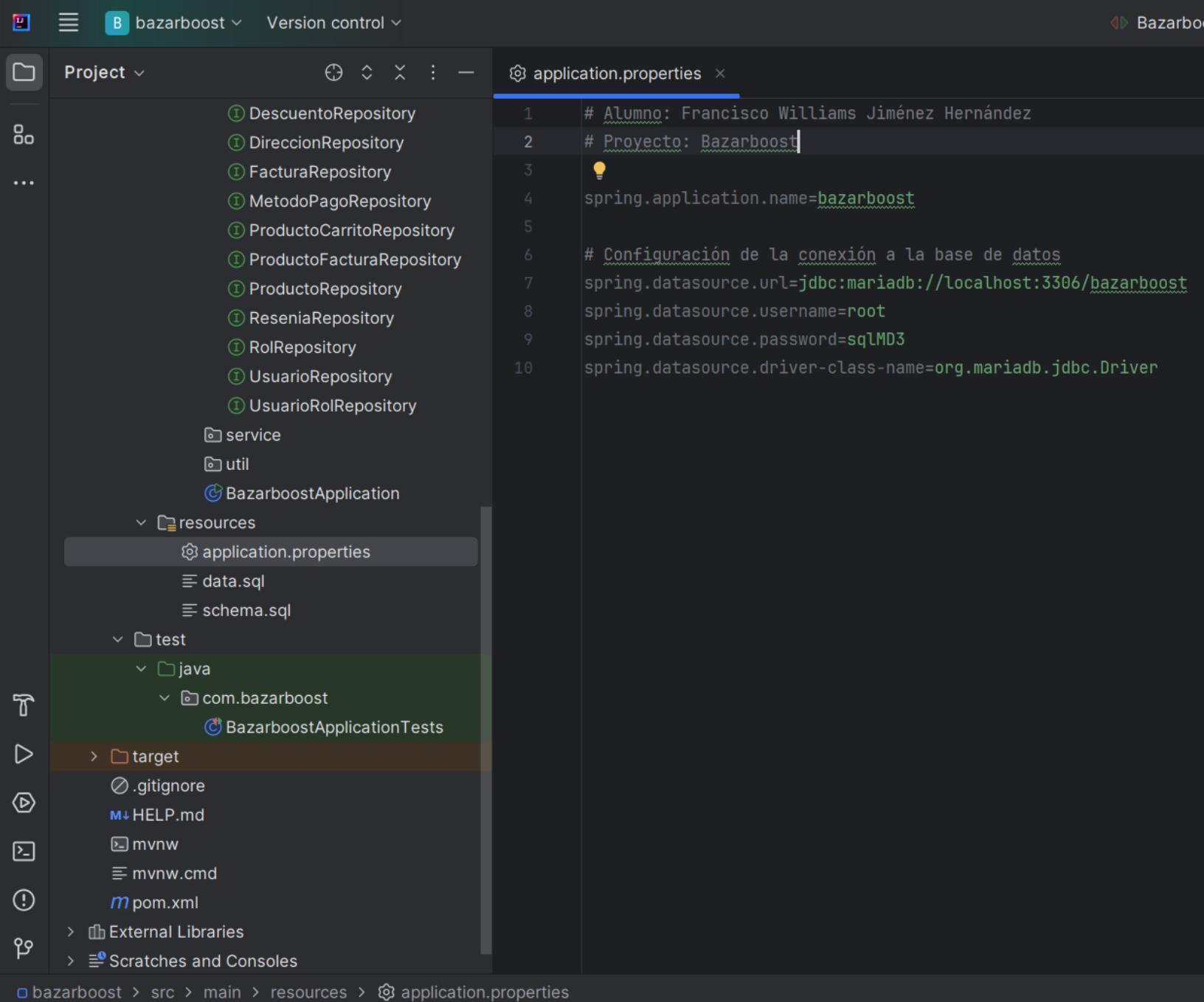


Ejercicio 2

1. Configuración inicial

Para llevar a cabo este ejercicio, inicialmente configuré la conexión a la base de datos a través del archivo **application.properties**. En este archivo, especifiqué los parámetros correspondientes a la base de datos, incluyendo el nombre de la base de datos, el usuario, la contraseña y el controlador (driver) necesario para establecer la conexión de manera adecuada.

Archivo: application.properties



The screenshot shows a Java IDE interface with a dark theme. On the left, the project structure is visible under the 'Project' tab, showing packages like 'com.bazarboost' containing classes such as 'BazarboostApplicationTests'. Below the project tree, there are tabs for 'External Libraries' and 'Scratches and Consoles'. The main editor window on the right displays the 'application.properties' file. The code in the file is as follows:

```
# Alumno: Francisco Williams Jiménez Hernández
# Proyecto: Bazarboost
spring.application.name=bazarboost

# Configuración de la conexión a la base de datos
spring.datasource.url=jdbc:mariadb://localhost:3306/bazarboost
spring.datasource.username=root
spring.datasource.password=sqlMD3
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
```

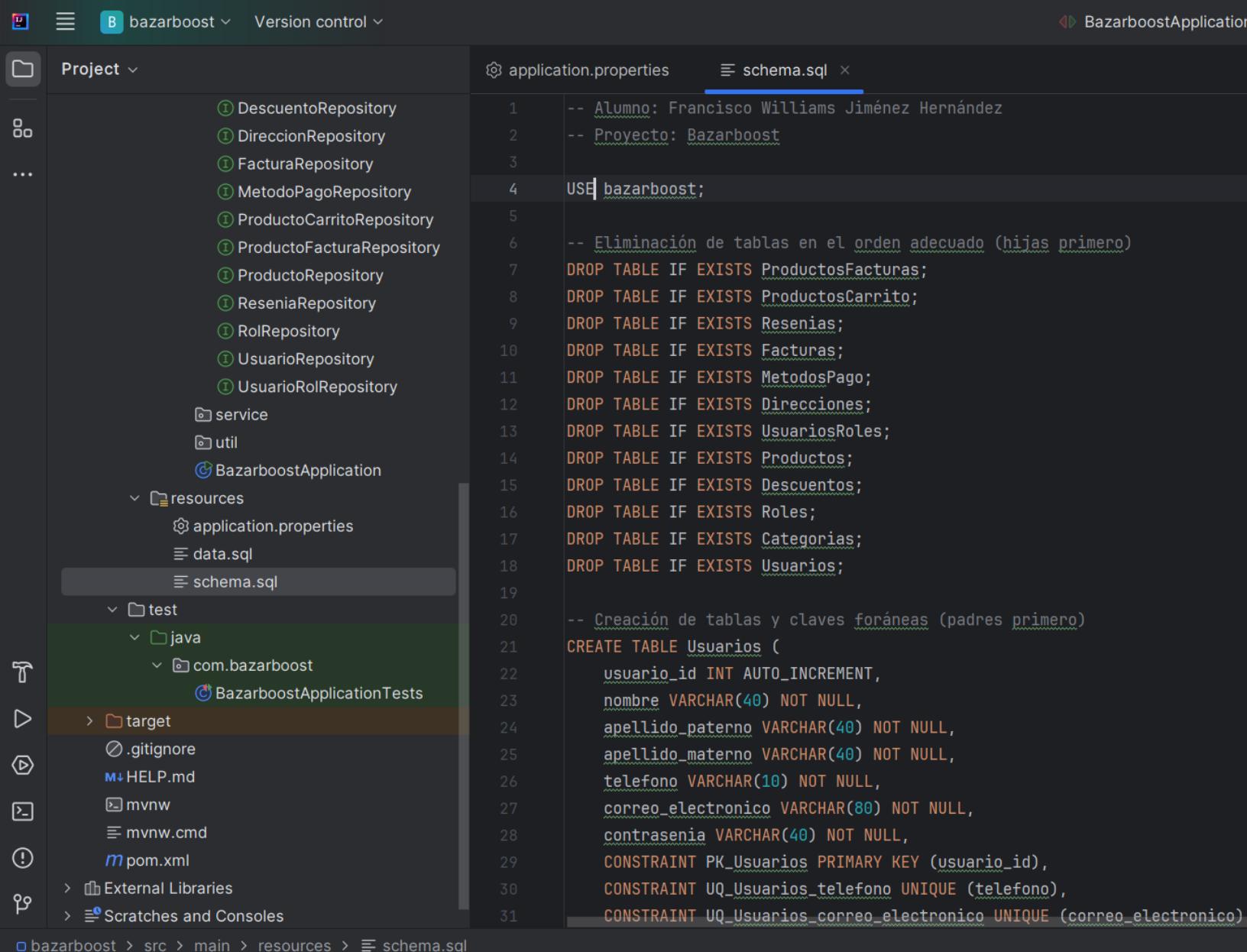
Captura 1. Código del archivo application.properties del proyecto para la configuración de la conexión a la base de datos.

Francisco Williams Jiménez Hernández

Ejercicio 2

Posteriormente, elaboré los scripts **schema.sql** y **data.sql** para la creación del esquema de la base de datos y la inserción de los datos de prueba. Estos archivos fueron ubicados en la carpeta **resources** del proyecto. A continuación, se presenta el contenido de ambos archivos.

Archivo: schema.sql



The screenshot shows a Java IDE interface with the following details:

- Project View:** Shows the project structure under "Project". It includes repositories for Descuento, Direccion, Factura, MetodoPago, ProductoCarrito, ProductoFactura, Producto, Resenia, Rol, Usuario, and UsuarioRol. It also contains "service", "util", and the main application class "BazarboostApplication".
- Resources View:** Shows files under "resources": "application.properties", "data.sql", and "schema.sql".
- Editor View:** Shows the content of "schema.sql".
- Bottom Status Bar:** Shows the path: "bazarboost > src > main > resources > schema.sql".

The content of the "schema.sql" file is as follows:

```
-- Alumno: Francisco Williams Jiménez Hernández
-- Proyecto: Bazarboost

USE bazarboost;

-- Eliminación de tablas en el orden adecuado (hijas primero)
DROP TABLE IF EXISTS ProductosFacturas;
DROP TABLE IF EXISTS ProductosCarrito;
DROP TABLE IF EXISTS Resenias;
DROP TABLE IF EXISTS Facturas;
DROP TABLE IF EXISTS MetodosPago;
DROP TABLE IF EXISTS Direcciones;
DROP TABLE IF EXISTS UsuariosRoles;
DROP TABLE IF EXISTS Productos;
DROP TABLE IF EXISTS Descuentos;
DROP TABLE IF EXISTS Roles;
DROP TABLE IF EXISTS Categorias;
DROP TABLE IF EXISTS Usuarios;

-- Creación de tablas y claves foráneas (padres primero)
CREATE TABLE Usuarios (
    usuario_id INT AUTO_INCREMENT,
    nombre VARCHAR(40) NOT NULL,
    apellido_paterno VARCHAR(40) NOT NULL,
    apellido_materno VARCHAR(40) NOT NULL,
    telefono VARCHAR(10) NOT NULL,
    correo_electronico VARCHAR(80) NOT NULL,
    contrasenia VARCHAR(40) NOT NULL,
    CONSTRAINT PK_Usuarios PRIMARY KEY (usuario_id),
    CONSTRAINT UQ_Usuarios_telefono UNIQUE (telefono),
    CONSTRAINT UQ_Usuarios_correo_electronico UNIQUE (correo_electronico)
```

Captura 2. Código del archivo schema.sql para la creación del esquema.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: data.sql

```
-- Alumno: Francisco Williams Jiménez Hernández
-- Proyecto: Bazarboost

-- Inserts de datos de prueba

-- Inserciones en la tabla Usuarios
INSERT INTO Usuarios (nombre, apellido_paterno, apellido_materno, correo_electronico, contrasenia, telefono) VALUES
('Juan', 'Pérez', 'López', 'juan.perez@example.com', 'password123', '55512345671'),
('María', 'Gómez', 'Martínez', 'maria.gomez@example.com', 'password123', '5557654321'),
('Pedro', 'Sánchez', 'Hernández', 'pedro.sanchez@example.com', 'password123', '5556781234'),
('Ana', 'Ramírez', 'García', 'ana.ramirez@example.com', 'password123', '5559876543'),
('Carlos', 'Fernández', 'Pérez', 'carlos.fernandez@example.com', 'password123', '5554567890');

-- Inserciones en la tabla Categorías
INSERT INTO Categorias (nombre) VALUES
('Electrónica'),
('Ropa'),
('Hogar'),
('Juguetes'),
('Libros');

-- Inserciones en la tabla Roles
INSERT INTO Roles (nombre, descripcion) VALUES
('Administrador', 'Acceso total al sistema'),
('Cliente', 'Puede realizar compras y dejar reseñas'),
('Vendedor', 'Puede gestionar productos y ventas');

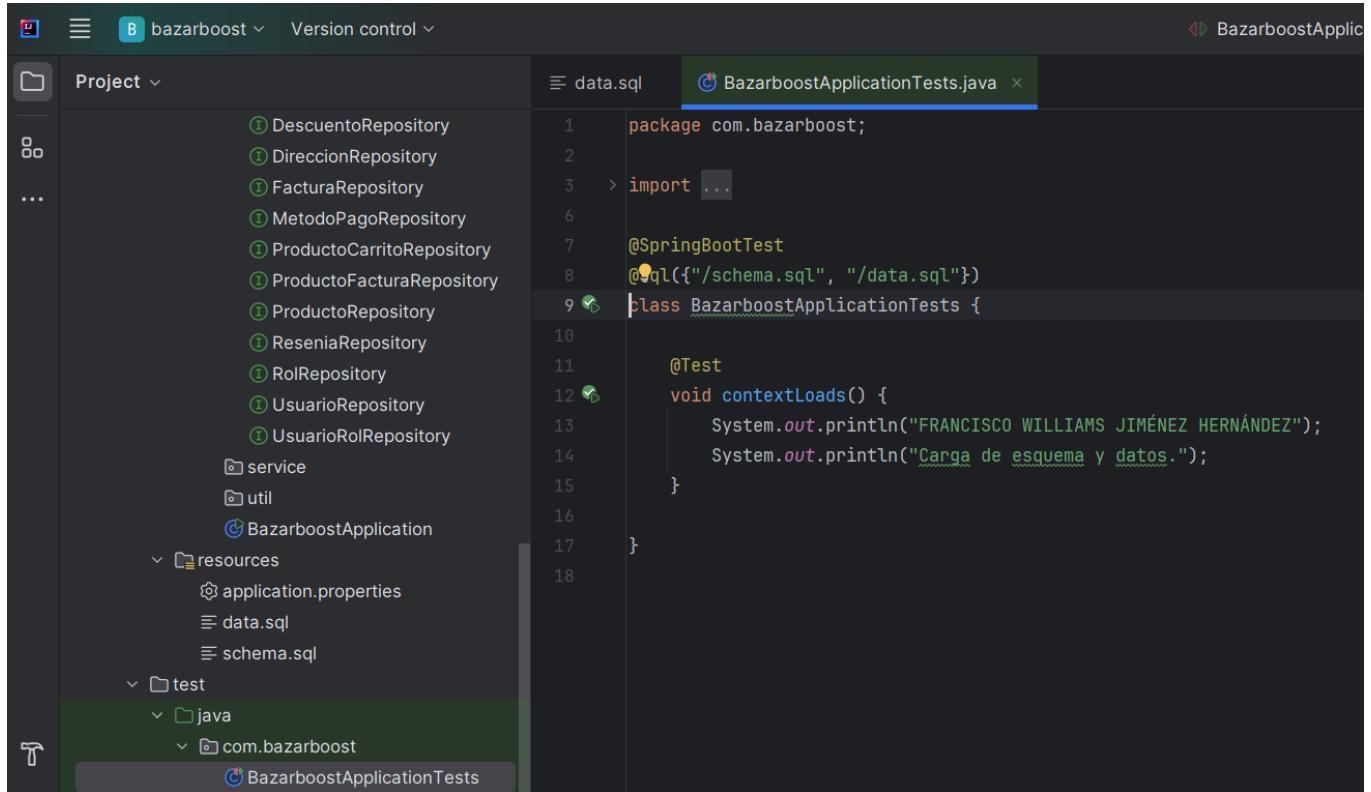
-- Inserciones en la tabla Descuentos
INSERT INTO Descuentos (porcentaje, nombre, usuario_id) VALUES
(10, 'Descuento de Bienvenida', 1),
(15, 'Descuento de Verano', 2),
```

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

Francisco Williams Jiménez Hernández

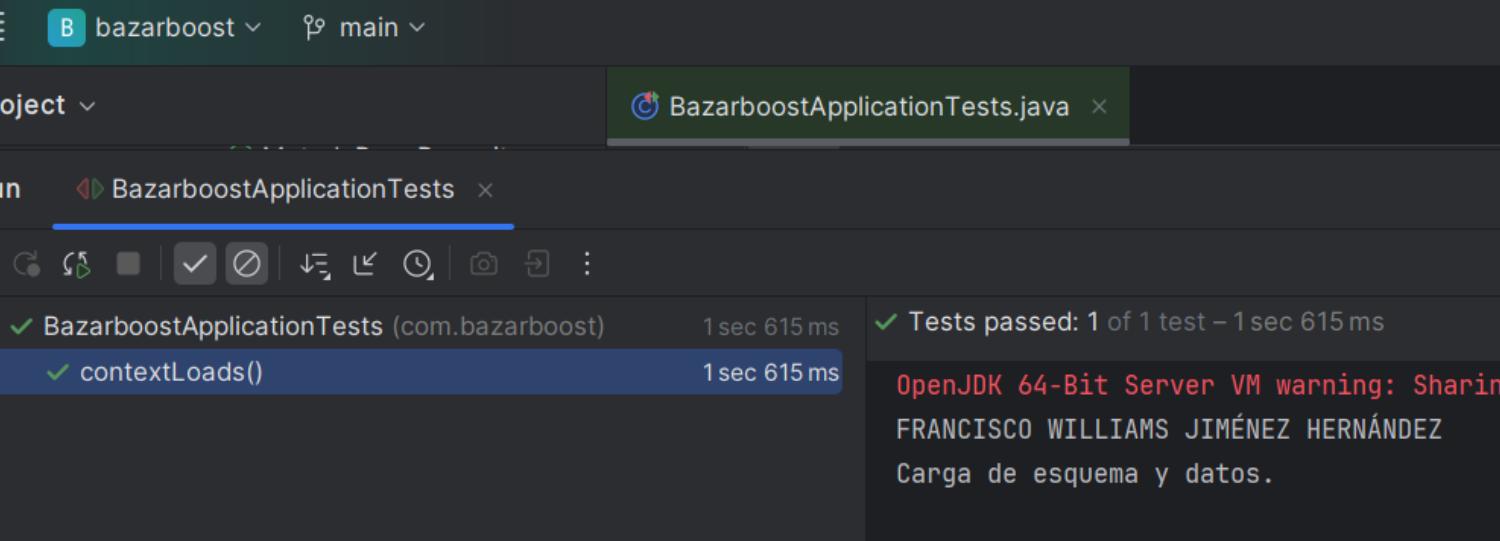
Ejercicio 2

Finalmente, utilicé la anotación `@Sql` en el archivo de pruebas `BazarboostApplicationTests.java` para indicar la ejecución de los scripts `schema.sql` y `data.sql` cada vez que se ejecute dicha clase de pruebas. Esto permite inicializar y/o restablecer el estado de la base de datos según sea necesario para garantizar un entorno consistente en cada ejecución.



The screenshot shows the IntelliJ IDEA IDE interface. The top navigation bar includes icons for file, edit, and version control, followed by the project name "bazarboost" and a "Version control" dropdown. On the right, there's a status bar with "BazarboostApplic". The main area has a "Project" tool window on the left containing a tree view of Java packages, repositories, and configuration files like application.properties, data.sql, and schema.sql. The central editor window is titled "BazarboostApplicationTests.java" and contains the following code:

```
package com.bazarboost;
import ...
@SpringBootTest
@Sql({"/schema.sql", "/data.sql"})
class BazarboostApplicationTests {
    @Test
    void contextLoads() {
        System.out.println("FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ");
        System.out.println("Carga de esquema y datos.");
    }
}
```



The screenshot shows the IntelliJ IDEA interface during a test run. The top navigation bar includes icons for file, edit, and run, followed by the project name "bazarboost" and a "main" dropdown. The "Run" tab is selected, showing a list of tests: "BazarboostApplicationTests (com.bazarboost)" and "contextLoads()". The "contextLoads()" test is highlighted in blue. The bottom right pane displays the test results: "Tests passed: 1 of 1 test – 1 sec 615 ms". Below this, a warning message from OpenJDK is shown: "OpenJDK 64-Bit Server VM warning: Sharing FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ Carga de esquema y datos.". The status bar at the bottom indicates "1 sec 615 ms".

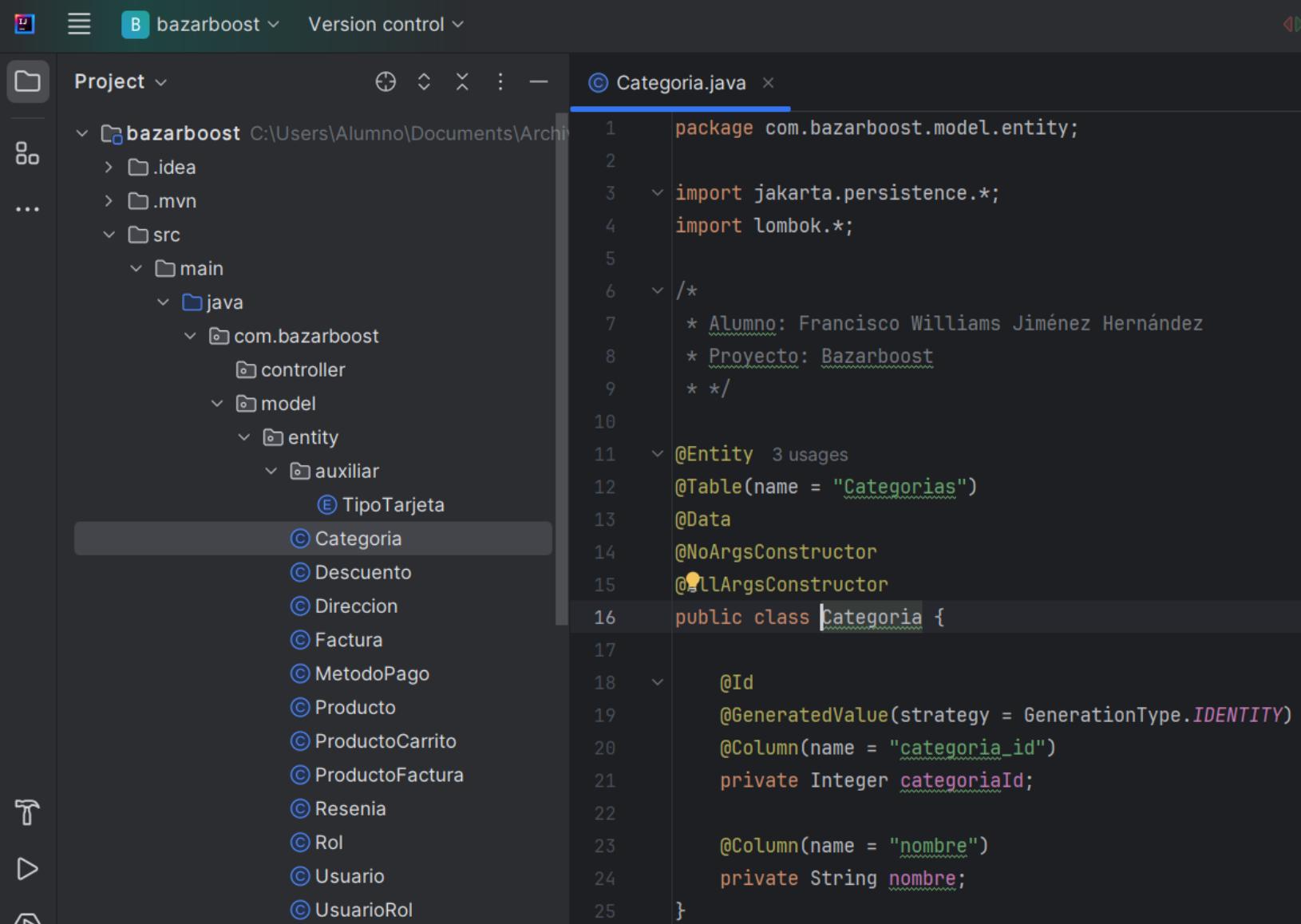
Captura 4. Código del archivo `BazarboostApplicationTests.java` y su salida de ejecución para la creación del esquema y la carga de datos de prueba.

Ejercicio 2

2. Creación de entidades

Una vez completadas las configuraciones iniciales para la conexión a la base de datos, así como la carga del esquema y los datos de prueba, el siguiente paso fue la creación de las clases correspondientes a las entidades. Para ello, se emplearon las anotaciones de **Spring Data JPA**, las cuales permitieron el mapeo de las clases a las tablas relacionales de la base de datos. Además, se utilizaron las anotaciones de **Lombok** con el fin de reducir el código repetitivo, como los métodos **constructores, getters, setters**, entre otros.

Archivo: Categoría.java



The screenshot shows a Java code editor in an IDE. On the left is a project tree with the following structure:

- bazarboost (C:\Users\Alumno\Documents\Archivos)
- .idea
- .mvn
- src
 - main
 - java
 - com.bazarboost
 - controller
 - model
 - entity
 - TipoTarjeta
 - Categoría
 - Descuento
 - Direccion
 - Factura
 - MetodoPago
 - Producto
 - ProductoCarrito
 - ProductoFactura
 - Resenia
 - Rol
 - Usuario
 - UsuarioRol

Captura 5. Clase de entidad para *Categoría*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: Descuento.java

The screenshot shows a Java development environment with the following details:

- Project:** bazarboost (C:\Users\Alumno\Documents\Archivos)
- File:** Descuento.java
- Code Content:** The code defines a class `Descuento` with annotations for Entity, Table, Data, NoArgsConstructor, and AllArgsConstructor. It includes fields for `descuento_id`, `porcentaje`, and `nombre`, and a many-to-one relationship with `Usuario`. The code is annotated with copyright information for Francisco Williams Jiménez Hernández and the project Bazarboost.

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 3 usages
@Table(name = "Descuentos")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Descuento {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "descuento_id")
    private Integer descuentoId;

    @Column(name = "porcentaje")
    private Integer porcentaje;

    @Column(name = "nombre")
    private String nombre;

    @ManyToOne
    @JoinColumn(name = "usuario_id")
    private Usuario usuario;
}
```

- File Explorer:** Shows the project structure with packages like `bazarboost`, `.idea`, `.mvn`, and `src`. Under `src/main/java/com.bazarboost/model/entity`, files include `TipoTarjeta.java`, `Categoría.java`, `Descuento.java`, `Dirección.java`, `Factura.java`, `MétodoPago.java`, `Producto.java`, `ProductoCarrito.java`, `ProductoFactura.java`, `Reseña.java`, `Rol.java`, `Usuario.java`, `UsuarioRol.java`, `exception`, `repository`, and `FacturaRepository.java`.
- Status Bar:** Shows the path `bazarboost > src > main > java > com > bazarboost > model > entity > Descuento.java`.

Captura 6. Clase de entidad para *Descuento*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: Direccion.java

The screenshot shows a Java project structure on the left and the content of the `Direccion.java` file on the right. The project structure includes a `bazarboost` module with `.idea`, `.mvn`, and `src` directories. The `src/main/java/com/bazarboost/model/entity` package contains several classes like `TipoTarjeta`, `Categoria`, `Descuento`, `Direccion`, etc. The `Direccion.java` file is selected in the editor.

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 3 usages
@Table(name = "Direcciones")
@data
@NoArgsConstructor
@AllArgsConstructor
public class Direccion {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "direccion_id")
    private Integer direccionId;

    @Column(name = "estado")
    private String estado;

    @Column(name = "ciudad")
    private String ciudad;

    @Column(name = "colonia")
    private String colonia;

    @Column(name = "calle")
    private String calle;

    @Column(name = "numero_domicilio")
    private Integer numeroDomicilio;
}
```

The status bar at the bottom shows the full path: `bazarboost > src > main > java > com > bazarboost > model > entity > Direccion.java`.

Captura 7. Clase de entidad para *Direccion*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: Factura.java

The screenshot shows a Java IDE interface with the following details:

- Project View:** Shows the project structure under "bazarboost". The "entity" package is expanded, and "Factura" is selected.
- Code Editor:** Displays the content of the Factura.java file.
- Code Content:**

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 3 usages
@Table(name = "Facturas")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Factura {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "factura_id")
    private Integer facturaId;

    @Column(name = "fecha")
    private LocalDateTime fecha;

    @Column(name = "subtotal")
    private Double subtotal;

    @Column(name = "total")
    private Double total;

    @Column(name = "porcentaje_impuestos")
    private Integer porcentajeImpuestos;

    @ManyToOne
    @JoinColumn(name = "usuario_id")
}
```
- Status Bar:** Shows the current path: bazarboost > src > main > java > com > bazarboost > model > entity > Factura

Captura 8. Clase de entidad para *Factura*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: MetodoPago.java

The screenshot shows a Java IDE interface with the following details:

- Project Explorer:** Shows the project structure under 'bazarboost'. The 'model' package is selected, containing classes like Categoria, Descuento, Direccion, Factura, MetodoPago, Producto, ProductoCarrito, ProductoFactura, Resenia, Rol, Usuario, UsuarioRol, exception, and repository.
- Editor:** Displays the code for 'MetodoPago.java'. The code defines a class 'MetodoPago' with annotations for Entity, Table, Data, NoArgsConstructor, and AllArgsConstructor. It includes fields for metodo_pago_id (@Id, @GeneratedValue(strategy = GenerationType.IDENTITY), @Column(name = "metodo_pago_id")), nombre_titular (private String nombreTitular; @Column(name = "nombre_titular")), numero_tarjeta (private String numeroTarjeta; @Column(name = "numero_tarjeta")), fecha_expiracion (private LocalDate fechaExpiracion; // Cambio a LocalDate), tipo_tarjeta (private TipoTarjeta tipoTarjeta; @Column(name = "tipo_tarjeta") @Enumerated(EnumType.STRING)), and monto (private Integer monto; @Column(name = "monto")).

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */

@Entity 5 usages  ↳ Williams
@Table(name = "MetodosPago")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class MetodoPago {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "metodo_pago_id")
    private Integer metodoPagoId;

    @Column(name = "nombre_titular")
    private String nombreTitular;

    @Column(name = "numero_tarjeta")
    private String numeroTarjeta;

    @Column(name = "fecha_expiracion")
    private LocalDate fechaExpiracion; // Cambio a LocalDate

    @Column(name = "tipo_tarjeta")
    @Enumerated(EnumType.STRING)
    private TipoTarjeta tipoTarjeta;

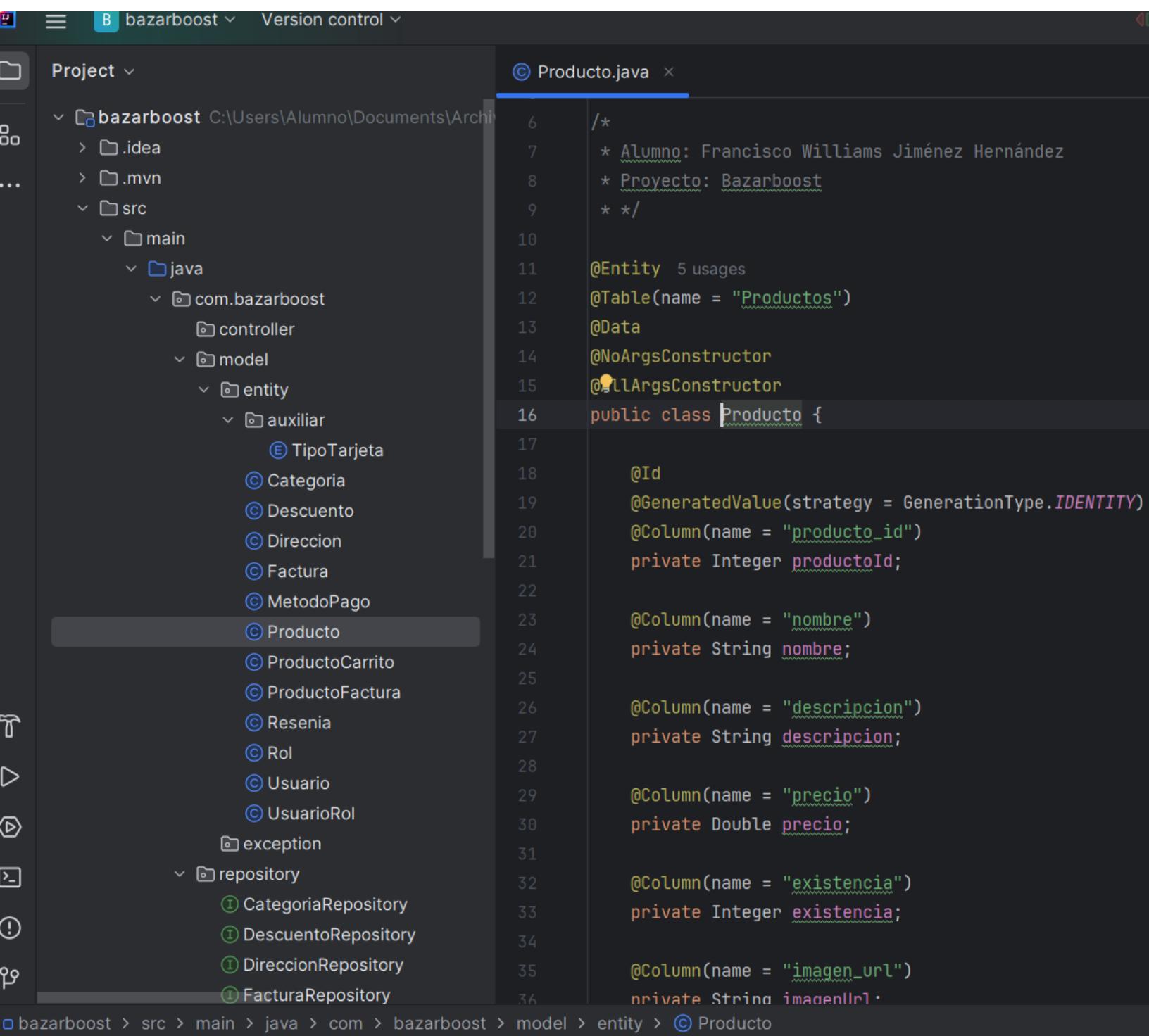
    @Column(name = "monto")
    private Integer monto;
}
```

Captura 9. Clase de entidad para *MetodoPago*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: Producto.java



The screenshot shows an IDE interface with a project tree on the left and a code editor on the right. The project tree shows a structure under 'bazarboost': .idea, .mvn, src (main, java, com.bazarboost, controller, model, entity, auxiliar, exception, repository, CategoriaRepository, DescuentoRepository, DireccionRepository, FacturaRepository). The 'entity' folder contains several classes: TipoTarjeta, Categoria, Descuento, Direccion, Factura, MetodoPago, Producto (which is selected in the tree), ProductoCarrito, ProductoFactura, Resenia, Rol, Usuario, UsuarioRol, and exception. The code editor displays the 'Producto.java' file with the following content:

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 5 usages
@Table(name = "Productos")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Producto {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "producto_id")
    private Integer productoId;

    @Column(name = "nombre")
    private String nombre;

    @Column(name = "descripcion")
    private String descripcion;

    @Column(name = "precio")
    private Double precio;

    @Column(name = "existencia")
    private Integer existencia;

    @Column(name = "imagen_url")
    private String imagenUrl;
}
```

The status bar at the bottom shows the full path: bazarboost > src > main > java > com > bazarboost > model > entity > Producto.java

Captura 10. Clase de entidad para *Producto*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: ProductoCarrito.java

The screenshot shows a Java code editor with the file `ProductoCarrito.java` open. The code defines a class `ProductoCarrito` with various annotations and fields. The code is as follows:

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 2 usages
@Table(name = "ProductosCarrito")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class ProductoCarrito {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "producto_carrito_id")
    private Integer productoCarritoId;

    @Column(name = "cantidad")
    private Integer cantidad;

    @Column(name = "total")
    private Double total;

    @ManyToOne
    @JoinColumn(name = "usuario_id")
    private Usuario usuario;

    @ManyToOne
    @JoinColumn(name = "producto_id")
    private Producto producto;
}
```

The code editor interface includes a project tree on the left showing the directory structure of the `bazarboost` project, and a toolbar at the top with various icons.

Captura 11. Clase de entidad para `ProductoCarrito`.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: ProductoFactura.java

The screenshot shows a Java IDE interface with the following details:

- Project:** bazarboost (C:\Users\Alumno\Documents\Archivos)
- File:** ProductoFactura.java
- Code Content:**

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 2 usages
@Table(name = "ProductosFacturas")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class ProductoFactura {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "producto_factura_id")
    private Integer productoFacturaId;

    @Column(name = "cantidad")
    private Integer cantidad;

    @Column(name = "total")
    private Double total;

    @ManyToOne
    @JoinColumn(name = "factura_id")
    private Factura factura;

    @ManyToOne
    @JoinColumn(name = "producto_id")
    private Producto producto;
}
```
- File Tree:** Shows the project structure with packages like .idea, .mvn, and src containing main/java/com/bazarboost/controller, main/java/com/bazarboost/model/entity, and main/java/com/bazarboost/model/auxiliar.
- Status Bar:** Shows the path: bazarboost > src > main > java > com > bazarboost > model > entity > ProductoFactura.java

Captura 12. Clase de entidad para *ProductoFactura*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: Resenia.java

The screenshot shows a Java code editor with the file `Resenia.java` open. The code defines a class `Resenia` with various annotations and fields. The code editor interface includes a left sidebar for navigating through the project structure, showing the package tree under `bazarboost`. The main window displays the code with line numbers on the left.

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 2 usages
@Table(name = "Resenias")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Resenia {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "resenia_id")
    private Integer reseniaId;

    @Column(name = "comentario")
    private String comentario;

    @Column(name = "calificacion")
    private Integer calificacion;

    @Column(name = "fecha")
    private LocalDateTime fecha;

    @ManyToOne
    @JoinColumn(name = "usuario_id")
    private Usuario usuario;
}
```

Captura 13. Clase de entidad para *Resenia*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: Rol.java

The screenshot shows a Java development environment with the following details:

- Project View:** Shows the project structure under "bazarboost". The "model" package contains several entity classes: Resenia, Rol, Usuario, ProductoFactura, ProductoCarrito, Producto, Direccion, Categoria, and Tipotarjeta.
- Rol.java File Content:** The code defines the entity class `Rol`. It includes annotations for the primary key (`@Id`), table name (`@Table(name = "Roles")`), and columns (`@Column(name = "rol_id")` and `@Column(name = "nombre")`). It also includes annotations for constructor generation (`@NoArgsConstructor` and `@AllArgsConstructor`).

```
5  /*
6   * Alumno: Francisco Williams Jiménez Hernández
7   * Proyecto: Bazarboost
8   */
9
10
11 @Entity 3 usages
12 @Table(name = "Roles")
13 @Data
14 @NoArgsConstructor
15 @AllArgsConstructor
16 public class Rol {
17
18     @Id
19     @GeneratedValue(strategy = GenerationType.IDENTITY)
20     @Column(name = "rol_id")
21     private Integer rolId;
22
23     @Column(name = "nombre")
24     private String nombre;
25
26     @Column(name = "descripcion")
27     private String descripcion;
28 }
```

Captura 14. Clase de entidad para *Rol*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: Usuario.java

The screenshot shows a Java IDE interface with the following details:

- Project:** bazarboost (C:\Users\Alumno\Documents\Archivos)
- File:** Usuario.java
- Content:** The code defines a Java entity class named `Usuario`. It includes annotations for the primary key (`@Id`), table name (`@Table(name = "Usuarios")`), and various columns (`@Column(name = "nombre")`, `@Column(name = "apellido_paterno")`, etc.). The class also contains private fields corresponding to these columns.
- Code Snippet:**

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@Entity 10 usages
@Table(name = "Usuarios")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Usuario {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "usuario_id")  
    private Integer usuarioId;  
  
    @Column(name = "nombre")  
    private String nombre;  
  
    @Column(name = "apellido_paterno")  
    private String apellidoPaterno;  
  
    @Column(name = "apellido_materno")  
    private String apellidoMaterno;  
  
    @Column(name = "telefono")  
    private String telefono;  
  
    @Column(name = "correo_electronico")  
    private String correoElectronico;
```
- IDE UI:** The left sidebar shows the project structure with the `Usuario` class highlighted. The bottom navigation bar shows the full path: bazarboost > src > main > java > com > bazarboost > model > entity > Usuario.java.

Captura 15. Clase de entidad para *Usuario*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: UsuarioRol.java

The screenshot shows a Java development environment with the following details:

- Project:** bazarboost (C:\Users\Alumno\Documents\Archivos)
- File:** UsuarioRol.java
- Content:** The code defines the `UsuarioRol` entity class. It includes annotations for the entity name (`@Entity`), table name (`@Table(name = "UsuariosRoles")`), and primary key (`@Id @GeneratedValue(strategy = GenerationType.IDENTITY)`). It also includes fields for `usuarioRoleId` (primary key), `fechaAsignacion` (date of assignment), and `usuario` (user reference). Additionally, it has a many-to-one relationship with the `Rol` entity.
- Annotations:** The code uses various annotations from the `javax.persistence` package, such as `@Entity`, `@Table`, `@Id`, `@GeneratedValue`, `@Column`, `@ManyToMany`, and `@JoinColumn`.

Captura 16. Clase de entidad para `UsuarioRol`.

3. Creación de repositorios

Una vez creadas las entidades, elaboré los repositorios correspondientes, cada repositorio contiene los métodos para consultas requeridas de acuerdo a las reglas de negocio de mi proyecto final; dichas queries incluyen consultas derivadas, nombradas, de CrudRepository y de PagingAndSorting.

Archivo: CategoriaRepository.java

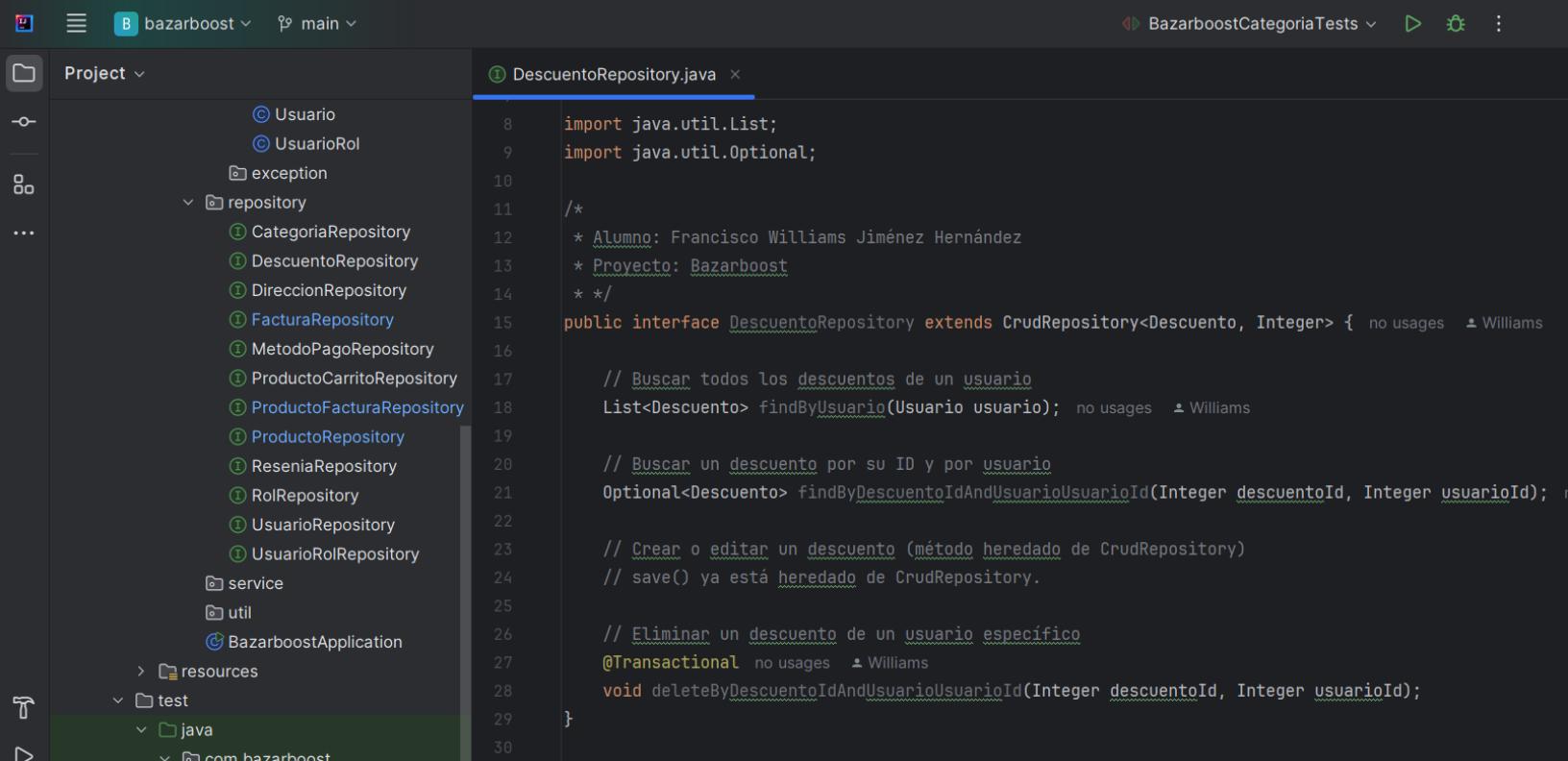
```
6 import java.util.List;
7 import java.util.Optional;
8
9
10 /*
11  * Alumno: Francisco Williams Jiménez Hernández
12  * Proyecto: Bazarboost
13  */
14 public interface CategoriaRepository extends CrudRepository<Categoria, Integer> {
15
16     // Encontrar todas las categorías
17     List<Categoria> findAll(); // Williams
18
19     // Buscar una categoría por su ID
20     Optional<Categoria> findById(Integer categoriaId); // 2 usages // Williams
21
22     // Eliminar una categoría por su ID
23     @Transactional 1 usage // Williams
24     void deleteById(Integer categoriaId);
25
26     // Crear o editar una categoría (método heredado de CrudRepository)
27     // save() ya está heredado de CrudRepository.
28 }
```

Captura 17. Clase de interfaz de repositorio para *Categoría*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: DescuentoRepository.java



The screenshot shows a Java IDE interface with the following details:

- Project:** bazarboost
- File:** DescuentoRepository.java
- Content:** The code defines an interface for a repository named DescuentoRepository. It extends the CrudRepository interface for the Descuento entity. The interface includes methods for finding all discounts for a user, finding a discount by its ID and the user's ID, and deleting a specific discount for a user. The code is annotated with Javadoc comments indicating the author is Francisco Williams Jiménez Hernández and the project is Bazarboost.

```
import java.util.List;
import java.util.Optional;

/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */

public interface DescuentoRepository extends CrudRepository<Descuento, Integer> { no usages  Williams

    // Buscar todos los descuentos de un usuario
    List<Descuento> findByUsuario(Usuario usuario); no usages  Williams

    // Buscar un descuento por su ID y por usuario
    Optional<Descuento> findByIdAndUsuario(Integer descuentoId, Integer usuarioId);

    // Crear o editar un descuento (método heredado de CrudRepository)
    // save() ya está heredado de CrudRepository.

    // Eliminar un descuento de un usuario específico
    @Transactional no usages  Williams
    void deleteByIdAndUsuario(Integer descuentoId, Integer usuarioId);
}
```

Captura 18. Clase de interfaz de repositorio para *Descuento*.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: DireccionRepository.java

The screenshot shows a Java IDE interface with the following details:

- Project:** bazarboost
- File:** DireccionRepository.java
- Content:** The code defines an interface for a repository named DireccionRepository that extends CrudRepository<Direccion, Integer>. The interface includes methods for finding all addresses of a user by ID, finding an address by ID and user ID, deleting an address by ID and user ID, and saving a new or edited address.
- Annotations:** The code includes several Javadoc-style annotations: `/* * Alumno: Francisco Williams Jiménez Hernández * Proyecto: Bazarboost */`, `@Transactional`, and `CrudRepository`.
- IDE UI:** The left sidebar shows a project tree with packages like entity and repository, and various repository classes listed under repository. The DireccionRepository class is highlighted.

Captura 19. Clase de interfaz de repositorio para *Dirección*.

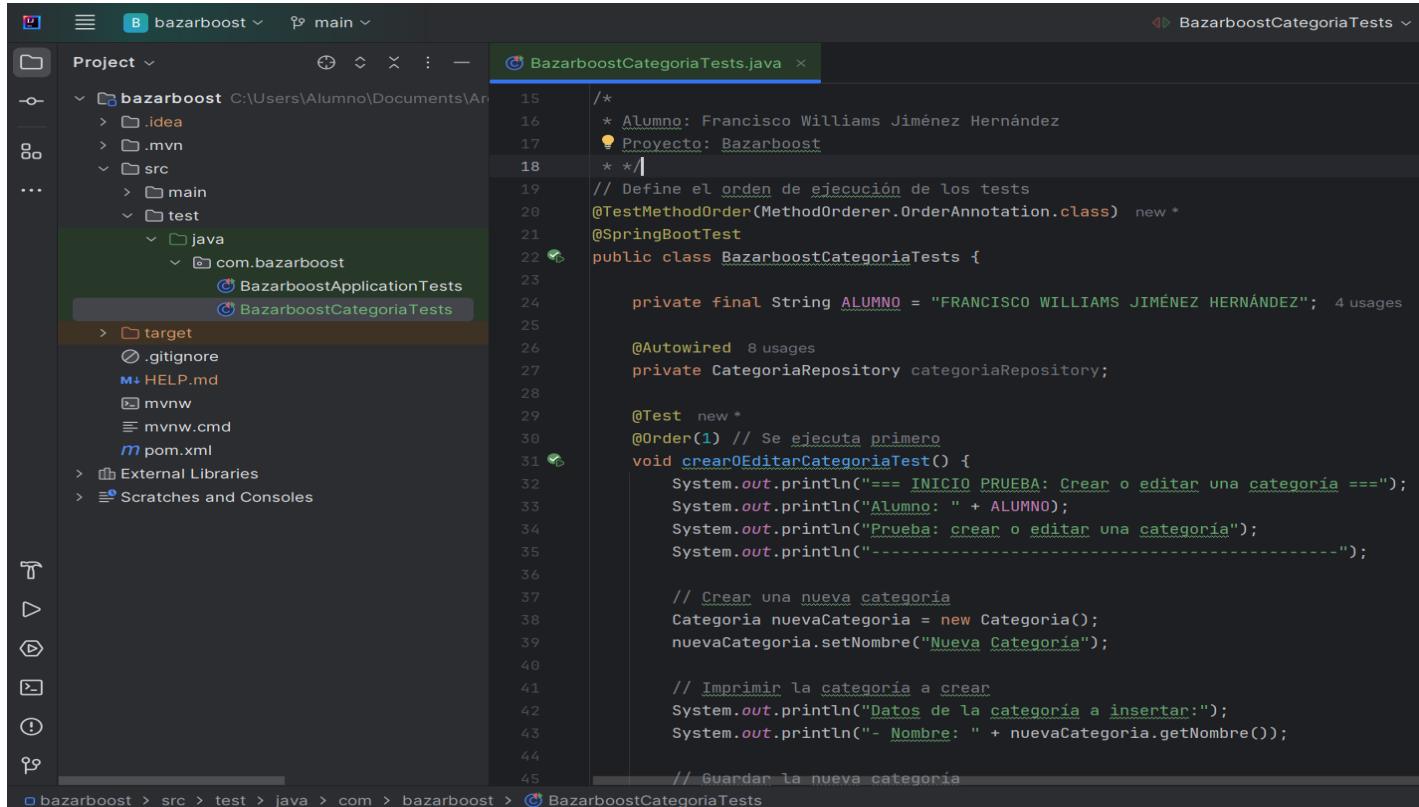
Francisco Williams Jiménez Hernández

Ejercicio 2

4. Clases de pruebas y salidas de ejecución

Después de crear los repositorios, desarrollé las clases de prueba correspondientes, una para cada entidad, e implementé los métodos necesarios para verificar el correcto funcionamiento de los métodos definidos en cada repositorio.

Archivo: BazarboostCategoriaTests.java y su salida de ejecución



The screenshot shows the Java code for `BazarboostCategoriaTests.java`. The code is annotated with comments and imports related to testing and Spring Boot. It includes a test method `crearEditarCategoriaTest()` that prints category creation and editing details to the console.

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
// Define el orden de ejecución de los tests
@TestMethodOrder(MethodOrderer.OrderAnnotation.class) new *
@SpringBootTest
public class BazarboostCategoriaTests {

    private final String ALUMNO = "FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ"; 4 usages

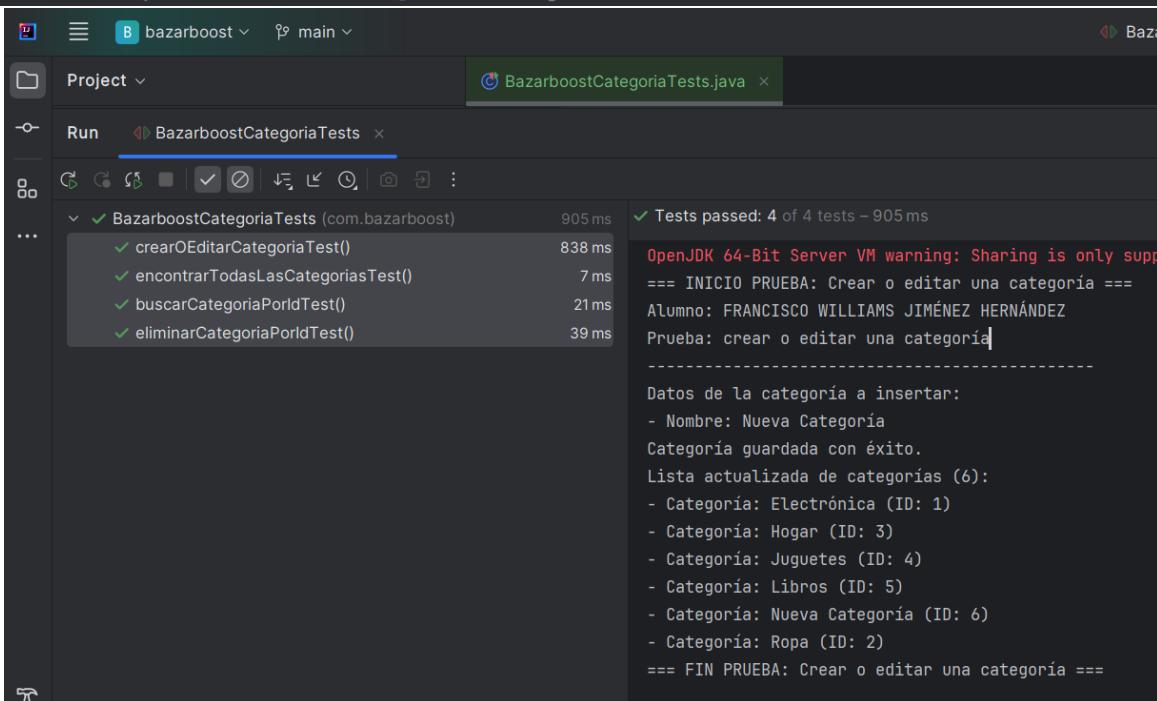
    @Autowired 8 usages
    private CategoriaRepository categoriaRepository;

    @Test new *
    @Order(1) // Se ejecuta primero
    void crearEditarCategoriaTest() {
        System.out.println("== INICIO PRUEBA: Crear o editar una categoría ==");
        System.out.println("Alumno: " + ALUMNO);
        System.out.println("Prueba: crear o editar una categoría");
        System.out.println("-----");

        // Crear una nueva categoría
        Categoria nuevaCategoria = new Categoria();
        nuevaCategoria.setNombre("Nueva Categoría");

        // Imprimir la categoría a crear
        System.out.println("Datos de la categoría a insertar:");
        System.out.println("- Nombre: " + nuevaCategoria.getNombre());

        // Guardar la nueva categoría
    }
}
```



The screenshot shows the execution results for the `BazarboostCategoriaTests` run. It lists four test methods: `crearEditarCategoriaTest()`, `encontrarTodasLasCategoriasTest()`, `buscarCategoriaPorIdTest()`, and `eliminarCategoriaPorIdTest()`. The results show all tests passed in 905 ms. The output window displays the printed test logs, including category creation and listing details.

Test Method	Time
crearEditarCategoriaTest()	838 ms
encontrarTodasLasCategoriasTest()	7 ms
buscarCategoriaPorIdTest()	21 ms
eliminarCategoriaPorIdTest()	39 ms

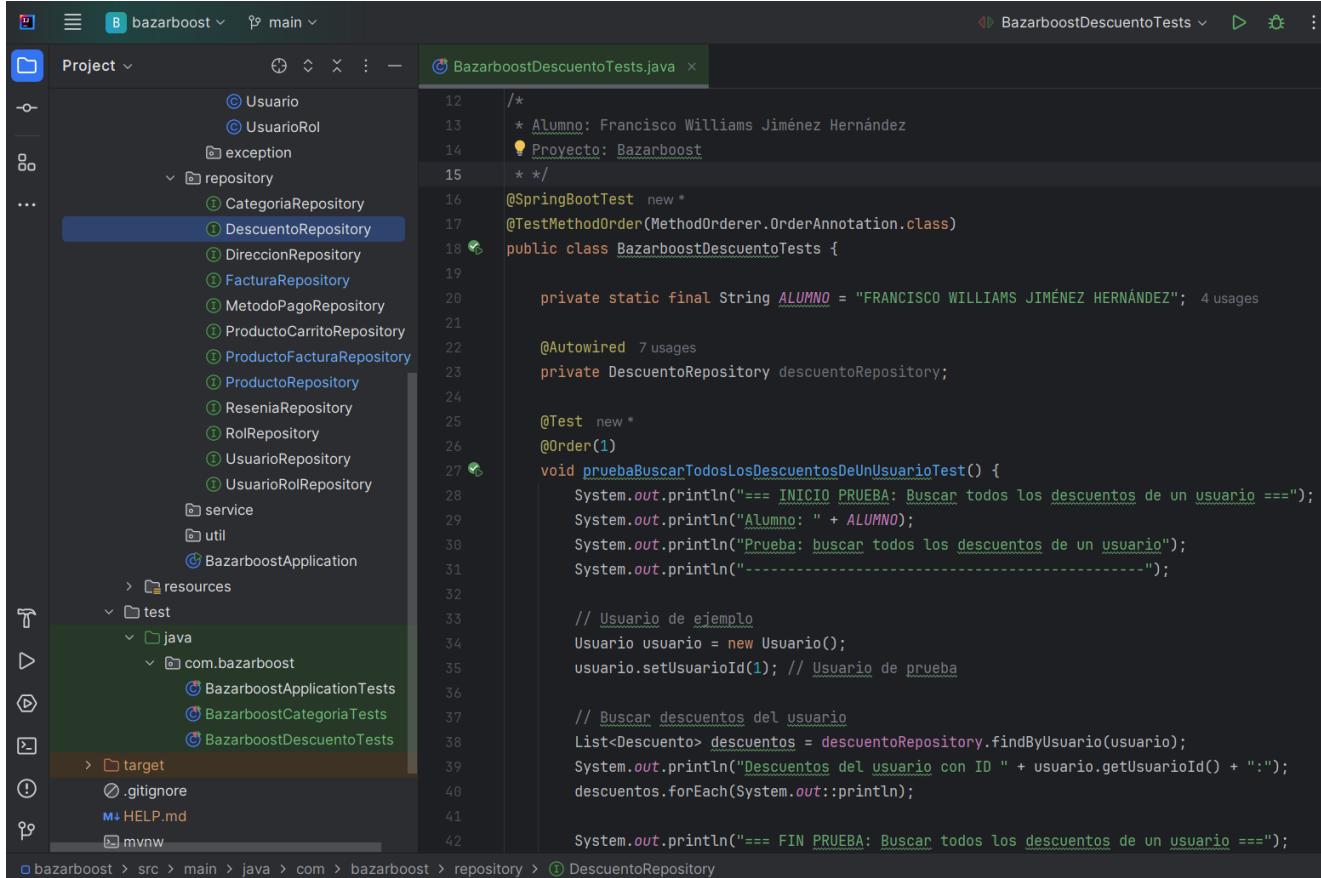
```
Tests passed: 4 of 4 tests – 905 ms
OpenJDK 64-Bit Server VM warning: Sharing is only supp
== INICIO PRUEBA: Crear o editar una categoría ==
Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba: crear o editar una categoría
-----
Datos de la categoría a insertar:
- Nombre: Nueva Categoría
Categoría guardada con éxito.
Lista actualizada de categorías (6):
- Categoría: Electrónica (ID: 1)
- Categoría: Hogar (ID: 3)
- Categoría: Juguetes (ID: 4)
- Categoría: Libros (ID: 5)
- Categoría: Nueva Categoría (ID: 6)
- Categoría: Ropa (ID: 2)
== FIN PRUEBA: Crear o editar una categoría ==
```

Captura X. Clase de pruebas para *Categoría* y su salida de ejecución.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: BazarboostDescuentoTests.java y su salida de ejecución



The screenshot shows the IntelliJ IDEA interface with the project 'bazarboost' open. The left sidebar displays the project structure, including packages like 'com.bazarboost' containing 'BazarboostApplicationTests', 'BazarboostCategoriaTests', and 'BazarboostDescuentoTests'. The right pane shows the code editor for 'BazarboostDescuentoTests.java'. The code is a Spring Boot test class for the 'DescuentoRepository'. It includes a static final string 'ALUMNO' set to 'FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ'. It contains a test method 'pruebaBuscarTodosLosDescuentosDeUnUsuarioTest()' that prints out the results of a query to the database.

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@SpringBootTest
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class BazarboostDescuentoTests {

    private static final String ALUMNO = "FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ";

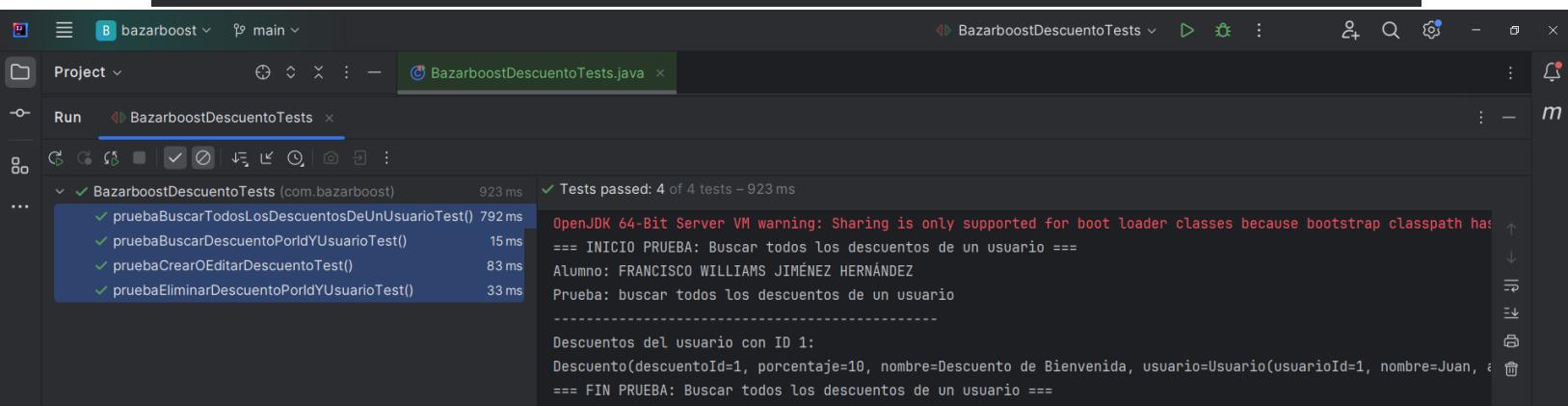
    @Autowired
    private DescuentoRepository descuentoRepository;

    @Test
    @Order(1)
    void pruebaBuscarTodosLosDescuentosDeUnUsuarioTest() {
        System.out.println("== INICIO PRUEBA: Buscar todos los descuentos de un usuario ==");
        System.out.println("Alumno: " + ALUMNO);
        System.out.println("Prueba: buscar todos los descuentos de un usuario");
        System.out.println("-----");

        // Usuario de ejemplo
        Usuario usuario = new Usuario();
        usuario.setUsuarioId(1); // Usuario de prueba

        // Buscar descuentos del usuario
        List<Descuento> descuentos = descuentoRepository.findByUsuario(usuario);
        System.out.println("Descuentos del usuario con ID " + usuario.getUsuarioId() + ":");
        descuentos.forEach(System.out::println);

        System.out.println("== FIN PRUEBA: Buscar todos los descuentos de un usuario ==");
    }
}
```



The screenshot shows the IntelliJ IDEA interface with the 'Run' tab selected. It displays the execution results for the 'BazarboostDescuentoTests' run. The results show 4 tests passed in 923 ms. The output window shows the printed statements from the test code, including the header, the student's name, the search query, the database results, and the footer.

Tests passed: 4 of 4 tests - 923 ms

OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been explicitly defined.

== INICIO PRUEBA: Buscar todos los descuentos de un usuario ==

Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ

Prueba: buscar todos los descuentos de un usuario

Descuentos del usuario con ID 1:

Descuento{descuentoId=1, porcentaje=10, nombre=Descuento de Bienvenida, usuario=Usuario{usuarioId=1, nombre=Juan, apellido=López}}

== FIN PRUEBA: Buscar todos los descuentos de un usuario ==

Captura X. Clase de pruebas para *Descuento* y su salida de ejecución.

Francisco Williams Jiménez Hernández

Ejercicio 2

Archivo: BazarboostDireccionTests.java y su salida de ejecución

The screenshot shows the IntelliJ IDEA interface. The left sidebar displays the project structure for 'bazarboost' with packages like repository, service, util, BazarboostApplication, resources, and test. Under the test package, 'BazarboostDireccionTests' is selected. The main editor window shows the code for BazarboostDireccionTests.java. The code includes annotations for testing (@SpringBootTest, @TestMethodOrder), a static final string ALUMNO, and a private field for the DireccionRepository. It contains a method pruebaEncontrarTodasLasDireccionesDeUsuarioTest() which prints out messages and lists all addresses for a user with ID 1. The code uses System.out.println for output.

```
/*
 * Alumno: Francisco Williams Jiménez Hernández
 * Proyecto: Bazarboost
 */
@SpringBootTest
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class BazarboostDireccionTests {

    private static final String ALUMNO = "FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ"; 4 usages

    @Autowired 7 usages
    private DireccionRepository direccionRepository;

    @Test new *
    @Order(1)
    void pruebaEncontrarTodasLasDireccionesDeUsuarioTest() {
        System.out.println("== INICIO PRUEBA: Encontrar todas las direcciones de un usuario ==");
        System.out.println("Alumno: " + ALUMNO);
        System.out.println("Prueba: encontrar todas las direcciones de un usuario");
        System.out.println("-----");

        // Usuario de ejemplo
        Integer usuarioId = 1;

        // Buscar todas las direcciones de un usuario
        List<Direccion> direcciones = direccionRepository.findByUsuarioUsuarioId(usuarioId);
        System.out.println("Direcciones del usuario con ID " + usuarioId + ":");
        direcciones.forEach(System.out::println);

        System.out.println("== FIN PRUEBA: Encontrar todas las direcciones de un usuario ==");
        System.out.println();
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the 'Run' tab selected. The 'Run' tool window displays the execution results for 'BazarboostDireccionTests'. It shows 4 tests passed in 1 second and 70 milliseconds. The output console shows the printed messages from the test code, including the initial message, the student information, the test name, the user ID used, the list of addresses for that user, and the final message.

Test	Time
pruebaEncontrarTodasLasDireccionesDeUsuarioTest()	921 ms
pruebaBuscarDireccionPorIdYUsuarioTest()	13 ms
pruebaCrearOEditarDireccionTest()	100 ms
pruebaEliminarDireccionPorIdYUsuarioTest()	36 ms

```
Tests passed: 4 of 4 tests - 1 sec 70 ms

OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has:
== INICIO PRUEBA: Encontrar todas las direcciones de un usuario ==
Alumno: FRANCISCO WILLIAMS JIMÉNEZ HERNÁNDEZ
Prueba: encontrar todas las direcciones de un usuario
-----
Direcciones del usuario con ID 1:
Direccion(direccionId=1, estado=Ciudad de México, ciudad=Benito Juárez, colonia=Del Valle, calle=Avenida Insurgentes)
== FIN PRUEBA: Encontrar todas las direcciones de un usuario ==
```

Captura X. Clase de pruebas para *Direccion* y su salida de ejecución.