la inserción
48         System.out.println("Lista de categorías ANTES de la inserción:");
49         categoriaRepository.findAll().forEach(System.out::println);
50
51         // Inserción de una nueva categoría
52         Optional<Categoria> existingCategory = categoriaRepository.findById(CATEGORÍA_ID_INSERTADA);
53         if (existingCategory.isPresent()) {
54             System.out.println("\nLa categoría con ID " + CATEGORÍA_ID_INSERTADA + " ya existe. No se insertará.");
55         } else {
56             Categoria nuevaCategoria = new Categoria(CATEGORÍA_ID_INSERTADA, CATEGORÍA_NUEVA_NOMBRE);
57             categoriaRepository.save(nuevaCategoria);
58             System.out.println("\nNueva categoría insertada correctamente: " + nuevaCategoria);
59         }
60     }
61 }
```

Captura 12.1. Clase de pruebas CRUD para el catálogo “Categoría”.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Categoría.java (parte 2)

The screenshot shows an IDE interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" folder contains "main" and "test" packages. "main" has a "java" folder with "M8E1" and "datos" subfolders, which contain "Categoria", "CategoriaRepository", "Descuento", "DescuentoRepository", "Marca", "MarcaRepository", "Rol", "RolRepository", and "M8E1FranciscoJimenezApplication". "resources" contains "static", "templates", "application.properties", "data.sql", and "schema.sql". "test" contains "java" with "M8E1" and "M8E1_CRUD_Tests_Categoría", "M8E1_CRUD_Tests_Descuento", "M8E1_CRUD_Tests_Marca", "M8E1_CRUD_Tests_Rol", and "M8E1FranciscoJimenezApplicationTests".
- M8E1_CRUD_Tests_Categoría.java Content:**

```
public class M8E1_CRUD_Tests_Categoría {
    void insertarUnaCategoriaTest() {
        // Lista de categorías después de la inserción
        System.out.println("\nLista de categorías DESPUÉS de la inserción:");
        categoriaRepositorio.findAll().forEach(System.out::println);
    }

    @Test
    @Order(2)
    void buscarUnaCategoriaPorIdTest() {
        System.out.println("=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 2: Buscar una categoría por ID");
        System.out.println("-----");
        System.out.println("ID de la categoría a buscar: " + CATEGORIA_ID_BUSCAR);

        // Búsqueda por ID
        Optional<Categoria> optional = categoriaRepositorio.findById(CATEGORIA_ID_BUSCAR);
        if (optional.isPresent()) {
            System.out.println("Categoria encontrada: " + optional.get());
        } else {
            System.out.println("No se encontró una categoría con id " + CATEGORIA_ID_BUSCAR + ".");
        }
    }

    @Test
    @Order(3)
    void buscarTodasLasCategoriasTest() {
        System.out.println("=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 3: Buscar todas las categorías");
        System.out.println("-----");

        // Lista de todas las categorías
        Iterable<Categoria> categorias = categoriaRepositorio.findAll();
        System.out.println("Lista de todas las categorías:");
        categorias.forEach(System.out::println);
    }

    @Test
    @Order(4)
    void actualizarUnaCategoriaTest() {
    }
}
```
- Bottom Status Bar:** Shows the current file path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Categoría > insertarUnaCategoriaTest.

Captura 12.2. Clase de pruebas CRUD para el catálogo “Categoría”.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Categoría.java (parte 3)

```
public class M8E1_CRUD_Tests_Categoría {
    void actualizarUnaCategoriaTest() {
        System.out.println("\n=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 4: Actualizar una categoría");
        System.out.println("-----");

        // Búsqueda de la categoría para actualizar
        Optional<Categoria> optional = categoriaRepositorio.findById(CATEGORIA_ID_ELIMINADA);
        if (optional.isPresent()) {
            Categoria categoria = optional.get();
            System.out.println("Categoría ANTES de actualizar: " + categoria);

            // Actualización de la categoría
            categoria.setNombre(CATEGORIA_ACTUALIZADA_NOMBRE);
            categoriaRepositorio.save(categoria);
            System.out.println("Categoría DESPUÉS de actualizar: " + categoria);
        } else {
            System.out.println("No se encontró la categoría con ID " + CATEGORIA_ID_ELIMINADA + " para actualizar.");
        }
    }

    @Test
    @Order(5)
    void eliminarUnaCategoriaTest() {
        System.out.println("\n=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 5: Eliminar una categoría");
        System.out.println("-----");

        // Verificación de la categoría antes de eliminar
        Optional<Categoria> optional = categoriaRepositorio.findById(CATEGORIA_ID_ELIMINADA);
        if (optional.isPresent()) {
            Categoria categoriaAEliminar = optional.get();
            System.out.println("Categoría a eliminar: " + categoriaAEliminar);

            // Lista antes de la eliminación
            System.out.println("\nLista de categorías ANTES de la eliminación:");
            categoriaRepositorio.findAll().forEach(System.out::println);

            // Eliminación
            categoriaRepositorio.delete(categoriaAEliminar);
        }
    }
}
```

Captura 12.3. Clase de pruebas CRUD para el catálogo “Categoría”.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1 CRUD_Tests_Categoría.java (parte 4)

The screenshot shows an IDE interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" folder contains "main" and "test" packages. "main/java" includes classes like Categoría, CategoriaRepository, Descuento, DescuentoRepository, Marca, MarcaRepository, Rol, and RolRepository, along with M8E1FranciscoJimenezApplication. "test/java" includes test classes for Categoría, Descuento, Marca, Rol, and M8E1FranciscoJimenezApplicationTests.
- Code Editor:** The active file is "M8E1 CRUD_Tests_Categoría.java". The code implements several test methods for the Categoría entity:
 - eliminarUnaCategoríaTest():** Prints the list of categories after deleting one.
 - buscarVariasCategoríasTest():** Prints the list of categories for IDs 1, 3, and 5.
 - insertarVariasCategoríasTest():** Prints the list of categories before inserting three new ones (Sombreros, Bufandas, Guantes).
- Bottom Status Bar:** Shows the current path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1 CRUD_Tests_Categoría.

Captura 12.4. Clase de pruebas CRUD para el catálogo “Categoría”.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Categoría.java (parte 5)

The screenshot shows the IntelliJ IDEA interface with the project 'M8E1_Francisco_Jimenez' open. The left sidebar displays the project structure, including 'src' (containing 'main' and 'test') and 'resources'. The 'test/java' directory contains several test classes: 'M8E1_CRUD_Tests_Categoría', 'M8E1_CRUD_Tests_Descuento', 'M8E1_CRUD_Tests_Marca', 'M8E1_CRUD_Tests_Rol', and 'M8E1FranciscoJimenezApplicationTests'. The current file being edited is 'M8E1_CRUD_Tests_Categoría.java'. The code implements two test methods: 'insertarVariasCategoríasTest()' and 'eliminarVariasCategoríasTest()'. Both tests interact with a 'CategoriaRepository' to manage category data.

```
public class M8E1_CRUD_Tests_Categoría {
    void insertarVariasCategoríasTest() {
        List<Categoria> nuevasCategorías = new ArrayList<>();
        nuevasCategorías.add(new Categoria( categoriaId: 6, nombre: "Sombreros"));
        nuevasCategorías.add(new Categoria( categoriaId: 7, nombre: "Bufandas"));
        nuevasCategorías.add(new Categoria( categoriaId: 8, nombre: "Guantes"));

        categoriaRepositorio.saveAll(nuevasCategorías);

        // Lista después de la inserción
        System.out.println("\nLista de categorías DESPUÉS de la inserción:");
        categoriaRepositorio.findAll().forEach(System.out::println);
    }

    @Test
    @Order(8)
    void eliminarVariasCategoríasTest() {
        System.out.println("=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 8: Eliminar varias categorías");
        System.out.println("-----");

        // Lista antes de la eliminación
        System.out.println("Lista de categorías ANTES de la eliminación:");
        categoriaRepositorio.findAll().forEach(System.out::println);

        // Eliminación de varias categorías
        List<Integer> idsEliminar = List.of(6, 7, 8); // IDs de las categorías a eliminar
        Iterable<Categoria> categoríasAEliminar = categoriaRepositorio.findAllById(idsEliminar);
        categoriaRepositorio.deleteAll(categoríasAEliminar);

        // Lista después de la eliminación
        System.out.println("\nLista de categorías DESPUÉS de la eliminación:");
        categoriaRepositorio.findAll().forEach(System.out::println);
    }
}
```

Captura 12.5. Clase de pruebas CRUD para el catálogo “Categoría”.

Francisco Williams Jiménez Hernández

Ejercicio 1

A continuación, se muestra la salida en consola de la ejecución de la clase de pruebas **M8E1_CRUD_Tests_Categoría**

Consola

The screenshot shows the IntelliJ IDEA IDE interface with the project 'M8E1_Francisco_Jimenez' selected. The 'Run' tool window is open, showing the results of running the test 'M8E1_CRUD_Tests_Categoría'. The output pane displays the following text:

```
Tests passed: 8 of 8 tests - 2 sec 850 ms
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 1: Insertar una categoría
-----
Lista de categorías ANTES de la inserción:
Categoria{categoriaId=4, nombre='Accesorios'}
Categoria{categoriaId=1, nombre='Camisetas'}
Categoria{categoriaId=3, nombre='Chaquetas'}
Categoria{categoriaId=2, nombre='Pantalones'}

Nueva categoría insertada correctamente: Categoria{categoriaId=5, nombre='Calzado'}

Lista de categorías DESPUÉS de la inserción:
Categoria{categoriaId=4, nombre='Accesorios'}
Categoria{categoriaId=5, nombre='Calzado'}
Categoria{categoriaId=1, nombre='Camisetas'}
Categoria{categoriaId=3, nombre='Chaquetas'}
Categoria{categoriaId=2, nombre='Pantalones'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 2: Buscar una categoría por ID
-----
ID de la categoría a buscar: 3
Categoría encontrada: Categoria{categoriaId=3, nombre='Chaquetas'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 3: Buscar todas las categorías
-----
Lista de todas las categorías:
Categoria{categoriaId=4, nombre='Accesorios'}
Categoria{categoriaId=1, nombre='Camisetas'}
Categoria{categoriaId=3, nombre='Chaquetas'}
Categoria{categoriaId=2, nombre='Pantalones'}
```

Captura 13.1. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Categoría.

Francisco Williams Jiménez Hernández

Ejercicio 1

```
M8E1_Francisco_Jimenez main
```

```
Project Run M8E1_CRUD_Tests_Categoría
```

```
Tests passed: 8 of 8 tests - 2 sec 850 ms
```

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 4: Actualizar una categoría
-----
Categoría ANTES de actualizar: Categoría{categoryId=1, nombre='Camisetas'}
Categoría DESPUÉS de actualizar: Categoría{categoryId=1, nombre='Playeras'}
```

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 5: Eliminar una categoría
-----
Categoría a eliminar: Categoría{categoryId=1, nombre='Camisetas'}
```

```
Lista de categorías ANTES de la eliminación:
Categoría{categoryId=4, nombre='Accesorios'}
Categoría{categoryId=1, nombre='Camisetas'}
Categoría{categoryId=3, nombre='Chaquetas'}
Categoría{categoryId=2, nombre='Pantalones'}
```

```
Lista de categorías DESPUÉS de la eliminación:
Categoría{categoryId=4, nombre='Accesorios'}
Categoría{categoryId=3, nombre='Chaquetas'}
Categoría{categoryId=2, nombre='Pantalones'}
```

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 6: Buscar varias categorías por ID
-----
Categorías encontradas:
Categoría{categoryId=1, nombre='Camisetas'}
Categoría{categoryId=3, nombre='Chaquetas'}
```

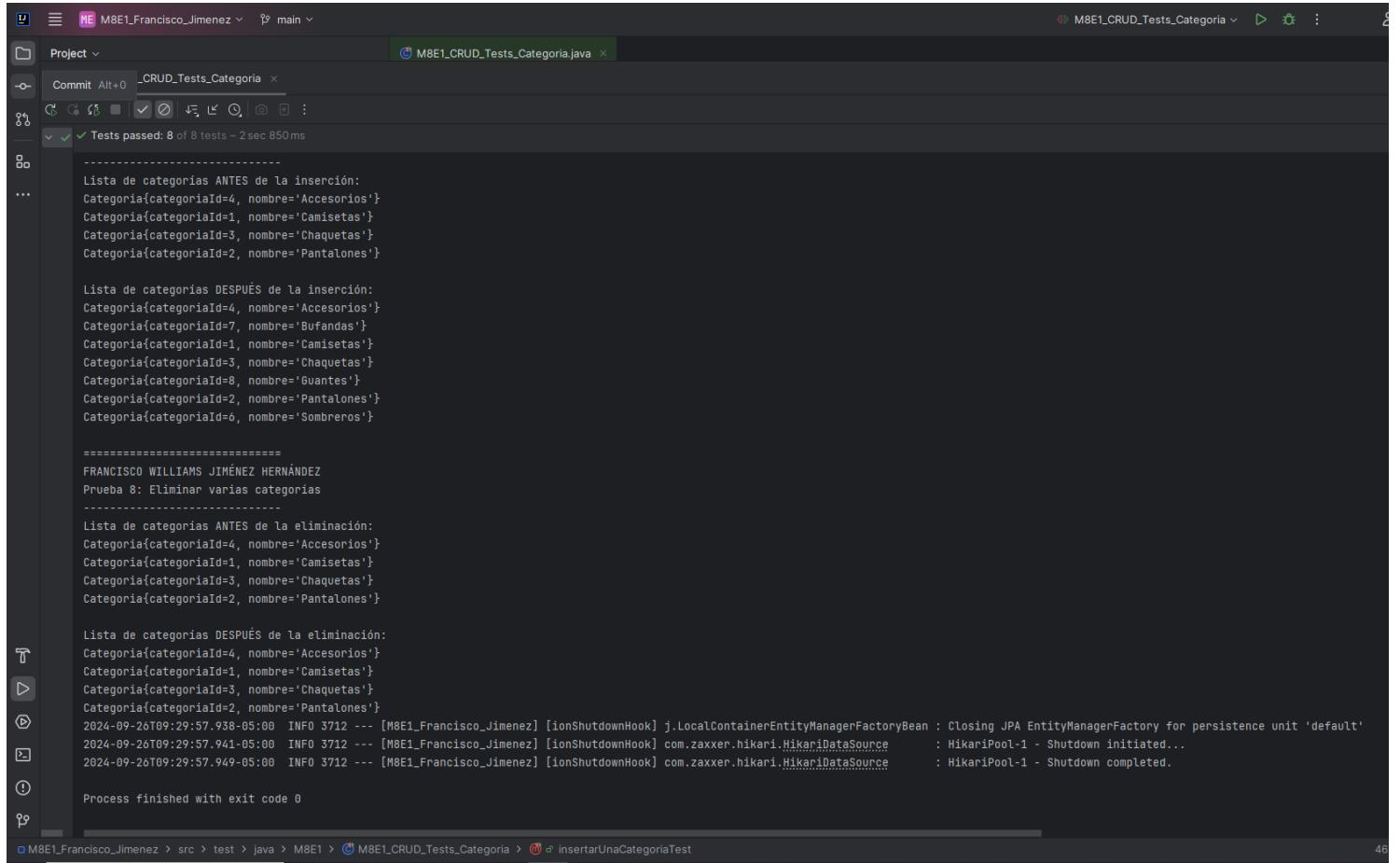
```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 7: Insertar varias categorías
-----
Lista de categorías ANTES de la inserción:
```

```
M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Categoría > insertarUnaCategoriaTest
```

Captura 13.2. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Categoría.

Francisco Williams Jiménez Hernández

Ejercicio 1



The screenshot shows an IDE interface with a terminal window displaying the output of a Java test suite named `M8E1_CRUD_Tests_Categoría.java`. The terminal shows the following text:

```
Tests passed: 8 of 8 tests - 2 sec 850 ms
-----
Lista de categorías ANTES de la inserción:
Categoria{id=4, nombre='Accesorios'}
Categoria{id=1, nombre='Camisetas'}
Categoria{id=3, nombre='Chaquetas'}
Categoria{id=2, nombre='Pantalones'}

Lista de categorías DESPUÉS de la inserción:
Categoria{id=4, nombre='Accesorios'}
Categoria{id=7, nombre='Bufandas'}
Categoria{id=1, nombre='Camisetas'}
Categoria{id=3, nombre='Chaquetas'}
Categoria{id=8, nombre='Guantes'}
Categoria{id=2, nombre='Pantalones'}
Categoria{id=6, nombre='Sombreros'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 8: Eliminar varias categorías
-----
Lista de categorías ANTES de la eliminación:
Categoria{id=4, nombre='Accesorios'}
Categoria{id=1, nombre='Camisetas'}
Categoria{id=3, nombre='Chaquetas'}
Categoria{id=2, nombre='Pantalones'

Lista de categorías DESPUÉS de la eliminación:
Categoria{id=4, nombre='Accesorios'}
Categoria{id=1, nombre='Camisetas'}
Categoria{id=3, nombre='Chaquetas'}
Categoria{id=2, nombre='Pantalones'}
2024-09-26T09:29:57.938-05:00 INFO 3712 --- [M8E1_Francisco_Jimenez] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2024-09-26T09:29:57.941-05:00 INFO 3712 --- [M8E1_Francisco_Jimenez] com.zaxxer.hikari.HikariDataSource      : HikariPool-1 - Shutdown initiated...
2024-09-26T09:29:57.949-05:00 INFO 3712 --- [M8E1_Francisco_Jimenez] com.zaxxer.hikari.HikariDataSource      : HikariPool-1 - Shutdown completed.

Process finished with exit code 0
```

The terminal also displays the file path: `M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Categoría > insertarUnaCategoríaTest`.

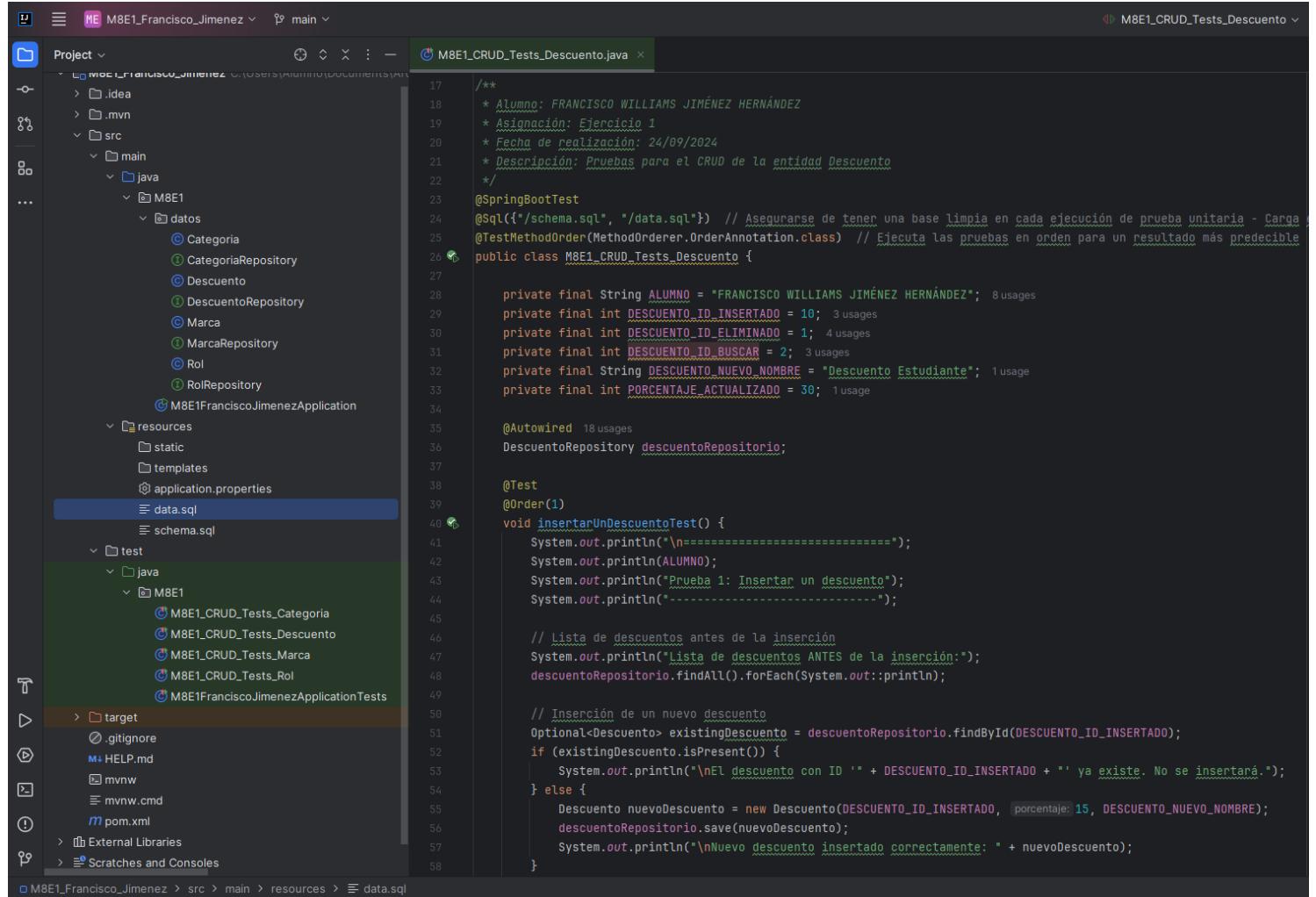
Captura 13.3. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Categoría.

Francisco Williams Jiménez Hernández

Ejercicio 1

4.2.CRUD para Descuento

Archivo: M8E1_CRUD_Tests_Descuento.java (parte 1)



The screenshot shows an IDE interface with the following details:

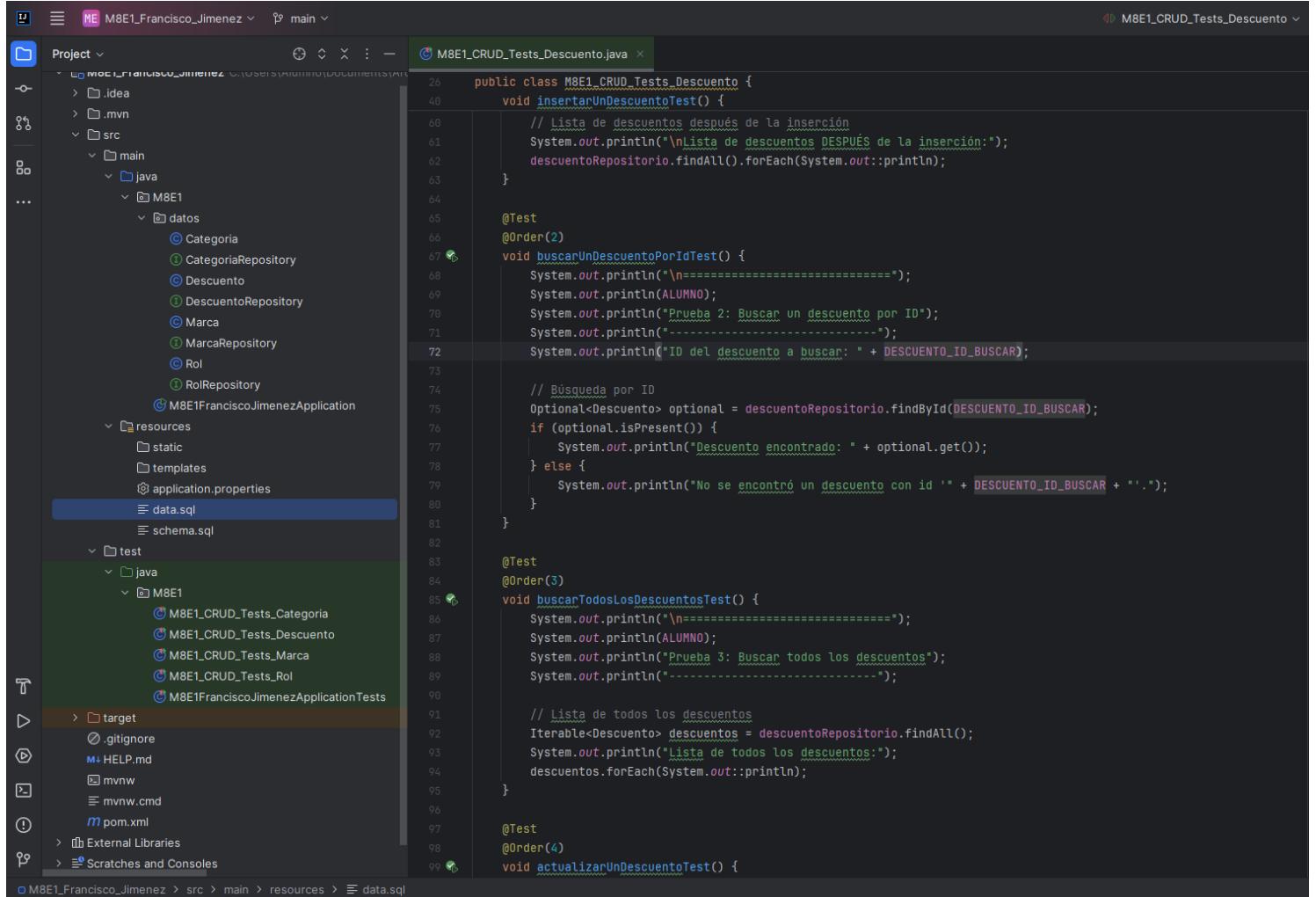
- Project:** M8E1_Francisco_Jimenez
- File:** M8E1_CRUD_Tests_Descuento.java
- Code Content:** The code is a Spring Boot test class for the Descuento entity. It includes annotations like @SpringBootTest, @Sql, and @TestMethodOrder. The class contains methods for inserting, finding, updating, and deleting Descuento objects, along with corresponding assertions and output statements.
- Project Structure:** The project structure on the left shows the following hierarchy:
 - src
 - main
 - java
 - M8E1
 - datos
 - Categoría
 - CategoríaRepository
 - Descuento
 - DescuentoRepository
 - Marca
 - MarcaRepository
 - Rol
 - RolRepository
 - M8E1FranciscoJimenezApplication
 - resources
 - static
 - templates
 - application.properties
 - data.sql
 - schema.sql
 - test
 - java
 - M8E1
 - M8E1_CRUD_Tests_Categoría
 - M8E1_CRUD_Tests_Descuento
 - M8E1_CRUD_Tests_Marca
 - M8E1_CRUD_Tests_Rol
 - M8E1FranciscoJimenezApplicationTests
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - External Libraries
 - Scratches and Consoles

Captura 14.1. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Descuento.java (parte 2)



The screenshot shows an IDE interface with the following details:

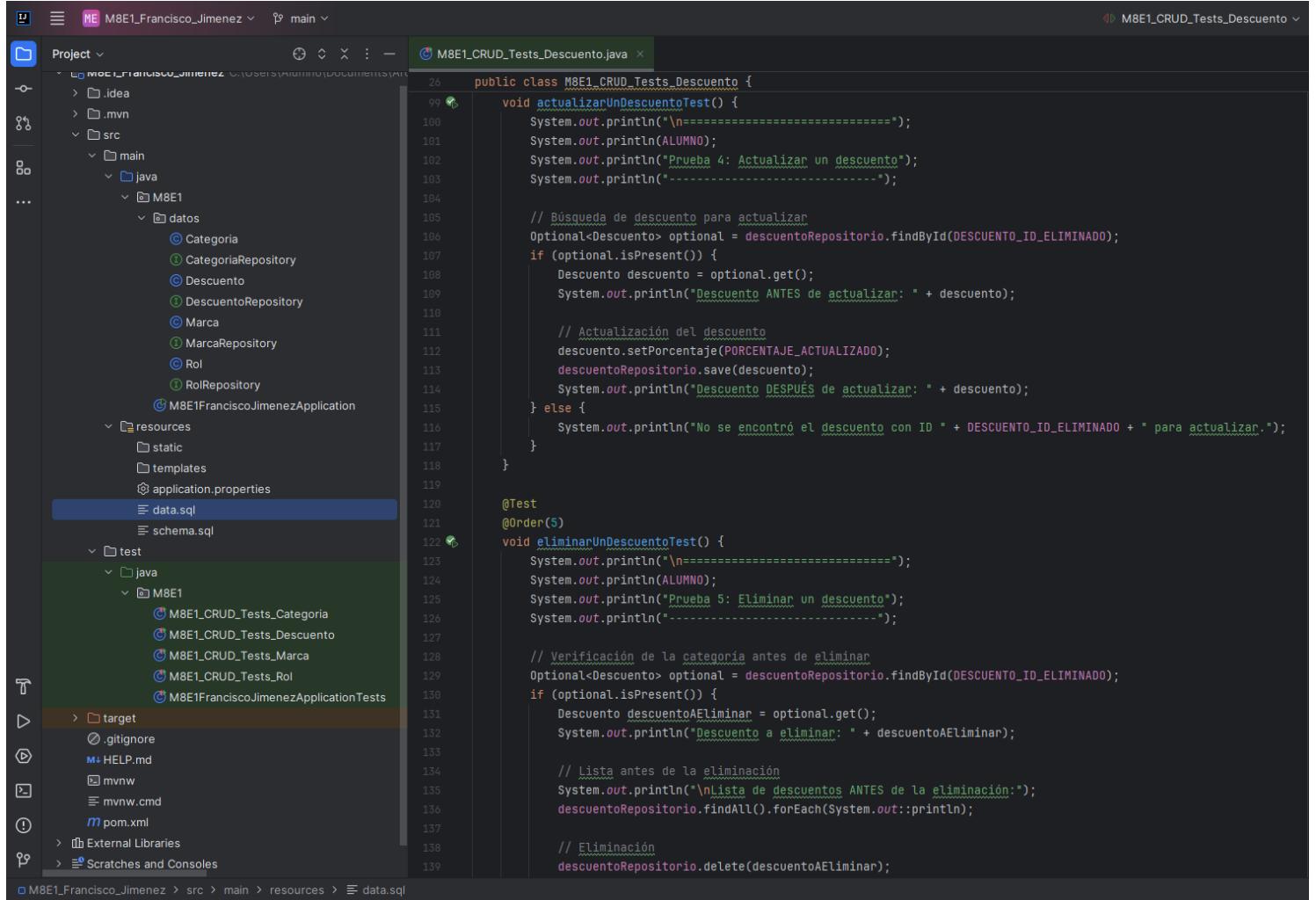
- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src/main/java/M8E1" package contains several repository classes (CategoriaRepository, DescuentoRepository, MarcaRepository, RolRepository) and an application class (M8E1FranciscoJimenezApplication). The "src/test/java/M8E1" package contains test classes for Categoria, Descuento, Marca, Rol, and M8E1FranciscoJimenezApplicationTests.
- Code Editor:** The file "M8E1_CRUD_Tests_Descuento.java" is open. It contains Java code for testing the Descuento repository. The code includes methods for inserting a new discount, searching by ID, and searching all discounts.
- Toolbars and Status Bar:** Standard IDE toolbars and a status bar at the bottom indicating the current file path: "M8E1_Francisco_Jimenez > src > main > resources > data.sql".

Captura 14.2. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1 CRUD_Tests_Descuento.java (parte 3)



```
public class M8E1_CRUD_Tests_Descuento {
    void actualizarUnDescuentoTest() {
        System.out.println("\n=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 4: Actualizar un descuento");
        System.out.println("-----");

        // Búsqueda de descuento para actualizar
        Optional<Descuento> optional = descuentoRepository.findById(DESCUENTO_ID_ELIMINADO);
        if (optional.isPresent()) {
            Descuento descuento = optional.get();
            System.out.println("Descuento ANTES de actualizar: " + descuento);

            // Actualización del descuento
            descuento.setPorcentaje(PORCENTAJE_ACTUALIZADO);
            descuentoRepository.save(descuento);
            System.out.println("Descuento DESPUÉS de actualizar: " + descuento);
        } else {
            System.out.println("No se encontró el descuento con ID " + DESCUENTO_ID_ELIMINADO + " para actualizar.");
        }
    }

    @Test
    @Order(5)
    void eliminarUnDescuentoTest() {
        System.out.println("\n=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 5: Eliminar un descuento");
        System.out.println("-----");

        // Verificación de la categoría antes de eliminar
        Optional<Descuento> optional = descuentoRepository.findById(DESCUENTO_ID_ELIMINADO);
        if (optional.isPresent()) {
            Descuento descuentoAEliminar = optional.get();
            System.out.println("Descuento a eliminar: " + descuentoAEliminar);

            // Lista antes de la eliminación
            System.out.println("\nLista de descuentos ANTES de la eliminación:");
            descuentoRepository.findAll().forEach(System.out::println);

            // Eliminación
            descuentoRepository.delete(descuentoAEliminar);
        }
    }
}
```

Captura 14.3. Salida de la ejecución de la suite de pruebas M8E1 CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Descuento.java (parte 4)

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" directory contains "main" and "test" packages. "main" has "M8E1" and "datos" sub-directories, which further contain "Categoria", "CategoriaRepository", "Descuento", "DescuentoRepository", "Marca", "MarcaRepository", "Rol", "RolRepository", and "M8E1FranciscoJimenezApplication". "test" has "java" and "resources" sub-directories, with "java" containing "M8E1", "M8E1_CRUD_Tests_Categoría", "M8E1_CRUD_Tests_Descuento", "M8E1_CRUD_Tests_Marca", "M8E1_CRUD_Tests_Rol", and "M8E1FranciscoJimenezApplicationTests".
- Code Editor:** The active tab is "M8E1_CRUD_Tests_Descuento.java". The code implements several test methods for the Descuento repository:
 - `void eliminarUnDescuentoTest()`: Prints the list of discounts after deletion.
 - `void buscarVariosDescuentosTest()`: Prints the list of discounts for specific IDs.
 - `void insertarVariosDescuentosTest()`: Prints the list of discounts and inserts new ones.
- Bottom Status Bar:** Shows the current file path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Descuento > buscarUnDescuentoPorIdTest.

Captura 14.4. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Descuento.java (parte 5)

The screenshot shows an IDE interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" folder contains "main" and "test" packages. "main" has "java" and "datos" subfolders, which contain various repository and application classes. "test" has "java" and "resources" subfolders, with "M8E1" and "M8E1FranciscoJimenezApplication" test classes.
- Code Editor:** The active file is "M8E1_CRUD_Tests_Descuento.java". The code implements two test methods: `insertarVariosDescuentosTest()` and `eliminarVariosDescuentosTest()`.
 - `insertarVariosDescuentosTest()` creates a new `Descuento` object with id 12, percentage 25, and name "Descuento Corporativo", then saves it to the repository.
 - `eliminarVariosDescuentosTest()` prints a separator line, lists the number of students, and then deletes all students from the repository.
- Status Bar:** Shows the current path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Descuento > buscarVariosDescuentosTest.

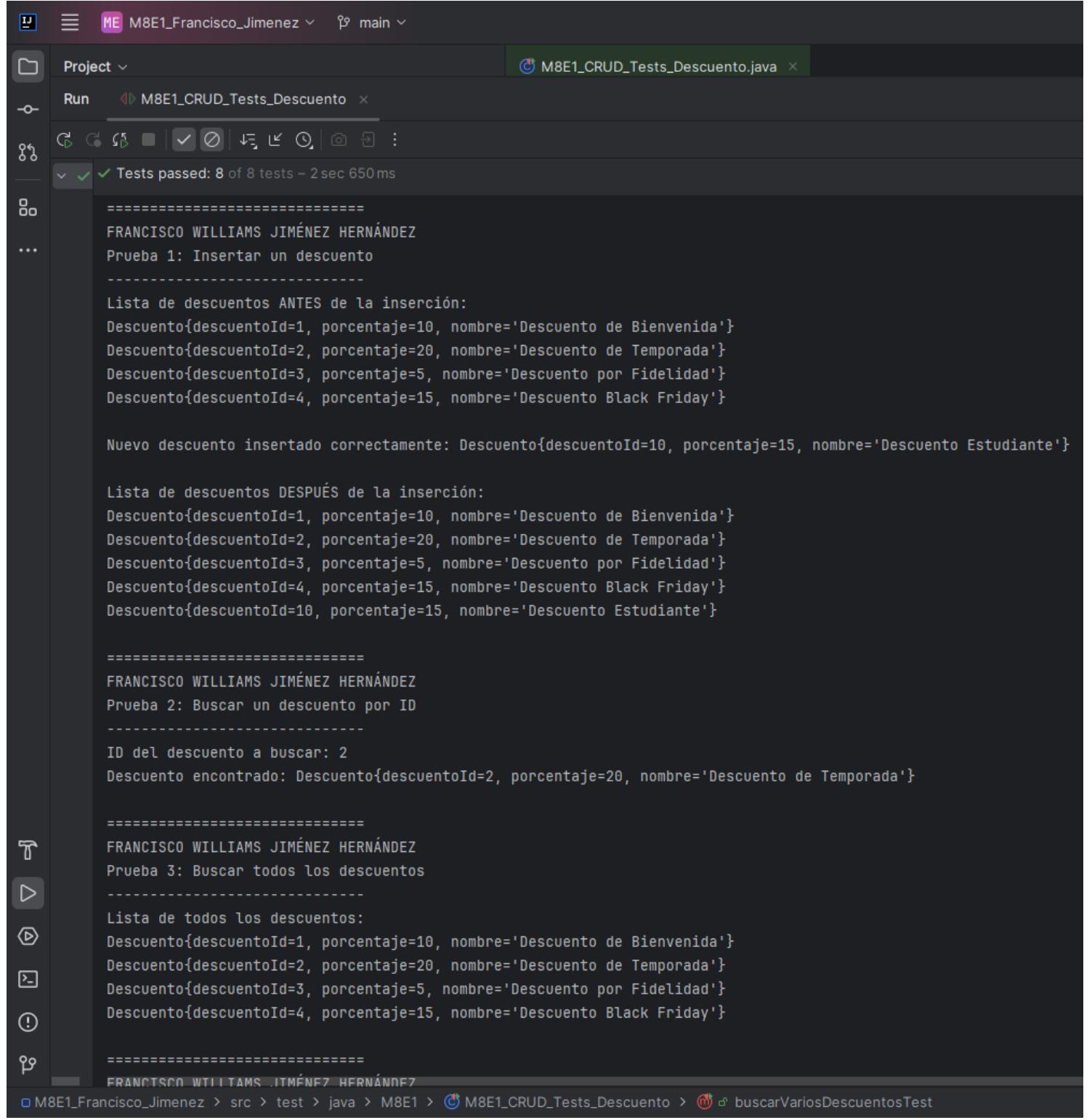
Captura 14.5. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

A continuación, se muestra la salida en consola de la ejecución de la clase de pruebas **M8E1_CRUD_Tests_Descuento**

Consola



The screenshot shows the IntelliJ IDEA interface with the project 'M8E1_Francisco_Jimenez' selected. The current file is 'M8E1_CRUD_Tests_Descuento.java'. The 'Run' tool window is open, showing the test results: 'Tests passed: 8 of 8 tests - 2 sec 650 ms'. The console output displays the following test results:

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 1: Insertar un descuento
-----
Lista de descuentos ANTES de la inserción:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}

Nuevo descuento insertado correctamente: Descuento{descuentoId=10, porcentaje=15, nombre='Descuento Estudiante'}

Lista de descuentos DESPUÉS de la inserción:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}
Descuento{descuentoId=10, porcentaje=15, nombre='Descuento Estudiante'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 2: Buscar un descuento por ID
-----
ID del descuento a buscar: 2
Descuento encontrado: Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 3: Buscar todos los descuentos
-----
Lista de todos los descuentos:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}
```

The status bar at the bottom indicates the path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Descuento > buscarVariosDescuentosTest.

Captura 15.1. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

The screenshot shows the IntelliJ IDEA interface during the execution of the test class `M8E1_CRUD_Tests_Descuento.java`. The top bar displays the project name "M8E1_Francisco_Jimenez" and the file "main". The left sidebar includes icons for file, project, run, and others. The main window shows the test results and the output of the executed code.

Tests passed: 8 of 8 tests – 2 sec 650 ms

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 4: Actualizar un descuento
-----
Descuento ANTES de actualizar: Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento DESPUÉS de actualizar: Descuento{descuentoId=1, porcentaje=30, nombre='Descuento de Bienvenida'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 5: Eliminar un descuento
-----
Descuento a eliminar: Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}

Lista de descuentos ANTES de la eliminación:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}

Lista de descuentos DESPUÉS de la eliminación:
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 6: Buscar varios descuentos por ID
-----
Descuentos encontrados:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
```

=====

FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ

Prueba 7: Insertar varios descuentos

=====

Descuentos después de la inserción:

M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Descuento > buscarVariosDescuentosTest

Captura 15.2. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

The screenshot shows an IDE interface with a dark theme. The top bar displays the project name "M8E1_Francisco_Jimenez" and the file "main". A tab labeled "M8E1_CRUD_Tests_Descuento.java" is open. The main window displays the output of the test execution:

```
Tests passed: 8 of 8 tests - 2 sec 650 ms

Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 7: Insertar varios descuentos
-----
Descuentos después de la inserción:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}
Descuento{descuentoId=11, porcentaje=12, nombre='Descuento por Amigo'}
Descuento{descuentoId=12, porcentaje=25, nombre='Descuento Corporativo'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 8: Eliminar varios descuentos
-----
Descuentos ANTES de la eliminación:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}

Descuentos DESPUÉS de la eliminación:
Descuento{descuentoId=1, porcentaje=10, nombre='Descuento de Bienvenida'}
Descuento{descuentoId=2, porcentaje=20, nombre='Descuento de Temporada'}
Descuento{descuentoId=3, porcentaje=5, nombre='Descuento por Fidelidad'}
Descuento{descuentoId=4, porcentaje=15, nombre='Descuento Black Friday'}
```

At the bottom, there is a log message from the application:

```
2024-09-26T15:10:42.403+05:00 INFO 7452 --- [M8E1_Francisco_Jimenez] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2024-09-26T15:10:42.403+05:00 INFO 7452 --- [M8E1_Francisco_Jimenez] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2024-09-26T15:10:42.419+05:00 INFO 7452 --- [M8E1_Francisco_Jimenez] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
```

The status bar at the bottom shows the path "M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Descuento" and the test name "buscarVariosDescuentosTest".

Captura 15.3. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Descuento.

Francisco Williams Jiménez Hernández

Ejercicio 1

4.3. CRUD para Marca

Archivo: M8E1_CRUD_Tests_Marca.java (parte 1)

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" directory contains "main" and "test" packages. "main/java" includes "M8E1" and "datos" (containing Category, Descuento, DescuentoRepository, Marca, MarcaRepository, Rol, and RolRepository). "test/java" includes "M8E1" (containing M8E1_CRUD_Tests_Categoría, M8E1_CRUD_Tests_Descuento, M8E1_CRUD_Tests_Marca, M8E1_CRUD_Tests_Rol, and M8E1FranciscoJimenezApplicationTests).
- M8E1_CRUD_Tests_Marca.java Content:**

```
/*
 * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
 * Asignación: Ejercicio 1
 * Fecha de realización: 24/09/2024
 * Descripción: Pruebas para el CRUD de la entidad Marca
 */
@SpringBootTest
@Sql({"/schema.sql", "/data.sql"}) // Asegurarse de tener una base limpia en cada ejecución de prueba unitaria - Carga esquema y datos de inicio
@TestMethodOrder(MethodOrderer.OrderAnnotation.class) // Ejecuta las pruebas en orden para un resultado más predecible
public class M8E1_CRUD_Tests_Marca {

    private final String ALUMNO = "FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ"; 8 usages
    private final int MARCA_ID_INSERTADA = 10; 3 usages
    private final int MARCA_ID_ELIMINADA = 1; 5 usages
    private final int MARCA_ID_BUSCAR = 2; 3 usages
    private final String MARCA_NUEVO_NOMBRE = "Puma"; 2 usages

    @Autowired 19 usages
    MarcaRepository marcaRepository;

    @Test
    @Order(1)
    void insertarUnaMarcaTest() {
        System.out.println("=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 1: Insertar una marca");
        System.out.println("=====");

        // Lista de marcas antes de la inserción
        System.out.println("Lista de marcas ANTES de la inserción:");
        marcaRepository.findAll().forEach(System.out::println);

        // Inserción de nueva marca
        Optional<Marca> existingMarca = marcaRepository.findById(MARCA_ID_INSERTADA);
        if (existingMarca.isPresent()) {
            System.out.println("La marca con ID " + MARCA_ID_INSERTADA + " ya existe. No se insertará.");
        } else {
            Marca nuevaMarca = new Marca(MARCA_ID_INSERTADA, MARCA_NUEVO_NOMBRE);
            marcaRepository.save(nuevaMarca);
            System.out.println("\nNueva marca insertada correctamente: " + nuevaMarca);
        }
    }
}
```
- Status Bar:** Shows "71:70" at the bottom right.

Captura 16.1. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Marca.java (parte 2)

The screenshot shows an IDE interface with the following details:

- Project View:** Shows the project structure under "M8E1_FRANCISCO_JIMENEZ". The "src" folder contains "main" and "test" packages. "main" has "M8E1" and "datos" subfolders, which contain various repository classes like CategoríaRepository, DescuentoRepository, MarcaRepository, etc. "test" has "java" and "resources" subfolders, with "java" containing test classes for Categoría, Descuento, Marca, Rol, and M8E1FranciscoJimenezApplicationTests.
- M8E1_CRUD_Tests_Marca.java Content:**

```
public class M8E1_CRUD_Tests_Marca {
    void insertarUnaMarcaTest() {
        // Lista de marcas después de la inserción
        System.out.println("\nlista de marcas DESPUES de la inserción:");
        marcaRepositorio.findAll().forEach(System.out::println);
    }

    @Test
    @Order(2)
    void buscarUnaMarcaPorIdTest() {
        System.out.println("\n=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 2: Buscar una marca por ID");
        System.out.println("-----");
        System.out.println("ID de la marca a buscar: " + MARCA_ID_BUSCAR);

        // Búsqueda por ID
        Optional<Marca> optional = marcaRepositorio.findById(MARCA_ID_BUSCAR);
        if (optional.isPresent()) {
            System.out.println("Marca encontrada: " + optional.get());
        } else {
            System.out.println("No se encontró una marca con ID " + MARCA_ID_BUSCAR + ".");
        }
    }

    @Test
    @Order(3)
    void buscarTodasLasMarcasTest() {
        System.out.println("\n=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 3: Buscar todas las marcas");
        System.out.println("-----");

        // Lista de todas las marcas
        Iterable<Marca> marcas = marcaRepositorio.findAll();
        System.out.println("Lista de todas las marcas:");
        marcas.forEach(System.out::println);
    }

    @Test
    @Order(4)
    void actualizarUnaMarcaTest() {
    }
}
```
- Status Bar:** Shows the current file path: M8E1_FRANCISCO_JIMENEZ > src > test > java > M8E1 > M8E1_CRUD_Tests_Marca > buscarUnaMarcaPorIdTest.

Captura 16.2. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Marca.java (parte 3)

The screenshot shows an IDE interface with the following details:

- Project:** M8E1_Francisco_Jimenez
- File:** M8E1_CRUD_Tests_Marca.java
- Code Content:** The code is a Java test class named M8E1_CRUD_Tests_Marca. It contains methods for updating, deleting, and listing a brand. The code uses System.out.println for output and Optional<Marca> for handling optional values.
- Test Methods:** The class includes @Test annotated methods for updating and deleting a brand.
- IDE Navigation:** The left sidebar shows the project structure with main and test packages containing various repository and service classes.
- Bottom Status Bar:** Shows the current file path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Marca > buscarUnaMarcaPorIdTest

Captura 16.3. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Marca.java (parte 4)

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src/main/java" package contains several classes: Categoría, CategoríaRepository, Descuento, DescuentoRepository, Marca, MarcaRepository, Rol, RolRepository, and M8E1FranciscoJimenezApplication. The "src/test/java" package contains test classes: M8E1_CRUD_Tests_Categoría, M8E1_CRUD_Tests_Descuento, M8E1_CRUD_Tests_Marca (which is the active file), M8E1_CRUD_Tests_Rol, and M8E1FranciscoJimenezApplicationTests.
- M8E1_CRUD_Tests_Marca.java Content:**

```
public class M8E1_CRUD_Tests_Marca {
    void eliminarUnaMarcaTest() {
        // Eliminación
        marcaRepositorio.delete(marcaAEliminar);

        // Lista después de la eliminación
        System.out.println("\nLista de marcas DESPUÉS de la eliminación:");
        marcaRepositorio.findAll().forEach(System.out::println);
    } else {
        System.out.println("No se encontró la marca con ID " + MARCA_ID_ELIMINADA + " para eliminar.");
    }
}

@Test
@Order(6)
void buscarVariasMarcasTest() {
    System.out.println("\n=====");
    System.out.println(ALUMNO);
    System.out.println("Prueba 6: Buscar varias marcas por ID");
    System.out.println("-----");

    // Lista de IDs a buscar
    List<Integer> ids = Arrays.asList(1, 2, 3);
    Iterable<Marca> marcas = marcaRepositorio.findAllById(ids);
    System.out.println("Marcas encontradas:");
    marcas.forEach(System.out::println);
}

@Test
@Order(7)
void insertarVariasMarcasTest() {
    System.out.println("\n=====");
    System.out.println(ALUMNO);
    System.out.println("Prueba 7: Insertar varias marcas");
    System.out.println("-----");

    // Lista de marcas a insertar
    List<Marca> marcas = Arrays.asList(
        new Marca(marcId: 11, nombre: "Under Armour"),
        new Marca(marcId: 12, nombre: "Reebok")
    );
}
```
- Bottom Status Bar:** Shows the current path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Marca > buscarUnaMarcaPorIdTest.

Captura 16.4. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Marca.java (parte 5)

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src/main/java/M8E1" package contains several test classes: M8E1_CRUD_Tests_Categoría, M8E1_CRUD_Tests_Descuento, M8E1_CRUD_Tests_Marca (which is highlighted), M8E1_CRUD_Tests_Rol, and M8E1FranciscoJimenezApplicationTests.
- M8E1_CRUD_Tests_Marca.java Content:**

```
26  public class M8E1_CRUD_Tests_Marca {
27      void insertarVariasMarcasTest() {
28          new Marca( marcad: 12, nombre: "Reebok" );
29      }
30
31      // Inserción de varias marcas
32      marcaRepository.saveAll(marcas);
33
34      // Verificación
35      System.out.println("Marcas después de la inserción:");
36      marcaRepository.findAll().forEach(System.out::println);
37  }
38
39  @Test
40  @Order(8)
41  void eliminarVariasMarcasTest() {
42      System.out.println("\n=====");
43      System.out.println(ALUMNO);
44      System.out.println("Prueba 8: Eliminar varias marcas");
45      System.out.println("-----");
46
47      // Lista de IDs a eliminar
48      List<Integer> ids = Arrays.asList(11, 12);
49
50      // Lista antes de la eliminación
51      System.out.println("Marcas ANTES de la eliminación:");
52      marcaRepository.findAll().forEach(System.out::println);
53
54      // Eliminación de varias marcas
55      marcaRepository.deleteAllById(ids);
56
57      // Verificación después de eliminar
58      System.out.println("\nMarcas DESPUÉS de la eliminación:");
59      marcaRepository.findAll().forEach(System.out::println);
60  }
61 }
```
- Bottom Status Bar:** Shows the current file path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Marca > buscarUnaMarcaPorIdTest.

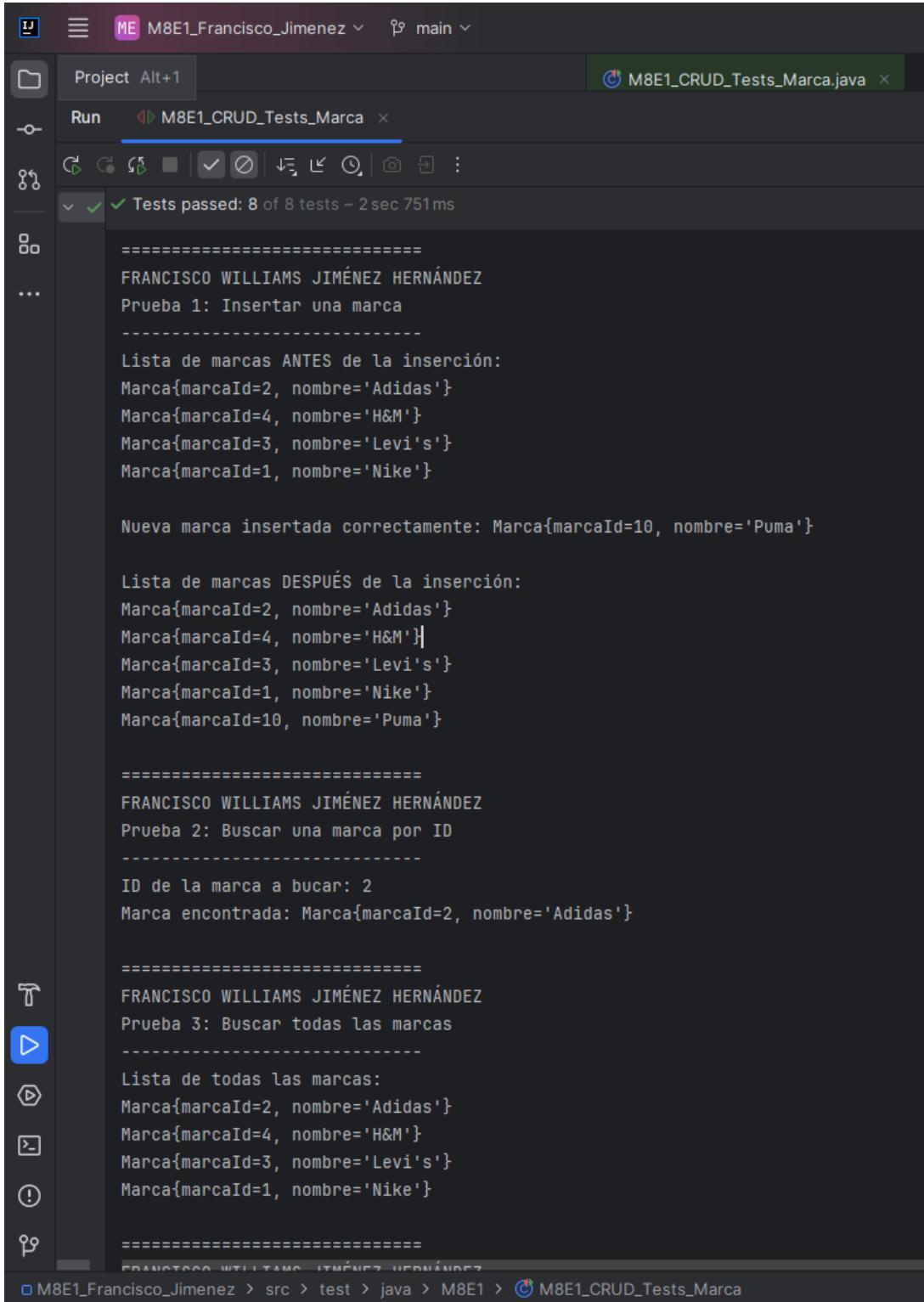
Captura 16.5. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

A continuación, se muestra la salida en consola de la ejecución de la clase de pruebas **M8E1_CRUD_Tests_Marca**

Consola



The screenshot shows the IntelliJ IDEA interface with the project 'M8E1_Francisco_Jimenez' selected. The 'Run' tab is active, showing the output of the test class 'M8E1_CRUD_Tests_Marca'. The console output is as follows:

```
Tests passed: 8 of 8 tests – 2 sec 751ms
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 1: Insertar una marca
-----
Lista de marcas ANTES de la inserción:
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}

Nueva marca insertada correctamente: Marca{marcaId=10, nombre='Puma'}

Lista de marcas DESPUÉS de la inserción:
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}
Marca{marcaId=10, nombre='Puma'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 2: Buscar una marca por ID
-----
ID de la marca a buscar: 2
Marca encontrada: Marca{marcaId=2, nombre='Adidas'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 3: Buscar todas las marcas
-----
Lista de todas las marcas:
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}
```

Captura 17.1. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

The screenshot shows the execution results of the Java test class `M8E1_CRUD_Tests_Marca.java`. The test suite consists of 8 tests, all of which passed. The output displays the results of four specific tests related to updating, deleting, searching, and inserting multiple brands.

```
Tests passed: 8 of 8 tests - 2 sec 751ms
```

=====

```
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 4: Actualizar una marca
-----
Marca ANTES de actualizar: Marca{marcaId=1, nombre='Nike'}
Marca DESPUÉS de actualizar: Marca{marcaId=1, nombre='Puma'}
```

=====

```
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 5: Eliminar una marca
-----
Marca a eliminar: Marca{marcaId=1, nombre='Nike'}
```

Lista de marcas ANTES de la eliminación:

```
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}
```

Lista de marcas DESPUÉS de la eliminación:

```
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
```

=====

```
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 6: Buscar varias marcas por ID
-----
Marcas encontradas:
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}
```

=====

```
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 7: Insertar varias marcas
-----
Marcas después de la inserción:
```

M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Marca

Captura 17.2. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

```
M8E1_Francisco_Jimenez main M8E1_CRUD_Tests_Marca M8E1_CRUD_Tests_Marca.java
Project Run M8E1_CRUD_Tests_Marca M8E1_CRUD_Tests_Marca.java
Tests passed: 8 of 8 tests - 2sec 751ms
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 7: Insertar varias marcas
-----
Marcas después de la inserción:
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}
Marca{marcaId=12, nombre='Reebok'}
Marca{marcaId=11, nombre='Under Armour'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 8: Eliminar varias marcas
-----
Marcas ANTES de la eliminación:
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}

Marcas DESPUÉS de la eliminación:
Marca{marcaId=2, nombre='Adidas'}
Marca{marcaId=4, nombre='H&M'}
Marca{marcaId=3, nombre='Levi's'}
Marca{marcaId=1, nombre='Nike'}
2024-09-26T15:20:47.803-05:00 INFO 12236 --- [M8E1_Francisco_Jimenez] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2024-09-26T15:20:47.805-05:00 INFO 12236 --- [M8E1_Francisco_Jimenez] ionShutdownHook com.zaxxer.hikari.HikariDataSource      : HikariPool-1 - Shutdown initiated...
2024-09-26T15:20:47.811-05:00 INFO 12236 --- [M8E1_Francisco_Jimenez] ionShutdownHook com.zaxxer.hikari.HikariDataSource      : HikariPool-1 - Shutdown completed.

Process finished with exit code 0
```

Captura 17.3. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Marca.

Francisco Williams Jiménez Hernández

Ejercicio 1

4.4.CRUD para Rol

Archivo: M8E1_CRUD_Tests_Rol.java (parte 1)

The screenshot shows the IntelliJ IDEA interface with the project 'M8E1_Francisco_Jimenez' open. The left sidebar displays the project structure, including modules like 'main' and 'test', and various Java classes and resources. The right pane shows the code editor for 'M8E1_CRUD_Tests_Rol.java'. The code is a test class for the 'Rol' entity, using Spring Boot annotations like @SpringBootTest and @TestMethodOrder. It includes constants for roles and their descriptions, and a test method 'insertarUnRolTest' that prints the current list of roles before insertion and then inserts a new role with ID 10, checking if it exists in the database.

```
/*
 * Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
 * Asignación: Ejercicio 1
 * Fecha de realización: 24/09/2024
 * Descripción: Pruebas para el CRUD de la entidad Rol
 */
@SpringBootTest
@Sql({"/schema.sql", "/data.sql"}) // Asegurarse de tener una base limpia en cada ejecución de prueba unitaria - Carga esquema y datos de inicio
@TestMethodOrder(MethodOrderer.OrderAnnotation.class) // Ejecuta las pruebas en orden para un resultado más predecible
public class M8E1_CRUD_Tests_Rol {

    private final String ALUMNO = "FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ"; 8 usages
    private final int ROL_ID_INSERTADO = 10; // ID para el nuevo rol 3 usages
    private final int ROL_ID_ELIMINADO = 1; // ID para el rol a eliminar 5 usages
    private final int ROL_ID_BUSCAR = 2; // ID para el rol a buscar 3 usages
    private final String ROL_NUEVO_NOMBRE = "Gestor"; 2 usages
    private final String ROL_NUEVA_DESCRIPCION = "Gestiona tareas administrativas"; 2 usages

    @Autowired 19 usages
    RolRepository rolRepository;

    @Test
    @Order(1)
    void insertarUnRolTest() {
        System.out.println("=====");
        System.out.println(ALUMNO);
        System.out.println("Prueba 1: Insertar un rol");
        System.out.println("-----");

        // Lista de roles antes de la inserción
        System.out.println("Lista de roles ANTES de la inserción:");
        rolRepository.findAll().forEach(System.out::println);

        Optional<Rol> existingRol = rolRepository.findById(ROL_ID_INSERTADO);
        if (existingRol.isPresent()) {
            System.out.println("El rol con ID '" + ROL_ID_INSERTADO + "' ya existe. No se insertará.");
        } else {
            Rol nuevoRol = new Rol(ROL_ID_INSERTADO, ROL_NUEVO_NOMBRE, ROL_NUEVA_DESCRIPCION);
            rolRepository.save(nuevoRol);
            System.out.println("\nRol nuevo insertado correctamente: " + nuevoRol);
        }
    }
}
```

Captura 18.1. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Rol.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Rol.java (parte 2)

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" directory contains "main" and "test" packages. "main" has "M8E1" with "datos" containing various repository classes like Categoría, CategoríaRepository, Descuento, DescuentoRepository, Marca, MarcaRepository, Rol, and RolRepository. It also includes "M8E1FranciscoJimenezApplication". "test" has "java" with "M8E1" containing test classes: M8E1_CRUD_Tests_Categoría, M8E1_CRUD_Tests_Descuento, M8E1_CRUD_Tests_Marca, M8E1_CRUD_Tests_Rol, and M8E1FranciscoJimenezApplicationTests.
- M8E1_CRUD_Tests_Rol.java Content:** The code is annotated with JUnit annotations (@Test, @Order). It includes methods for inserting a role, searching by ID, searching all roles, and updating a role.
- Status Bar:** Shows the current file path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Rol.

Captura 18.2. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Rol.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Rol.java (parte 3)

The screenshot shows an IDE interface with the following details:

- Project:** M8E1_Francisco_Jimenez
- File:** M8E1_CRUD_Tests_Rol.java
- Code Content (Excerpt):**

```
26  public class M8E1_CRUD_Tests_Rol {
96    void actualizarUnRolTest() {
99      System.out.println("Prueba 4: Actualizar un rol");
100     System.out.println("-----");
101
102     Optional<Rol> optional = rolRepository.findById(ROL_ID_ELIMINADO);
103     if (optional.isPresent()) {
104       Rol rol = optional.get();
105       System.out.println("Rol ANTES de actualizar: " + rol);
106
107       rol.setNombre(ROL_NUEVO_NOMBRE);
108       rol.setDescripcion(ROL_NUEVA_DESCRIPCION);
109       rolRepository.save(rol);
110
111       Rol rolActualizado = rolRepository.findById(ROL_ID_ELIMINADO).get();
112       System.out.println("Rol DESPUES de actualizar: " + rolActualizado);
113     } else {
114       System.out.println("No se encontró el rol con ID " + ROL_ID_ELIMINADO + " para actualizar.");
115     }
116
117     @Test
118     @Order(5)
119     void eliminarUnRolTest() {
120       System.out.println("=====");
121       System.out.println(ALUMNO);
122       System.out.println("Prueba 5: Eliminar un rol");
123       System.out.println("-----");
124
125       Optional<Rol> optional = rolRepository.findById(ROL_ID_ELIMINADO);
126       if (optional.isPresent()) {
127         Rol rolAEliminar = optional.get();
128         System.out.println("Rol a eliminar: " + rolAEliminar);
129
130         // Lista antes de la eliminación
131         System.out.println("\nLista de roles ANTES de la eliminación:");
132         rolRepository.findAll().forEach(System.out::println);
133
134         // Eliminación
135         rolRepository.delete(rolAEliminar);
136
137         // Lista después de la eliminación
138       }
139     }
140   }
141 }
```

- IDE Tools:** The interface includes standard tools like Project, Build, Run, and Help.
- Bottom Status Bar:** Shows the current path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Rol > buscarUnRolPorIdTest

Captura 18.3. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Rol.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Rol.java (parte 4)

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" directory contains "main" and "test" packages. "main" has "M8E1" with "datos" containing various repository classes like Categoría, CategoríaRepository, Descuento, DescuentoRepository, Marca, MarcaRepository, Rol, and RolRepository. It also includes "M8E1FranciscoJimenezApplication", "resources" (static, templates, application.properties, data.sql, schema.sql), and "M8E1" with test classes M8E1_CRUD_Tests_Categoría, M8E1_CRUD_Tests_Descuento, M8E1_CRUD_Tests_Marca, M8E1_CRUD_Tests_Rol, and M8E1FranciscoJimenezApplicationTests.
- Code Editor:** The current file is "M8E1_CRUD_Tests_Rol.java". The code implements several test methods for the Rol entity:
 - eliminarUnRoleTest():** Prints a message indicating the deletion of roles after finding all roles.
 - buscarVariosRolesTest():** Prints messages for searching multiple roles by ID. It uses a list of IDs [1, 2, 3] and prints the found roles.
 - insertarVariosRolesTest():** Prints messages for inserting multiple roles. It creates two new roles (rolId 11, nombre "Supervisor", descripción "Supervisa actividades de las operaciones" and rolId 12, nombre "Operador", descripción "Gestiona las operaciones diarias") and saves them.
 - buscarUnRolPorIdTest():** Prints a message for searching a role by ID.
- Status Bar:** Shows the path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Rol > buscarUnRolPorIdTest.

Captura 18.4. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Rol.

Francisco Williams Jiménez Hernández

Ejercicio 1

Archivo: M8E1_CRUD_Tests_Rol.java (parte 5)

The screenshot shows an IDE interface with the following details:

- Project View:** Shows the project structure under "M8E1_Francisco_Jimenez". The "src" folder contains "main" and "test" packages. "main" has "M8E1" and "datos" subfolders, which contain various repository and application classes. "test" has "M8E1" and "M8E1FranciscoJimenezApplicationTests" subfolders, with "M8E1_Crud_Tests_Rol" being the active file.
- M8E1_Crud_Tests_Rol.java Content:**

```
public class M8E1_Crud_Tests_Rol {
    void insertarVariosRolesTest() {
        new Rol(rolId: 12, nombre: "Operador", descripción: "Gestiona las operaciones diarias");
    }

    // Inserción de varios roles
    rolRepository.saveAll(roles);

    // Verificación
    System.out.println("Roles después de la inserción:");
    rolRepository.findAll().forEach(System.out::println);
}

@Test
@Order(8)
void eliminarVariosRolesTest() {
    System.out.println("\n=====");
    System.out.println(ALUMNO);
    System.out.println("Prueba 8: Eliminar varios roles");
    System.out.println("-----");

    // Lista de IDs a eliminar
    List<Integer> ids = Arrays.asList(11, 12);

    // Lista antes de la eliminación
    System.out.println("Roles ANTES de la eliminación:");
    rolRepository.findAll().forEach(System.out::println);

    // Eliminación de varios roles
    rolRepository.deleteAllById(ids);

    // Verificación después de la eliminación
    System.out.println("\nRoles DESPUÉS de la eliminación:");
    rolRepository.findAll().forEach(System.out::println);
}
```
- Bottom Status Bar:** Displays the path "M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_Crud_Tests_Rol" and a search bar "buscarUnRolPorIdTest".

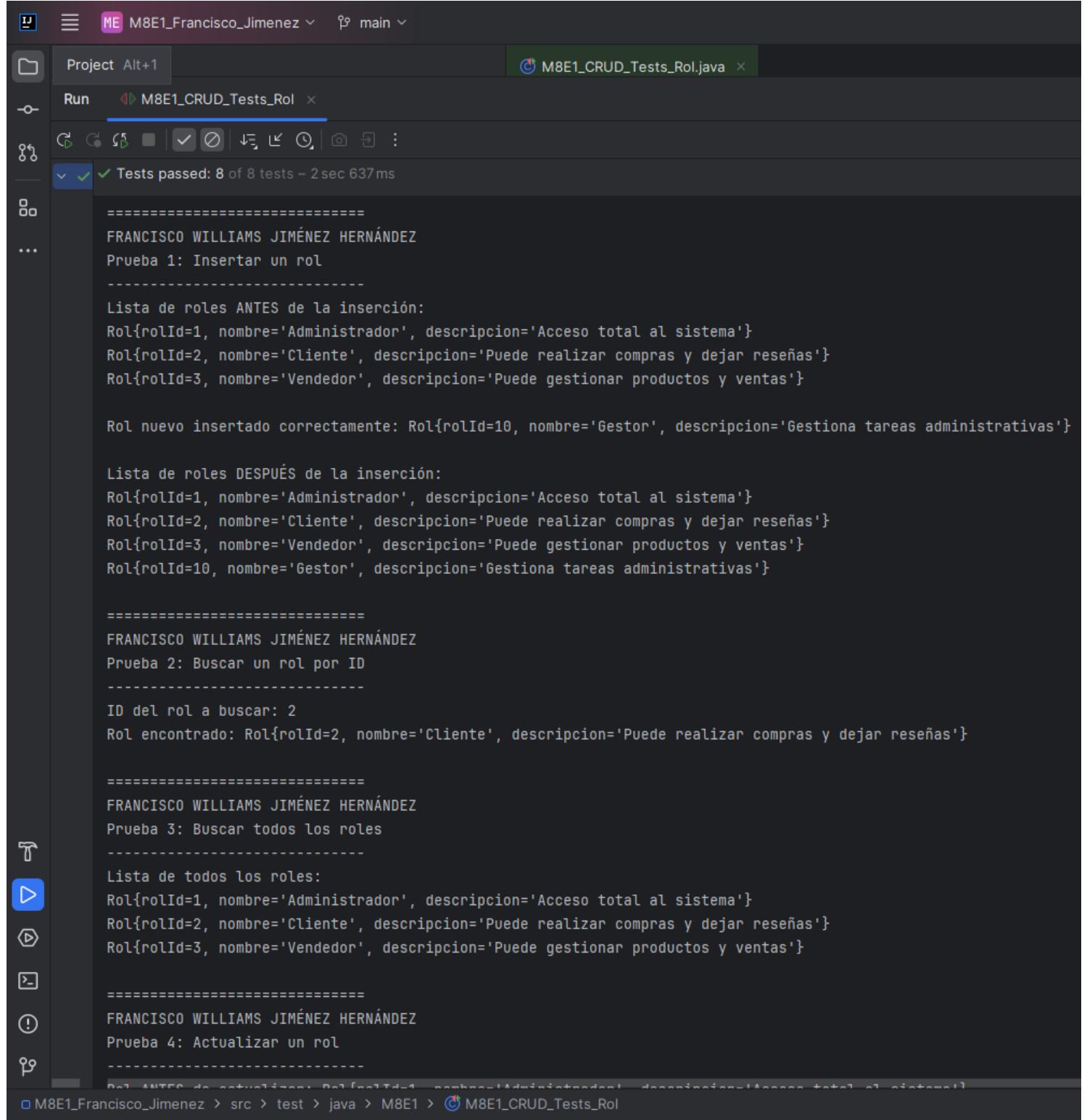
Captura 18.5. Salida de la ejecución de la suite de pruebas M8E1_Crud_Tests_Rol.

Francisco Williams Jiménez Hernández

Ejercicio 1

A continuación, se muestra la salida en consola de la ejecución de la clase de pruebas **M8E1_CRUD_Tests_Rol**

Consola



The screenshot shows the IntelliJ IDEA interface with the project 'M8E1_Francisco_Jimenez' selected. The 'Run' tab is active, showing the run configuration 'M8E1_CRUD_Tests_Rol'. The console output displays the results of the test suite:

```
Tests passed: 8 of 8 tests - 2 sec 637ms
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 1: Insertar un rol
-----
Lista de roles ANTES de la inserción:
Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{rolId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}

Rol nuevo insertado correctamente: Rol{rolId=10, nombre='Gestor', descripcion='Gestionar tareas administrativas'}

Lista de roles DESPUÉS de la inserción:
Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{rolId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}
Rol{rolId=10, nombre='Gestor', descripcion='Gestionar tareas administrativas'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 2: Buscar un rol por ID
-----
ID del rol a buscar: 2
Rol encontrado: Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 3: Buscar todos los roles
-----
Lista de todos los roles:
Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{rolId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 4: Actualizar un rol
-----
Rol ANTES de actualización: Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
```

Captura 19.1. Salida de la ejecución de la suite de pruebas **M8E1_CRUD_Tests_Rol**.

Francisco Williams Jiménez Hernández

Ejercicio 1

The screenshot shows the IntelliJ IDEA interface during the execution of the test class `M8E1_CRUD_Tests_Rol.java`. The top bar displays the project name "M8E1_Francisco_Jimenez" and the file "main". The "Run" tab is selected, showing the output of the tests.

Test Results:

```
Tests passed: 8 of 8 tests - 2 sec 637 ms
```

Test Output:

```
Rol ANTES de actualizar: Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol DESPUÉS de actualizar: Rol{rolId=1, nombre='Gestor', descripcion='Gestionar tareas administrativas'}
```

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 5: Eliminar un rol
=====
```

```
Rol a eliminar: Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
```

```
Lista de roles ANTES de la eliminación:
Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{rolId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}
```

```
Lista de roles DESPUÉS de la eliminación:
Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{rolId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}
```

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 6: Buscar varios roles por ID
=====
```

```
Roles encontrados:
Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{rolId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}
```

```
=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba 7: Insertar varios roles
=====
```

```
Roles después de la inserción:
Rol{rolId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{rolId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{rolId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}
Rol{rolId=11, nombre='Supervisor', descripcion='Supervisa actividades de las operaciones'}
Rol{rolId=12, nombre='Operador', descripcion='Gestiona las operaciones diarias'}
```

The bottom status bar shows the path: M8E1_Francisco_Jimenez > src > test > java > M8E1 > M8E1_CRUD_Tests_Rol.

Captura 19.2. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Rol.

Francisco Williams Jiménez Hernández

Ejercicio 1

```
Tests passed: 8 of 8 tests - 2 sec 637 ms
Prueba 6: Buscar varios roles por ID
-----
Roles encontrados:
Rol{roleId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{roleId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{roleId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ

Prueba 7: Insertar varios roles
-----
Roles después de la inserción:
Rol{roleId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{roleId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{roleId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}
Rol{roleId=11, nombre='Supervisor', descripcion='Supervisa actividades de las operaciones'}
Rol{roleId=12, nombre='Operador', descripcion='Gestiona las operaciones diarias'}

=====
FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ

Prueba 8: Eliminar varios roles
-----
Roles ANTES de la eliminación:
Rol{roleId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{roleId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{roleId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}

Roles DESPUÉS de la eliminación:
Rol{roleId=1, nombre='Administrador', descripcion='Acceso total al sistema'}
Rol{roleId=2, nombre='Cliente', descripcion='Puede realizar compras y dejar reseñas'}
Rol{roleId=3, nombre='Vendedor', descripcion='Puede gestionar productos y ventas'}
```

2024-09-20T15:30:43.841-05:00 INFO 14200 --- [M8E1_Francisco_Jimenez] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'

2024-09-20T15:30:43.841-05:00 INFO 14200 --- [M8E1_Francisco_Jimenez] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...

2024-09-20T15:30:43.841-05:00 INFO 14200 --- [M8E1_Francisco_Jimenez] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.

Process finished with exit code 0

Captura 19.3. Salida de la ejecución de la suite de pruebas M8E1_CRUD_Tests_Rol.