

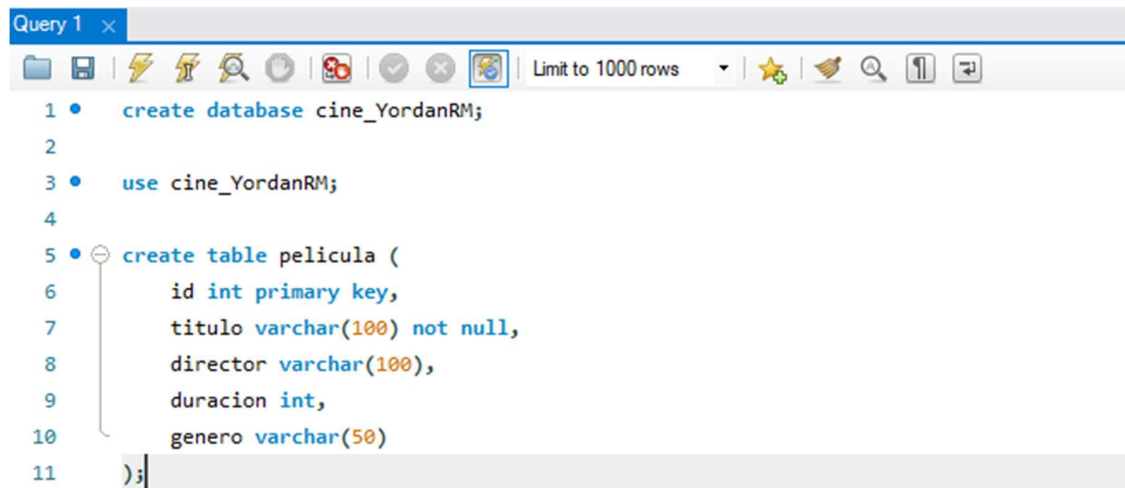
HITO INDIVIDUAL PROGRAMACIÓN

Yordan Rosenov Matov
09/05/2025

ÍNDICE

Base de datos.....	3
Pelicula.java	4
DBConnection.java.....	5
Main.java	6
PeliculaDAO.java	8
GitHub.....	14

Base de datos



The screenshot shows a SQL query editor window titled "Query 1". The editor contains the following SQL code:

```
1 • create database cine_YordanRM;
2
3 • use cine_YordanRM;
4
5 • create table pelicula (
6     id int primary key,
7     titulo varchar(100) not null,
8     director varchar(100),
9     duracion int,
10    genero varchar(50)
11 );
```

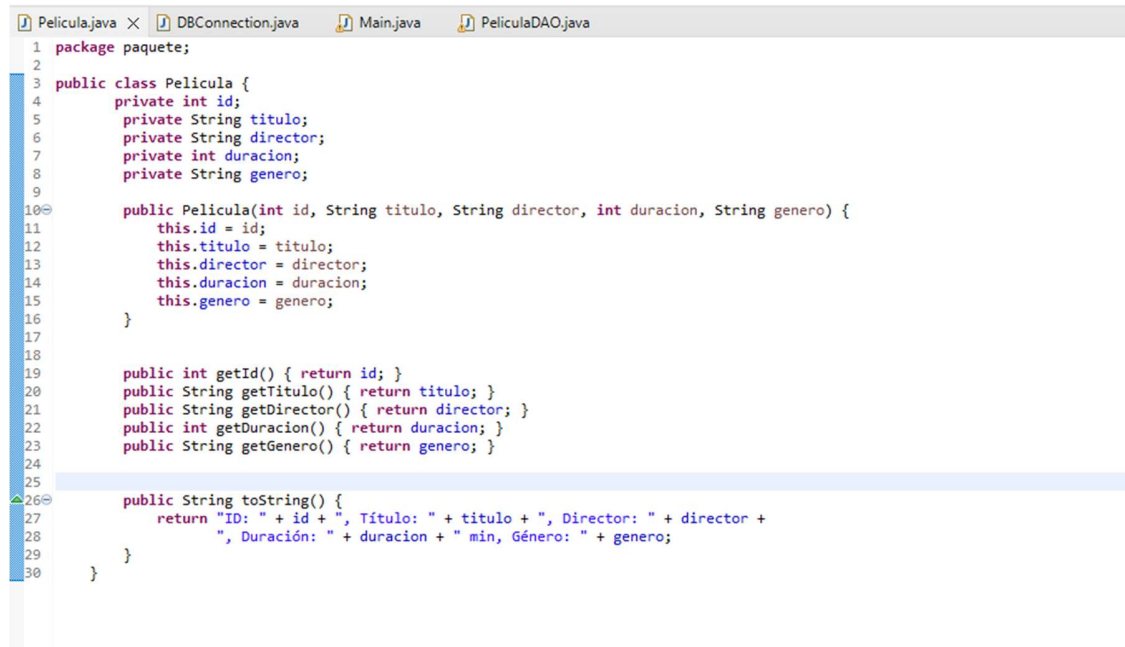
The interface includes a toolbar with various icons for file operations, execution, and viewing. A dropdown menu indicates "Limit to 1000 rows".

```
create database cine_YordanRM;
```

```
use cine_YordanRM;
```

```
create table pelicula (
    id int primary key,
    titulo varchar(100) not null,
    director varchar(100),
    duracion int,
    genero varchar(50)
);
```

Pelicula.java



```
1 package paquete;
2
3 public class Pelicula {
4     private int id;
5     private String titulo;
6     private String director;
7     private int duracion;
8     private String genero;
9
10    public Pelicula(int id, String titulo, String director, int duracion, String genero) {
11        this.id = id;
12        this.titulo = titulo;
13        this.director = director;
14        this.duracion = duracion;
15        this.genero = genero;
16    }
17
18    public int getId() { return id; }
19    public String getTitulo() { return titulo; }
20    public String getDirector() { return director; }
21    public int getDuracion() { return duracion; }
22    public String getGenero() { return genero; }
23
24
25    public String toString() {
26        return "ID: " + id + ", Título: " + titulo + ", Director: " + director +
27            ", Duración: " + duracion + " min, Género: " + genero;
28    }
29 }
30
```

```
package paquete;
```

```
public class Pelicula {
```

```
    private int id;
    private String titulo;
    private String director;
    private int duracion;
    private String genero;
```

```
    public Pelicula(int id, String titulo, String director, int
duracion, String genero) {
```

```
        this.id = id;
        this.titulo = titulo;
        this.director = director;
        this.duracion = duracion;
        this.genero = genero;
    }
```

```
    public int getId() { return id; }
    public String getTitulo() { return titulo; }
    public String getDirector() { return director; }
    public int getDuracion() { return duracion; }
    public String getGenero() { return genero; }
```

```
    public String toString() {
        return "ID: " + id + ", Título: " + titulo + ", Director: " +
director +
```

```

        ", Duración: " + duracion + " min, Género: " + genero;
    }
}

```

DBConnection.java

```

1 package paquete;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5
6 public class DBConnection {
7     private static Connection connection = null;
8
9     public static Connection getConnection() {
10         try {
11             if (connection == null || connection.isClosed()) {
12                 String url = "jdbc:mysql://localhost:3306/cine_YordanRM";
13                 String user = "root";
14                 String password = "1234";
15
16                 connection = DriverManager.getConnection(url, user, password);
17             }
18         } catch (SQLException e) {
19             System.out.println("Error al conectar a la base de datos.");
20             e.printStackTrace();
21         }
22         return connection;
23     }
24 }
25

```

```

package paquete;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
    private static Connection connection = null;

    public static Connection getConnection() {
        try {
            if (connection == null || connection.isClosed()) {
                String url =
"jdbc:mysql://localhost:3306/cine_YordanRM";
                String user = "root";
                String password = "1234";

                connection = DriverManager.getConnection(url, user,
password);
            }
        } catch (SQLException e) {
            System.out.println("Error al conectar a la base de datos.");
            e.printStackTrace();
        }
        return connection;
    }
}

```

```

    }
} catch (SQLException e) {
    System.out.println("Error al conectar a la base de datos.");
    e.printStackTrace();
}
return connection;
}
}
}

```

Main.java

```

1 package paquete;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.util.Scanner;
6
7 public class Main {
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10         PeliculaDAO peliculaDAO = new PeliculaDAO();
11
12         int opcion;
13         do {
14             System.out.println("\n--- MENÚ ---");
15             System.out.println("1 - Ver Peliculas");
16             System.out.println("2 - Añadir Pelicula");
17             System.out.println("3 - Eliminar Pelicula");
18             System.out.println("4 - Modificar Pelicula");
19             System.out.println("5 - Salir");
20             System.out.print("Elige una opción: ");
21
22             try {
23                 opcion = Integer.parseInt(sc.nextLine());
24                 switch (opcion) {
25                     case 1:
26                         peliculaDAO.verPeliculas();
27                         break;
28                     case 2:
29                         peliculaDAO.addPelicula(sc);
30                         break;
31                     case 3:
32                         peliculaDAO.eliminarPelicula(sc);
33                         break;
34                     case 4:
35                         peliculaDAO.modificarPelicula(sc);
36                         break;
37                     case 5:
38                         System.out.println("Saliendo del programa...");
39                         break;
40                     default:
41                         System.out.println("Opción no válida.");
42                 }
43             } catch (NumberFormatException e) {
44                 System.out.println("Por favor, introduce un número válido.");
45                 opcion = 0;
46             }
47         } while (opcion != 5);
48
49         sc.close();
50     }
51 }

```

```

package paquete;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

```

```

PelículaDAO peliculaDAO = new PelículaDAO();

int opcion;
do {
    System.out.println("\n--- MENÚ ---");
    System.out.println("1 - Ver Películas");
    System.out.println("2 - Añadir Película");
    System.out.println("3 - Eliminar Película");
    System.out.println("4 - Modificar Película");
    System.out.println("5 - Salir");
    System.out.print("Elige una opción: ");

    try {
        opcion = Integer.parseInt(sc.nextLine());
        switch (opcion) {
            case 1:
                peliculaDAO.verPelículas();
                break;
            case 2:
                peliculaDAO.addPelícula(sc);
                break;
            case 3:
                peliculaDAO.eliminarPelícula(sc);
                break;
            case 4:
                peliculaDAO.modificarPelícula(sc);
                break;
            case 5:
                System.out.println("Saliendo del programa...");
                break;
            default:
                System.out.println("Opción no válida.");
        }
    } catch (NumberFormatException e) {
        System.out.println("Por favor, introduce un número válido.");
        opcion = 0;
    }
} while (opcion != 5);

sc.close();
}
}

```

PeliculaDAO.java

```
Pelicula.java DBConnection.java Main.java PeliculaDAO.java X
1 package paquete;
2 import java.sql.*;
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.Scanner;
6
7 public class PeliculaDAO {
8     public void verPeliculas() {
9         String sql = "SELECT * FROM peliculas";
10        try (Connection conn = DBConnection.getConnection();
11             Statement stmt = conn.createStatement();
12             ResultSet rs = stmt.executeQuery(sql)) {
13
14            while (rs.next()) {
15                System.out.println(new Pelicula(
16                    rs.getInt("id"),
17                    rs.getString("titulo"),
18                    rs.getString("director"),
19                    rs.getInt("duracion"),
20                    rs.getString("genero")
21                ));
22            }
23        } catch (SQLException e) {
24            System.out.println("Error al mostrar las películas.");
25            e.printStackTrace();
26        }
27    }
28 }
```



```

29
30 public void addPelícula(Scanner sc) {
31     System.out.print("Introduce ID: ");
32     int id = Integer.parseInt(sc.nextLine());
33
34     if (existePelícula(id)) {
35         System.out.println("Error: Ya existe una película con ese ID.");
36         return;
37     }
38
39     System.out.print("Título: ");
40     String titulo = sc.nextLine();
41
42     System.out.print("Director: ");
43     String director = sc.nextLine();
44
45     System.out.print("Duración (en minutos): ");
46     int duracion = Integer.parseInt(sc.nextLine());
47
48     System.out.print("Género: ");
49     String genero = sc.nextLine();
50
51     String sql = "INSERT INTO peliculas(id, titulo, director, duracion, genero) VALUES (?, ?, ?, ?, ?)";
52     try (Connection conn = DBConnection.getConnection();
53         PreparedStatement pstmt = conn.prepareStatement(sql)) {
54
55         pstmt.setInt(1, id);
56         pstmt.setString(2, titulo);
57         pstmt.setString(3, director);
58         pstmt.setInt(4, duracion);
59         pstmt.setString(5, genero);
60
61         pstmt.executeUpdate();
62         System.out.println("Película añadida correctamente.");
63
64     } catch (SQLException | NumberFormatException e) {
65         System.out.println("Error al añadir la película.");
66         e.printStackTrace();
67     }
68 }
69
70
71 public void eliminarPelícula(Scanner sc) {
72     System.out.print("Introduce el ID de la película a eliminar: ");
73     int id = Integer.parseInt(sc.nextLine());
74
75     if (!existePelícula(id)) {
76         System.out.println("Error: No existe una película con ese ID.");
77         return;
78     }
79
80     String sql = "DELETE FROM peliculas WHERE id = ?";
81     try (Connection conn = DBConnection.getConnection();
82         PreparedStatement pstmt = conn.prepareStatement(sql)) {
83
84         pstmt.setInt(1, id);
85         pstmt.executeUpdate();
86         System.out.println("Película eliminada correctamente.");
87
88     } catch (SQLException | NumberFormatException e) {
89         System.out.println("Error al eliminar la película.");
90         e.printStackTrace();
91     }
92 }
93

```

```

94
95 public void modificarPelicula(Scanner sc) {
96     System.out.print("Introduce el ID de la película a modificar: ");
97     int id = Integer.parseInt(sc.nextLine());
98
99     if (!existePelicula(id)) {
100         System.out.println("Error: No existe una película con ese ID.");
101         return;
102     }
103
104     System.out.print("Nuevo título (dejar vacío si no se quiere cambiar): ");
105     String nuevoTitulo = sc.nextLine();
106
107     System.out.print("Nuevo director (dejar vacío si no se quiere cambiar): ");
108     String nuevoDirector = sc.nextLine();
109
110     StringBuilder sql = new StringBuilder("UPDATE peliculas SET ");
111     boolean hayDatos = false;
112
113     if (!nuevoTitulo.isEmpty()) {
114         sql.append("titulo = ").append(nuevoTitulo).append(", ");
115         hayDatos = true;
116     }
117
118     if (!nuevoDirector.isEmpty()) {
119         sql.append("director = ").append(nuevoDirector).append(", ");
120         hayDatos = true;
121     }
122
123     if (!hayDatos) {
124         System.out.println("No se han introducido cambios.");
125         return;
126     }
127
128
129     sql.setLength(sql.length() - 2);
130     sql.append(" WHERE id = ").append(id);
131
132     try (Connection conn = DBConnection.getConnection();
133         Statement stmt = conn.createStatement()) {
134
135         stmt.executeUpdate(sql.toString());
136         System.out.println("Película modificada correctamente.");
137
138     } catch (SQLException e) {
139         System.out.println("Error al modificar la película.");
140         e.printStackTrace();
141     }
142 }
143
...

```

```

144
145 private boolean existePelicula(int id) {
146     String sql = "SELECT COUNT(*) FROM peliculas WHERE id = ?";
147     try (Connection conn = DBConnection.getConnection();
148         PreparedStatement pstmt = conn.prepareStatement(sql)) {
149
150         pstmt.setInt(1, id);
151         ResultSet rs = pstmt.executeQuery();
152         if (rs.next()) {
153             return rs.getInt(1) > 0;
154         }
155     } catch (SQLException e) {
156         System.out.println("Error al verificar existencia de película.");
157         e.printStackTrace();
158     }
159     return false;
160 }
161 }

```

```

package paquete;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

```

```

public class PeliculaDAO {
    public void verPeliculas() {
        String sql = "SELECT * FROM peliculas";
        try (Connection conn = DBConnection.getConnection();
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql)) {

            while (rs.next()) {
                System.out.println(new Pelicula(
                    rs.getInt("id"),
                    rs.getString("titulo"),
                    rs.getString("director"),
                    rs.getInt("duracion"),
                    rs.getString("genero")
                ));
            }
        } catch (SQLException e) {
            System.out.println("Error al mostrar las películas.");
            e.printStackTrace();
        }
    }

    public void addPelicula(Scanner sc) {
        System.out.print("Introduce ID: ");
        int id = Integer.parseInt(sc.nextLine());

        if (existePelicula(id)) {
            System.out.println("Error: Ya existe una película con ese ID.");
        }
    }
}

```

```

        return;
    }

    System.out.print("Título: ");
    String titulo = sc.nextLine();

    System.out.print("Director: ");
    String director = sc.nextLine();

    System.out.print("Duración (en minutos): ");
    int duracion = Integer.parseInt(sc.nextLine());

    System.out.print("Género: ");
    String genero = sc.nextLine();

    String sql = "INSERT INTO peliculas(id, titulo, director, duracion,
genero) VALUES (?, ?, ?, ?, ?)";
    try (Connection conn = DBConnection.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setInt(1, id);
        pstmt.setString(2, titulo);
        pstmt.setString(3, director);
        pstmt.setInt(4, duracion);
        pstmt.setString(5, genero);

        pstmt.executeUpdate();
        System.out.println("Película añadida correctamente.");
    } catch (SQLException | NumberFormatException e) {
        System.out.println("Error al añadir la película.");
        e.printStackTrace();
    }
}

public void eliminarPelicula(Scanner sc) {
    System.out.print("Introduce el ID de la película a eliminar: ");
    int id = Integer.parseInt(sc.nextLine());

    if (!existePelicula(id)) {
        System.out.println("Error: No existe una película con ese ID.");
        return;
    }

    String sql = "DELETE FROM peliculas WHERE id = ?";
    try (Connection conn = DBConnection.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setInt(1, id);
        pstmt.executeUpdate();
        System.out.println("Película eliminada correctamente.");
    }
}

```

```

    } catch (SQLException | NumberFormatException e) {
        System.out.println("Error al eliminar la película.");
        e.printStackTrace();
    }
}

public void modificarPelicula(Scanner sc) {
    System.out.print("Introduce el ID de la película a modificar: ");
    int id = Integer.parseInt(sc.nextLine());

    if (!existePelicula(id)) {
        System.out.println("Error: No existe una película con ese ID.");
        return;
    }

    System.out.print("Nuevo título (dejar vacío si no se quiere cambiar): ");
    String nuevoTitulo = sc.nextLine();

    System.out.print("Nuevo director (dejar vacío si no se quiere cambiar): ");
    String nuevoDirector = sc.nextLine();

    StringBuilder sql = new StringBuilder("UPDATE peliculas SET ");
    boolean hayDatos = false;

    if (!nuevoTitulo.isEmpty()) {
        sql.append("titulo = ").append(nuevoTitulo).append(", ");
        hayDatos = true;
    }

    if (!nuevoDirector.isEmpty()) {
        sql.append("director = ").append(nuevoDirector).append(", ");
        hayDatos = true;
    }

    if (!hayDatos) {
        System.out.println("No se han introducido cambios.");
        return;
    }

    sql.setLength(sql.length() - 2);
    sql.append(" WHERE id = ").append(id);

    try (Connection conn = DBConnection.getConnection();
        Statement stmt = conn.createStatement()) {

        stmt.executeUpdate(sql.toString());
        System.out.println("Película modificada correctamente.");
    }
}

```

```

    } catch (SQLException e) {
        System.out.println("Error al modificar la película.");
        e.printStackTrace();
    }
}

private boolean existePelícula(int id) {
    String sql = "SELECT COUNT(*) FROM peliculas WHERE id = ?";
    try (Connection conn = DBConnection.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setInt(1, id);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            return rs.getInt(1) > 0;
        }
    } catch (SQLException e) {
        System.out.println("Error al verificar existencia de película.");
        e.printStackTrace();
    }
    return false;
}
}

```

GitHub

<https://github.com/CR1PTAS/programaci-n.git>