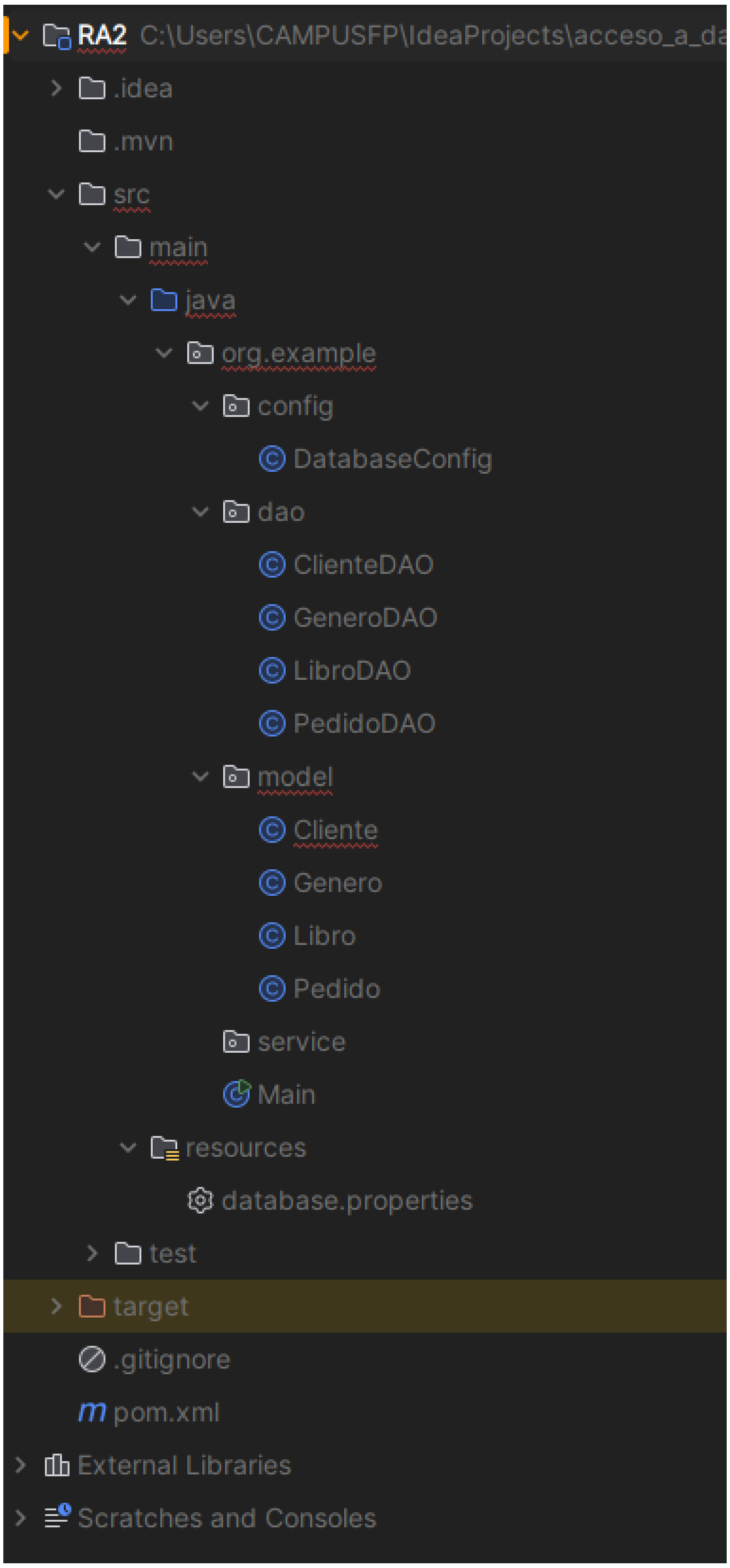
A2 RA2

ACCESO A DATOS

17 DE NOVIEMBRE DE 2025

DANIEL HERNANDEZ HERRERO
2ºDAM

FALLO EN LA CLASE CLIENTE



```
RA2  C:\Users\CAMPUSFP\IdeaProjects\acceso_a_da
  .idea
  .mvn
  src
    main
      java
        org.example
          config
            DatabaseConfig
          dao
            ClienteDAO
            GeneroDAO
            LibroDAO
            PedidoDAO
          model
            Cliente
            Genero
            Libro
            Pedido
          service
          Main
      resources
        database.properties
    test
  target
  .gitignore
  pom.xml
External Libraries
Scratches and Consoles
```

Database.config

```java
package org.example.config;

import com.zaxxer.hikari.HikariConfig;
import com.zaxxer.hikari.HikariDataSource;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.Properties;

public class DatabaseConfig {
    private static HikariDataSource dataSource;

    public static Connection getConnection() throws SQLException {
        return dataSource.getConnection();
    }

    public static void close() {
        if (dataSource != null) {
            dataSource.close();
        }

    }

    static {
        try {
            Properties props = new Properties();

            try (InputStream input =
DatabaseConfig.class.getClassLoader().getResourceAsStream("database.pr
operties")) {
                props.load(input);
            }

            HikariConfig config = new HikariConfig();
            config.setJdbcUrl(props.getProperty("db.url"));
            config.setUsername(props.getProperty("db.user"));
            config.setPassword(props.getProperty("db.password"));
            config.setDriverClassName(props.getProperty("db.driver"));

config.setMaximumPoolSize(Integer.parseInt(props.getProperty("db.poolS
ize", "10")));
            dataSource = new HikariDataSource(config);
        } catch (Exception e) {
            e.printStackTrace();
            throw new RuntimeException("Error configurando el pool de
conexiones", e);
        }
    }
}
```

ClienteDAO.java

```java
package org.example.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import org.example.config.DatabaseConfig;
import org.example.model.Cliente;

public class ClienteDAO {
    public <cliente> int crear(Cliente cliente) {
        String sql = "INSERT INTO cliente (nombre) VALUES (?)";

        try {
            byte var6;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql, 1);
            ) {
                stmt.setString(1, Cliente.getnombre());
                int filas = stmt.executeUpdate();
                if (filas != 0) {
                    try (ResultSet keys = stmt.getGeneratedKeys()) {
                        if (keys.next()) {
                            int var7 = keys.getInt(1);
                            return var7;
                        }

                        return -1;
                    }
                }

                var6 = -1;
            }

            return var6;
        } catch (SQLException e) {
            e.printStackTrace();
            return -1;
        }
    }

    public List<Cliente> obtenerTodos() {
        List<Cliente> lista = new ArrayList();
        String sql = "SELECT * FROM Cliente";

        try (
                Connection conn = DatabaseConfig.getConnection();
                PreparedStatement stmt = conn.prepareStatement(sql);
                ResultSet rs = stmt.executeQuery();
        ) {

        } catch (SQLException e) {
            e.printStackTrace();
        }

        return lista;
    }


    public boolean actualizar(Cliente cliente) {
        String sql = "UPDATE Cliente SET nombre=? WHERE id=?";

        try {
            boolean var5;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql);
            ) {
                stmt.setString(1, Cliente.getnombre());
                var5 = stmt.executeUpdate() > 0;
            }

            return var5;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    public boolean eliminar(int id) {
        String sql = "DELETE FROM Cliente WHERE id=?";

        try {
            boolean var5;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql);
            ) {
                stmt.setInt(1, id);
                var5 = stmt.executeUpdate() > 0;
            }

            return var5;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    private Cliente mapearProyecto(ResultSet rs) throws SQLException {
        return new Cliente(rs.getInt("id"), rs.getString("nombre"));
    }
}
```

GeneroDAO.java

```
package org.example.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import org.example.config.DatabaseConfig;
import org.example.model.Genero;

public class GeneroDAO {
    public static int crear(Genero genero) {
        String sql = "INSERT INTO genero (nombre_genero) VALUES (?)";
        try {
            byte var6;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql, 1);
            ) {
                stmt.setString(1, genero.getnombre_genero());
                int filas = stmt.executeUpdate();
                if (filas != 0) {
                    try (ResultSet keys = stmt.getGeneratedKeys()) {
                        if (keys.next()) {
                            int var7 = keys.getInt(1);
                            return var7;
                        }

                        return -1;
                    }
                }

                var6 = -1;
            }

            return var6;
        } catch (SQLException e) {
            e.printStackTrace();
            return -1;
        }
    }

    public static List<Genero> obtenerTodos() {
        List<Genero> lista = new ArrayList();
        String sql = "SELECT * FROM genero";

        try (
                Connection conn = DatabaseConfig.getConnection();
                PreparedStatement stmt = conn.prepareStatement(sql);
                ResultSet rs = stmt.executeQuery();
        ) {

        } catch (SQLException e) {
            e.printStackTrace();
        }

        return lista;
    }


    public boolean actualizar(Genero genero) {
        String sql = "UPDATE libro SET nombre=? WHERE id=?";

        try {
            boolean var5;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql);
            ) {
                stmt.setString(1, genero.getnombre_genero());
                var5 = stmt.executeUpdate() > 0;
            }

            return var5;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    public boolean eliminar(int id) {
        String sql = "DELETE FROM genero WHERE id=?";

        try {
            boolean var5;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql);
            ) {
                stmt.setInt(1, id);
                var5 = stmt.executeUpdate() > 0;
            }

            return var5;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    private Genero mapearProyecto(ResultSet rs) throws SQLException {
        return new Genero(rs.getString("nombre"));
    }
}
```

LibroDAO.java

```
package org.example.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import org.example.config.DatabaseConfig;
import org.example.model.Libro;

public class LibroDAO {
    public int crear(Libro libro) {
        String sql = "INSERT INTO libro (nombre) VALUES (?)";

        try {
            byte var6;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql, 1);
            ) {
                stmt.setString(1, libro.getNombre());
                int filas = stmt.executeUpdate();
                if (filas != 0) {
                    try (ResultSet keys = stmt.getGeneratedKeys()) {
                        if (keys.next()) {
                            int var7 = keys.getInt(1);
                            return var7;
                        }

                        return -1;
                    }
                }

                var6 = -1;
            }

            return var6;
        } catch (SQLException e) {
            e.printStackTrace();
            return -1;
        }
    }

    public List<Libro> obtenerTodos() {
        List<Libro> lista = new ArrayList();
        String sql = "SELECT * FROM libro";

        try (
                Connection conn = DatabaseConfig.getConnection();
                PreparedStatement stmt = conn.prepareStatement(sql);
                ResultSet rs = stmt.executeQuery();
        ) {
            while(rs.next()) {
                lista.add(this.mapearProyecto(rs));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return lista;
    }


    public boolean actualizar(Libro libro) {
        String sql = "UPDATE libro SET nombre=? WHERE id=?";

        try {
            boolean var5;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql);
            ) {
                stmt.setString(1, libro.getNombre());
                var5 = stmt.executeUpdate() > 0;
            }

            return var5;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    public boolean eliminar(int id) {
        String sql = "DELETE FROM libro WHERE id=?";

        try {
            boolean var5;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql);
            ) {
                stmt.setInt(1, id);
                var5 = stmt.executeUpdate() > 0;
            }

            return var5;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    private Libro mapearProyecto(ResultSet rs) throws SQLException {
        return new Libro(rs.getInt("id"), rs.getString("nombre"));
    }
}
```

PedidoDAO.java

```
package org.example.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import org.example.config.DatabaseConfig;
import org.example.model.Pedido;

public class PedidoDAO {
    public int crear(Connection conn, int id_libro, int id_cliente,
String genero_libro) throws SQLException {
        String sql = "INSERT INTO asignaciones (empleado_id,
proyecto_id) VALUES (?, ?)";

        try (PreparedStatement stmt = conn.prepareStatement(sql, 1)) {
            stmt.setInt(1, id_libro);
            stmt.setInt(2, id_cliente);
            stmt.setString(3, genero_libro);

            int filas = stmt.executeUpdate();
            if (filas == 0) {
                byte var14 = -1;
                return var14;
            }

            try (ResultSet keys = stmt.getGeneratedKeys()) {
                if (keys.next()) {
                    int var8 = keys.getInt(1);
                    return var8;
                }
            }
        }

        return -1;
    }

    public int crear(int id_libro, int id_cliente, String
genero_libro) {
        try (Connection conn = DatabaseConfig.getConnection()) {
            return this.crear(conn, id_libro, id_cliente,
genero_libro);
        } catch (SQLException e) {
            e.printStackTrace();
            return -1;
        }
    }

    public List<Pedido> obtenerTodos() {
        List<Pedido> lista = new ArrayList();
        String sql = "SELECT * FROM libro";

        try (
                Connection conn = DatabaseConfig.getConnection();
                PreparedStatement stmt = conn.prepareStatement(sql);
                ResultSet rs = stmt.executeQuery();
        ) {
            while(rs.next()) {
                lista.add(new Pedido(rs.getInt("id"),
rs.getInt("id_libro"), rs.getInt("id_Cliente"),
rs.getString("genero_libro")));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return lista;
    }

    public boolean eliminar(int id) {
        String sql = "DELETE FROM Pedido WHERE id = ?";

        try {
            boolean var5;
            try (
                    Connection conn = DatabaseConfig.getConnection();
                    PreparedStatement stmt =
conn.prepareStatement(sql);
            ) {
                stmt.setInt(1, id);
                var5 = stmt.executeUpdate() > 0;
            }

            return var5;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }
    public List<Pedido> obtenerPorLibro(int id_libro) {
        List<Pedido> lista = new ArrayList();
        String sql = "SELECT * FROM Pedido WHERE id_libro = ?";

        try (
                Connection conn = DatabaseConfig.getConnection();
                PreparedStatement stmt = conn.prepareStatement(sql);
        ) {
            stmt.setInt(1, id_libro);

            try (ResultSet rs = stmt.executeQuery()) {
                while(rs.next()) {
                    lista.add(new Pedido(rs.getInt("id"),
rs.getInt("id_libro"), rs.getInt("id_cliente"),
rs.getString("genero_libro")));
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return lista;
    }

    public List<Pedido> obtenerPorCliente(int id_cliente) {
        List<Pedido> lista = new ArrayList();
        String sql = "SELECT * FROM pedido WHERE id_cliente = ?";

        try (
                Connection conn = DatabaseConfig.getConnection();
                PreparedStatement stmt = conn.prepareStatement(sql);
        ) {
            stmt.setInt(1, id_cliente);

            try (ResultSet rs = stmt.executeQuery()) {
                while(rs.next()) {
                    lista.add(new Pedido(rs.getInt("id"),
rs.getInt("id_libro"), rs.getInt("id_cliente"),
rs.getString("genero_libro")));
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return lista;
    }

    public List<Pedido> obtenerPorGenero(int genero_libro) {
        List<Pedido> lista = new ArrayList();
        String sql = "SELECT * FROM pedido WHERE genero_libro = ?";

        try (
                Connection conn = DatabaseConfig.getConnection();
                PreparedStatement stmt = conn.prepareStatement(sql);
        ) {
            stmt.setInt(1, genero_libro);

            try (ResultSet rs = stmt.executeQuery()) {
                while(rs.next()) {
                    lista.add(new Pedido(rs.getInt("id"),
rs.getInt("id_libro"), rs.getInt("id_cliente"),
rs.getString("genero_libro")));
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return lista;
    }
}
```

Cliente.java

```
package org.example.model;

public class Cliente {
    private int id;
    private String nombre;

    public Cliente() {
    }

    public Cliente(int id, String nombre) {
        this.id = id;
        this.nombre = nombre;
    }

    public int getId() {
        return this.id;
    }

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

    public static String getnombre() {
        return nombre;
    }

    public void setnombre(String nombre) {
        this.nombre = nombre;
    }

    public String toString() {
        return "Cliente{id=" + this.id + ", nombre=" + this.nombre + "
}";
    }
}
```

Genero.java

```
package org.example.model;

public class Genero {
    private String nombre_genero;
```

```java
    public Genero() {
    }

    public Genero(String nombre_genero) {
        this.nombre_genero = nombre_genero;
    }


    public String getnombre_genero() {
        return this.nombre_genero;
    }

    public void setnombre_genero(String nombre_genero) {
        this.nombre_genero = nombre_genero;
    }

    public String toString() {
        return "Genero{nombre=" + this.nombre_genero + "}";
    }
}
```

Libro.java

```java
package org.example.model;

public class Libro {
    private int id;
    private String nombre;

    public Libro() {
    }

    public Libro(int id, String nombre) {
        this.id = id;
        this.nombre = nombre;
    }

    public int getId() {
        return this.id;
    }

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

    public String getNombre() {
        return this.nombre;
    }


    public String toString() {
        return "Clo{id=" + this.id + ", nombre='" + this.nombre + "}";
    }
}
```

Pedido.java

```java
package org.example.model;

public class Pedido {
    private int id;
    private int id_libro;
    private int id_cliente;
    private String genero_libro;

    public Pedido() {
    }

    public Pedido(int id, int id_libro, int id_cliente, String
genero_libro) {
        this.id = id;
        this.id_libro = id_libro;
        this.id_cliente = id_cliente;
        this.genero_libro = genero_libro;
    }

    public int getId() {
        return this.id;
    }

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

    public int getid_libro() {
        return this.id_libro;
    }

    public void setid_libro(int id_libro) {
        this.id_libro = id_libro;
    }

    public int getid_cliente() {
        return this.id_cliente;
    }

    public void setid_cliente(int id_cliente) {
        this.id_cliente = id_cliente;
    }

    public String getgenero_libro() {
        return this.genero_libro;
    }

    public void setgenero_libro(String genero_libro) {
        this.genero_libro = genero_libro;
    }

    public String toString() {
        return "Pedido{id=" + this.id + ", id_libro=" + this.id_libro
+ ", id_cliente=" + this.id_cliente + ", genero_libro= " +
this.genero_libro + " }";
    }
}
```

Main.java

```java
package org.example;

import java.io.PrintStream;
import java.math.BigDecimal;
import java.time.LocalDate;
import java.util.List;
import java.util.Objects;
import org.example.dao.GeneroDAO;
import org.example.dao.LibroDAO;
import org.example.dao.PedidoDAO;
import org.example.model.Genero;
import org.example.model.Libro;


public class Main {
    public static void main(String[] args) {
        LibroDAO libroDAO = new LibroDAO();
        GeneroDAO generoDAO = new GeneroDAO();
        PedidoDAO pedidoDAO = new PedidoDAO();
        Libro e1 = new Libro(2, "Luis");
        int id1 = libroDAO.crear(e1);
        System.out.println("Empleados creados con IDs: " + id1 + "
.");
        Genero p1 = new Genero("terror");
        int pid1 = GeneroDAO.crear(p1);
        System.out.println("Proyectos creados con IDs: " + pid1 +
".");
        List<Libro> libro = libroDAO.obtenerTodos();
        PrintStream var10001 = System.out;
        Objects.requireNonNull(var10001);
        libro.forEach(var10001::println);
        List<Genero> proyectos = GeneroDAO.obtenerTodos();
    }
}
```