

## **LABORATORIO 7 CVDS**

Presentado por:

Gabriela Castro Santamaria

Daniela García Romero

Presentado a:

Iván Darío Viasus

Escuela Colombiana de Ingeniería Julio Garavito

Materia: CVDS

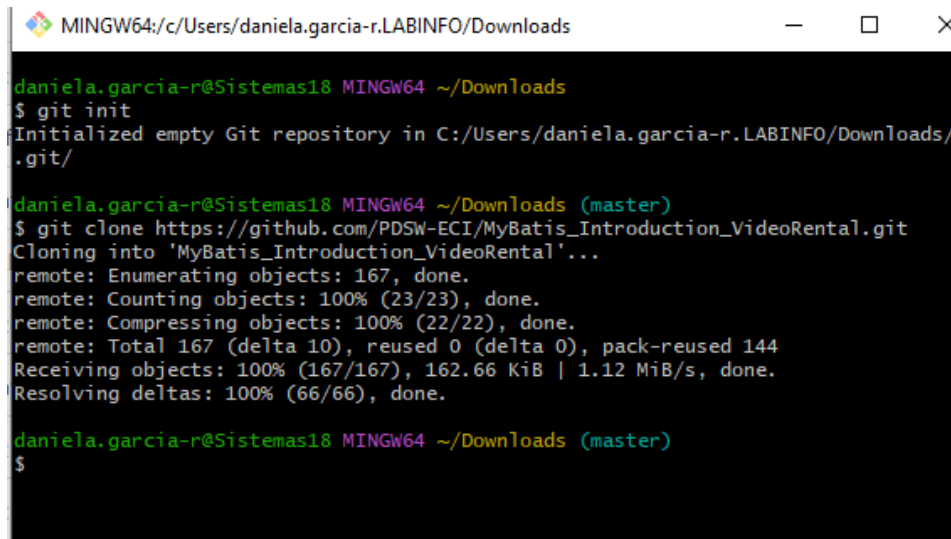
Grupo: 2

29 de septiembre de 2022

## LABORATORIO 7 CVDS

### SECCIÓN I

1. Clonar el proyecto MyBatis\_Introduction\_VideoRental de GitHub donde se realizará la implementación completa del laboratorio.

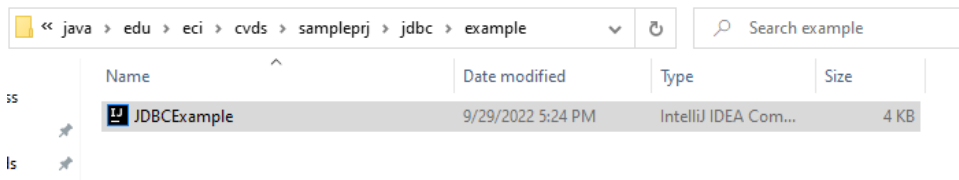
A terminal window titled 'MINGW64: c:/Users/daniela.garcia-r.LABINFO/Downloads' showing the execution of git commands. The user runs 'git init' and 'git clone https://github.com/PDSW-ECI/MyBatis\_Introduction\_VideoRental.git'. The output shows the repository being initialized and then cloned from GitHub. The clone operation is successful, with progress bars for enumerating, counting, and compressing objects, and a final message indicating the repository is ready for use.

```
daniela.garcia-r@Sistemas18 MINGW64 ~/Downloads
$ git init
Initialized empty Git repository in C:/Users/daniela.garcia-r.LABINFO/Downloads/.git/

daniela.garcia-r@Sistemas18 MINGW64 ~/Downloads (master)
$ git clone https://github.com/PDSW-ECI/MyBatis_Introduction_VideoRental.git
Cloning into 'MyBatis_Introduction_VideoRental'...
remote: Enumerating objects: 167, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 167 (delta 10), reused 0 (delta 0), pack-reused 144
Receiving objects: 100% (167/167), 162.66 KiB | 1.12 MiB/s, done.
Resolving deltas: 100% (66/66), done.

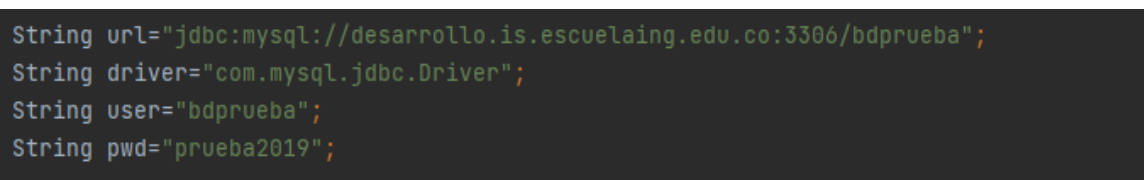
daniela.garcia-r@Sistemas18 MINGW64 ~/Downloads (master)
$
```

2. Descargue el archivo JDBCExample.java y agreguelo en el paquete "edu.eci.cvds.sampleprj.jdbc.example".



3. Desde esta clase se realizará una conexión a una base de datos MySQL por medio de JDBC y sus "Prepared Statements".

6. En la clase JDBCExample juste los parámetros de conexión a la base de datos con los datos reales

A code snippet showing the configuration of JDBC connection parameters in a Java string. The parameters include the URL, driver class, username, and password.

```
String url="jdbc:mysql://desarrollo.is.escuelaing.edu.co:3306/bdprueba";
String driver="com.mysql.jdbc.Driver";
String user="bdprueba";
String pwd="prueba2019";
```

7. Implementar las operaciones faltantes:

- NombresProductosPedido

```
public static List<String> nombresProductosPedido(Connection con, int codigoPedido){
    List<String> np=new LinkedList<>();

    String query = "SELECT * from ORD_PRODUCTOS join ORD_DETALLE_PEDIDO ON (ORD_DETALLE_PEDIDO.pedido_fk = ORD_PRODUCTOS.codigo) WHERE ORD_PRODUCTOS.codigo = ";
    //Crear prepared statement
    try(PreparedStatement statement = con.prepareStatement(query)) {
        //asignar parámetros
        nombresPedido.setInt(1, codigoPedido);

        //usar executeQuery
        Result rs = nombresPedidos.executeQuery();

        //Sacar resultados del ResultSet
        while (rs.next()) {
            String name = rs.getString("nombre");

            //Llenar la lista y retornarla
            np.add(name);
        }
    }catch(SQL.Exception e){
        System.out.println("El dato ya existe en la base de datos");
    }

    return np;
}
```

- ValorTotalPedido

```
//
public static int valorTotalPedido(Connection con, int codigoPedido){
    int value = 0;
    String query = "SELECT SUM(ORD_PRODUCTOS.precio * ORD_DETALLE_PEDIDO.cantidad) AS res FROM ORD_DETAL";

    //Crear prepared statement
    try(PreparedStatement statement = con.prepareStatement(query)) {
        //asignar parámetros
        nombresPedido.setInt(1, codigoPedido);
        //usar executeQuery
        ResultSet res = statement.executeQuery();

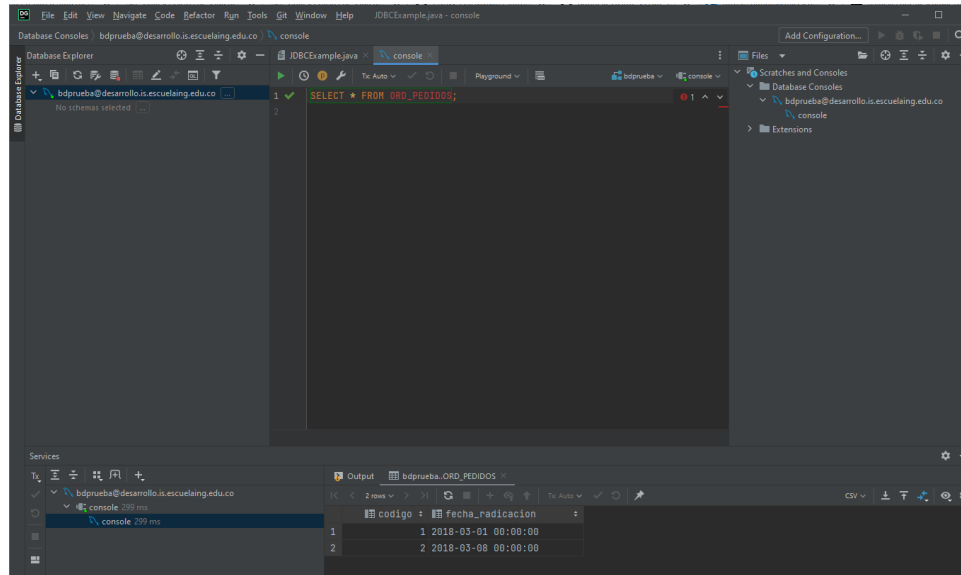
        //Sacar resultado del ResultSet
        while (res.next()) {
            value = res.getInt("res");
        }
    }catch (Exception e){
        e.printStackTrace();
    }

    returnn value;
}
```

- RegistrarNuevoProducto

```
public static void registrarNuevoProducto(Connection con, int codigo, String nombre,int precio) throws SQLException{
    //Crear preparedStatement
    String createProduct = "INSERT INTO ORD_PRODUCTOS VALUES(?, ?, ?)";
    try(PreparedStatement statement = con.prepareStatement(createProduct)){
        //Asignar parámetros
        statement.setInt(1, codigo);
        statement.setString(2, nombre);
        statement.setInt(3, precio);
        //usar 'execute'
        statement.execute();
        con.commit();
    } catch (Exception e){
        e.printStackTrace();
    }
    con.commit();
}
```

8. Verifique por medio de un cliente SQL, que la información retornada por el programa coincide con la que se encuentra almacenada en base de datos.



## SECCIÓN II

### PARTE I

1. Ubique los archivos de configuración para producción de MyBATIS (mybatis-config.xml). Éste está en la ruta donde normalmente se ubican los archivos de configuración de aplicaciones montadas en Maven (src/main/resources). Edítelos y agregue en éste, después de la sección `<settings>` los siguientes 'typeAliases':

```
<properties resource="config.properties">
</properties>
<typeAliases>
  <typeAlias type="edu.eci.cvds.samples.entities.Cliente" alias="Cliente"/>
  <typeAlias type="edu.eci.cvds.samples.entities.Item" alias="Item"/>
  <typeAlias type="edu.eci.cvds.samples.entities.ItemRentado" alias="ItemRentado"/>
  <typeAlias type="edu.eci.cvds.samples.entities.TipoItem" alias="TipoItem"/>
</typeAliases>
```