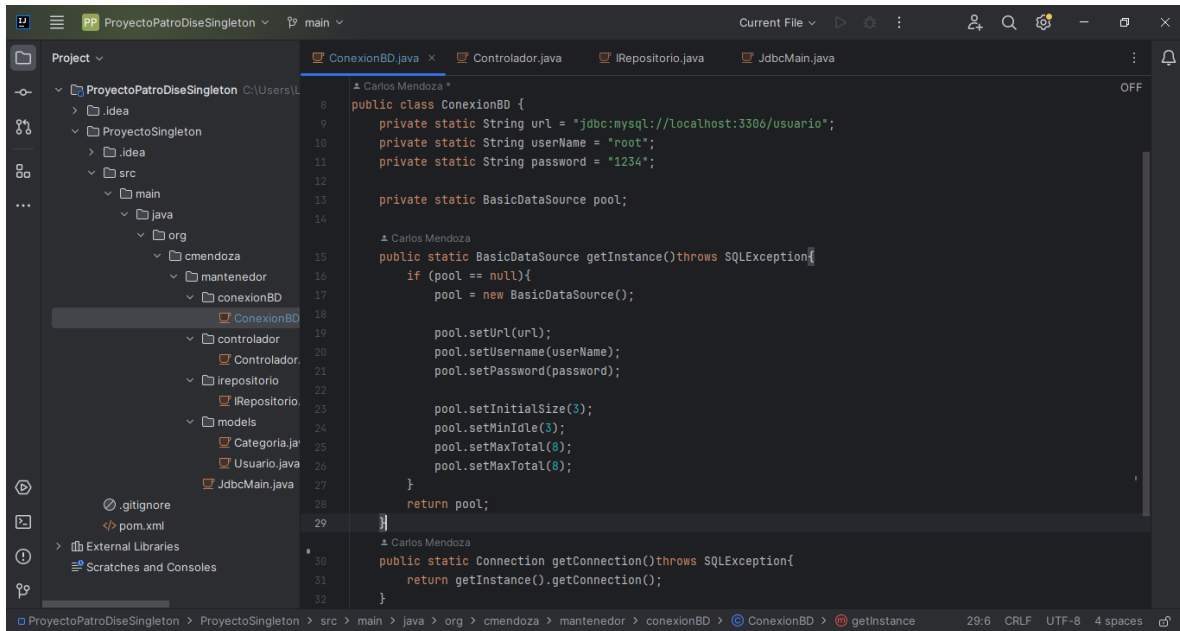


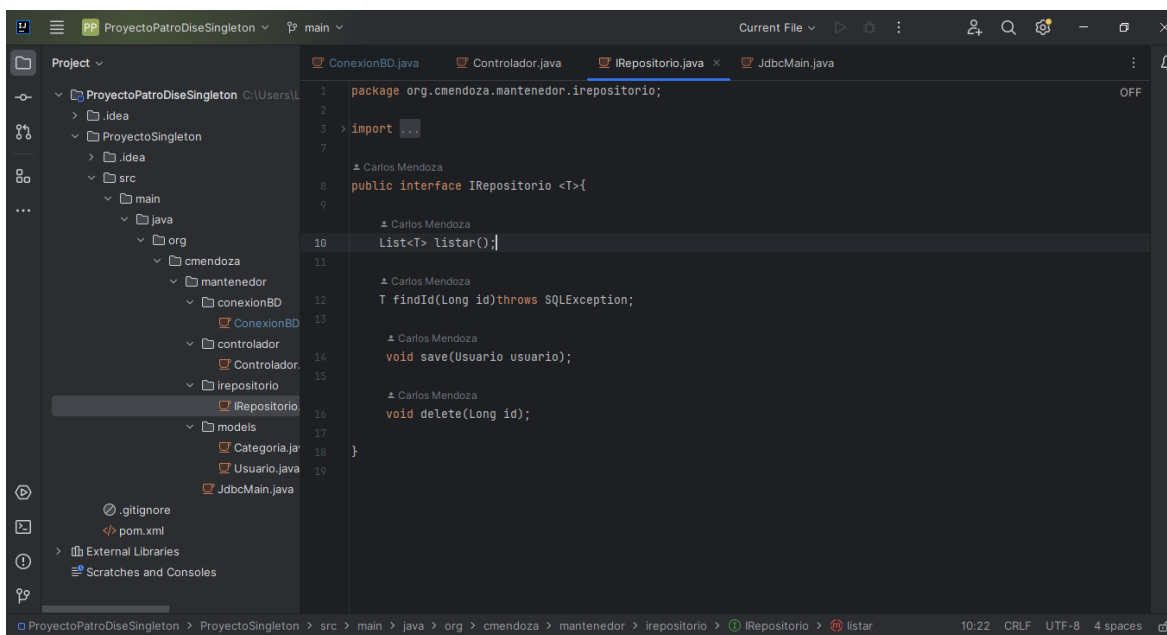
Programa Patrón de diseño singleton

Este programa realiza un Crud de usuarios con un singleton para la conexión a la base de datos



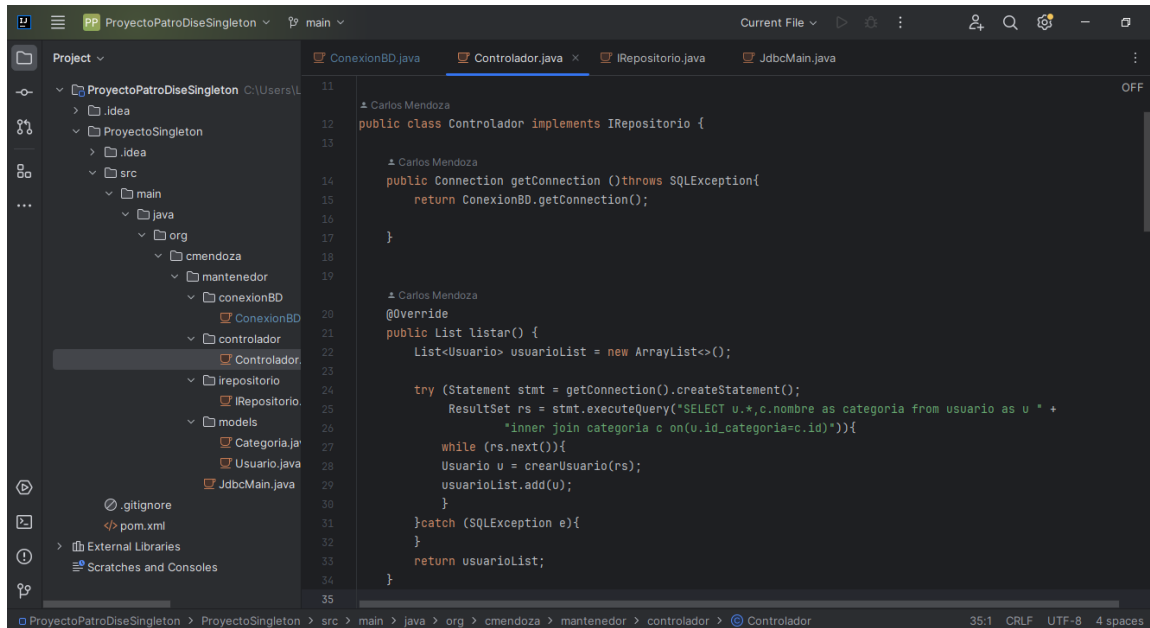
```
1 public class ConexionBD {
2     private static String url = "jdbc:mysql://localhost:3306/usuario";
3     private static String userName = "root";
4     private static String password = "1234";
5
6     private static BasicDataSource pool;
7
8     public static BasicDataSource getInstance() throws SQLException {
9         if (pool == null) {
10             pool = new BasicDataSource();
11
12             pool.setUrl(url);
13             pool.setUsername(userName);
14             pool.setPassword(password);
15
16             pool.setInitialSize(3);
17             pool.setMinIdle(3);
18             pool.setMaxTotal(8);
19             pool.setMaxTotal(8);
20         }
21         return pool;
22     }
23 }
24
25 public static Connection getConnection() throws SQLException {
26     return getInstance().getConnection();
27 }
28 }
```

Interfaz que contiene los métodos CRUD

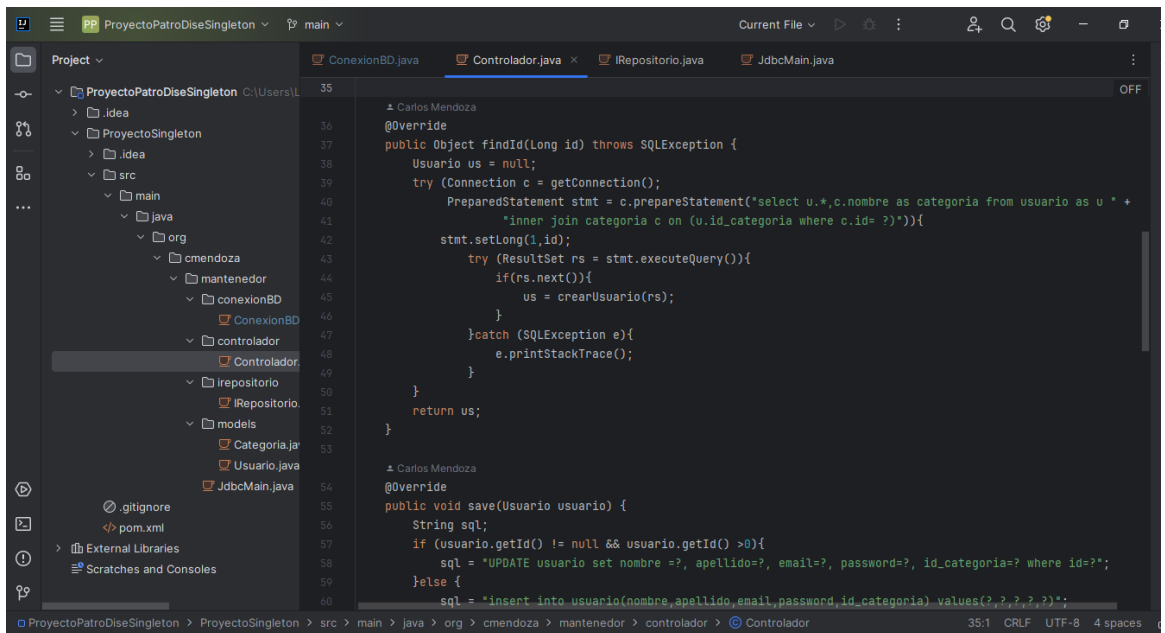


```
1 package org.cmendoza.mantenedor.irepositorio;
2
3 import java.util.List;
4
5 public interface IRepository<T> {
6
7     List<T> listar();
8
9     T findById(Long id) throws SQLException;
10
11     void save(Usuario usuario);
12
13     void delete(Long id);
14 }
15
16 }
```

Controlador para realizar las consultas a la base de datos



```
11
12 public class Controlador implements IRepository {
13
14     Carlos Mendoza
15     public Connection getConnection ()throws SQLException{
16         return ConexionBD.getConnection();
17     }
18
19     Carlos Mendoza
20     @Override
21     public List listar() {
22         List<Usuario> usuariolist = new ArrayList<>();
23
24         try (Statement stmt = getConnection().createStatement();
25             ResultSet rs = stmt.executeQuery("SELECT u.*,c.nombre as categoria from usuario as u " +
26                 "inner join categoria c on(u.id_categoria=c.id)")){
27             while (rs.next()){
28                 Usuario u = crearUsuario(rs);
29                 usuariolist.add(u);
30             }
31         }catch (SQLException e){
32             }
33         return usuariolist;
34     }
35 }
```



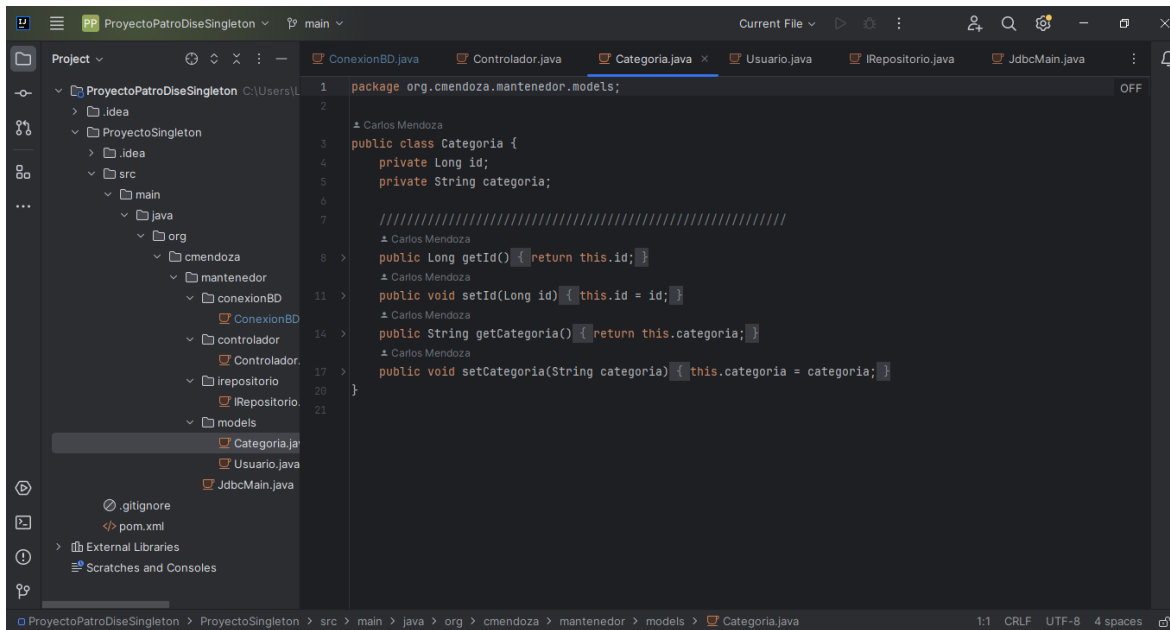
```
35
36 Carlos Mendoza
37 @Override
38 public Object findId(Long id) throws SQLException {
39     Usuario us = null;
40     try (Connection c = getConnection();
41         PreparedStatement stmt = c.prepareStatement("select u.*,c.nombre as categoria from usuario as u " +
42             "inner join categoria c on (u.id_categoria where c.id= ?)")){
43         stmt.setLong(1,id);
44         try (ResultSet rs = stmt.executeQuery()){
45             if(rs.next()){
46                 us = crearUsuario(rs);
47             }
48         }catch (SQLException e){
49             e.printStackTrace();
50         }
51     }
52     return us;
53 }
54 Carlos Mendoza
55 @Override
56 public void save(Usuario usuario) {
57     String sql;
58     if (usuario.getId() != null && usuario.getId() >0){
59         sql = "UPDATE usuario set nombre=?, apellido=?, email=?, password=?, id_categoria=? where id=?";
60     }else {
61         sql = "insert into usuario(nombre,apellido,email,password,id_categoria) values(?,?,?,?,?)";
62     }
63 }
```

```
81  @Override
82  public void delete(Long id) {
83      try(Connection c = getConnection();
84          PreparedStatement stmt = c.prepareStatement("delete from usuario where id = ?")) {
85          stmt.setLong(1,id);
86          stmt.executeUpdate();
87      } catch (SQLException e) {
88          throw new RuntimeException(e);
89      }
90  }
91
92  public static Usuario crearUsuario(ResultSet rs) throws SQLException{
93      Usuario us = new Usuario();
94      us.setId(rs.getLong("id"));
95      us.setNombre(rs.getString("nombre"));
96      us.setApellido(rs.getString("apellido"));
97      us.setEmail(rs.getString("email"));
98      us.setPassword(rs.getString("password"));
99      Categoria ct = new Categoria();
100      ct.setId(rs.getLong("id"));
101      ct.setCategoria(rs.getString("categoria"));
102      us.setCategoria(ct);
103      return us;
104  }
```

Clase usuario

```
3  public class Usuario {
4      private Long id;
5      private String nombre;
6      private String apellido;
7      private String email;
8      private String password;
9      private Categoria categoria;
10
11      //////////////////////////////////
12  public Long getId() { return this.id; }
13
14  public void setId(Long id) { this.id = id; }
15
16  public String getNombre() { return this.nombre; }
17
18  public void setNombre(String nombre) { this.nombre = nombre; }
19
20  public String getApellido() { return this.apellido; }
21
22  public void setApellido(String apellido) { this.apellido = apellido; }
23
24  public String getEmail() { return this.email; }
25
26  public void setEmail(String email) { this.email = email; }
27
28  public String getPassword() { return this.password; }
29  }
```

Clase categoría



The screenshot shows an IDE window with the project 'ProyectoPatroDiseSingleton' open. The file explorer on the left shows the project structure, with the 'Categoria.java' file selected under the 'models' package. The main editor displays the code for the 'Categoria' class, which is a simple entity class with an 'id' and a 'categoria' attribute, and methods to get and set these values. The code is as follows:

```
1 package org.cmendoza.mantenedor.models;
2
3 public class Categoria {
4     private Long id;
5     private String categoria;
6
7     ///////////////////////////////////////////////////
8     public Long getId() { return this.id; }
9     public void setId(Long id) { this.id = id; }
10    public String getCategoria() { return this.categoria; }
11    public void setCategoria(String categoria) { this.categoria = categoria; }
12 }
```

The status bar at the bottom indicates the file is 'Categoria.java' in the package 'org.cmendoza.mantenedor.models', with a line number of 1:1, CRLF line endings, UTF-8 encoding, and 4 spaces for indentation.