

DOCUMENTACIÓN PROYECTO CONJUNTO DE BD Y PROGRAMACIÓN (ENLAZAR CON BASE DE DATOS)

**Realizado por: Jose Antonio Vázquez Fernández
Hermenegildo Ramos Díaz
Juan Antonio Núñez Castaño**

ÍNDICE

→DESCARGA DE ARCHIVOS NECESARIOS PARA ENLAZAR LA BASE DE DATOS CON NUESTRO ECLIPSE

→PROCESOS A REALIZAR PARA EL CORRECTO ENLAZADO DENTRO DE DBEAVER

→PROCESOS A REALIZAR PARA EL CORRECTO ENLAZADO DENTRO DE ECLIPSE

→MÉTODO 1 (MOSTRAR INFORMACIÓN SOBRE LOS CLIENTES)

→RESULTADO MÉTODO 1

→MÉTODO 2 (MOSTRAR PEDIDOS)

→RESULTADO 2

→MÉTODO 3 (AÑADIR CLIENTE)

→RESULTADO 3

→MÉTODO 4 (MOSTRAR CLIENTE)

→RESULTADO 4

→MÉTODO 5 (ELIMINAR CLIENTE)

→RESULTADO 5






→MÉTODO 6 (AÑADIR PEDIDO)

→RESULTADO 6


→MÉTODO 7 (INCLUIR LÍNEAS A UN PEDIDO YA EXISTENTE)

Descargamos todos los archivos proporcionados por el profesor:

→Proyecto:

 jms3du Inserción de datos de cliente parametrizada	1ce1e02 on May 25, 2022	 3 commits
 resources	Cambio de base de datos de ejemplo	last year
 src	Inserción de datos de cliente parametrizada	last year
 pom.xml	Connecting db simple example.	last year

→Librerías necesarias:



 mysql-connector-java-8.0.32

 ojdbc8

Accedemos a nuestros IDE para programación (Eclipse) y para base de datos (Dbeaver)

DENTRO DE DBEAVER

1. Establecemos conexión con nuestra base de datos de MySQL
2. Creamos una nueva base de datos llamada tienda y ejecutamos el script proporcionado

 database_script.sql 

3. Entramos en la opción de editar conexión de nuestra base de datos de MySQL para averiguar nuestro usuario y contraseña



Authentication (Database Native)

Username:

Password: ☒ Save password locally

Advanced

DENTRO DE ECLIPSE

1. Una vez dentro de eclipse importamos el proyecto
2. Una vez el proyecto importado en nuestro Eclipse vamos a importar las dos librerías proporcionadas dentro de una carpeta que hemos creado llamada lib (en nuestro caso nos daba error al importar y la hemos copiado y pegado directamente)

lib

- mysql-connector-j-8.0.32.jar
- ojdbc8.jar

3. Vamos a la clase **db.properties** e introducimos los datos de usuario y password anteriormente obtenidos en Dbeaver

```
1 db.user=root
2 db.pass=rootpass
3 db.url=jdbc:mysql://localhost:3306/tienda
4
```

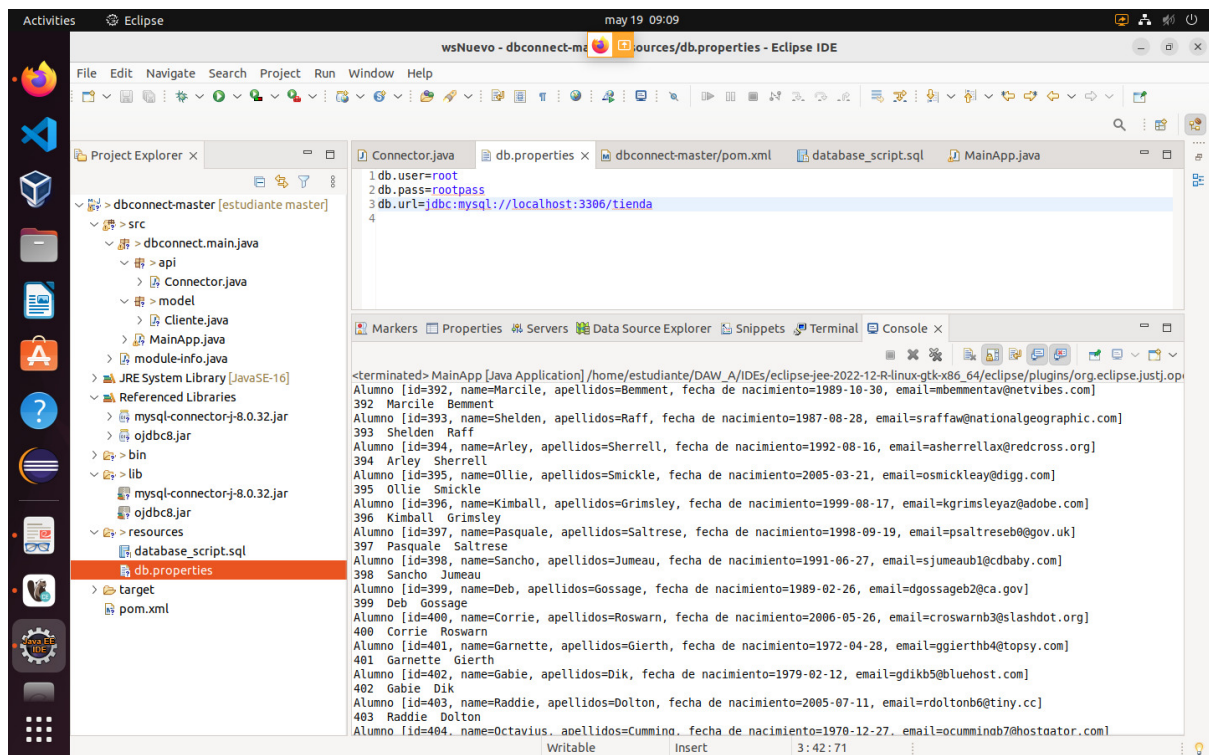
4. Modificamos el método **mostrarCliente()** e indicamos que en los parámetros del connect van a ser los datos anteriormente indicados en la clase **db.properties**

```
public void mostrarCliente() throws SQLException {
    connect=DriverManager.getConnection(JDBC_URL, USER, PASS);
    Statement statement = connect.createStatement();
    ResultSet rs=statement.executeQuery("select * from Cliente");

    while(rs.next()) { //Avanza de posición en el listado de registros y devuelve true si existe tal
        Cliente alumno = new Cliente(Integer.valueOf(rs.getString(1)), rs.getString(2), rs.getString(3),
            rs.getString(5), rs.getString(4));

        System.out.println(alumno);
        System.out.println(rs.getString(1)+" "+rs.getString(2)+" "+rs.getString(3));
    }
    connect.close();
}
```

5. Finalmente vamos a la clase **MainApp.java** y ejecutamos.



FINALMENTE YA TENDRÍAMOS ENLAZADA NUESTRA BD A ECLIPSE!!

→AHORA PASAMOS A LA SEGUNDA PARTE DEL TRABAJO CONSISTENTE EN PICAR EL CÓDIGO NECESARIO PARA EJECUTAR UN MENÚ QUE CUMPLA CON LAS SIGUIENTES FUNCIONES.

1. **Mostrar información sobre los clientes:** donde se pueda mostrar información sobre los clientes en el siguiente formato:

Nombre	Apellidos	email	Edad	Género
Merci	Morgan	mmorganro@auda.org.au	27	Mujer
Cleve	Herety	cheretyrp@smh.com.au	16	Hombre

2. **Mostrar pedidos** por importe (precio) decreciente con la siguiente información:

Codigo	Status	Nombre Completo Cliente	Nº productos	Importe
1038763517	ENVIADO	Cleve Herety	7	23,00

3. **Añadir cliente**
4. **Actualizar un cliente existente**
5. **Eliminar cliente**
6. **Añadir Pedido**
7. **Incluir Líneas** de pedido a un pedido existente en estado Procesando

→CAPTURA NECESARIA PARA EL PUNTO 1 (MOSTRAR INFORMACIÓN DEL CLIENTE)

```
public String mostrarInfoCliente() throws SQLException, ClassNotFoundException {
    StringBuilder sb = new StringBuilder();

    Connection connection = conect();

    Statement statement = statement(connection);

    ResultSet rs = statement.executeQuery("select * from Cliente");
    sb.append("nombre---apellidos---email---fechaNacimiento---genero").append(System.lineSeparator());
    while (rs.next()) { // Avanza de posición en el listado de registros y devuelve true si existe tal
        Cliente alumno = new Cliente(Integer.valueOf(rs.getString(1)), rs.getString(2), rs.getString(3),
            rs.getString(5), rs.getString(4));
        sb.append(rs.getString(2) + "---" + rs.getString(3) + "---" + rs.getString(4) + "---"
            + (LocalDate.now().getYear() - Integer.valueOf(rs.getString(5).substring(0, 4))) + "---"
            + rs.getString(6)).append(System.lineSeparator());
    }
    connection.close();

    return sb.toString();
}
```

→ RESULTADO 1

```
1
hombre---apellidos---email---fechaNacimiento---genero
Cleavland---Ricciardiello---cricciardiello@yelp.com---40---M
Jacob---Rhymes---jrhymes1@microsoft.com---37---M
Hedvige---Kenn---hkenn2@baidu.com---50---F
Virgie---Knight---vknight3@prlog.org---16---F
Ogdan---Durrant---odurrant4@rediff.com---19---M
Lucille---Mabey---lmabey5@shutterfly.com---27---F
Maximilian---Goodsal---mgoodsal6@microsoft.com---32---M
Lizette---Treske---ltreske7@hatena.ne.jp---28---F
Chen---Soltan---csoltan8@phoca.cz---38---M
Erin---Seint---eseint9@constantcontact.com---46---F
Debera---Matchett---dmatchetta@sina.com.cn---58---F
Sampson---Enderlein---senderleinb@imgur.com---50---M
Tansy---Burless---tburlessc@rambler.ru---58---F
Laure---Woodfine---lwoodfine@state.gov---26---F
Gibb---Cowper---gcowpere@alibaba.com---55---M
Donny---Wolton---dwoltonf@quantcast.com---29---F
Dylan---Neale---dnealeg@liveinternet.ru---34---M
Vendetta---Gassett---vgassett@ch1.com---30---F
```

→ CAPTURA NECESARIA PARA EL PUNTO 2 (MOSTRAR INFORMACIÓN DEL PEDIDO)

```
public String mostrarInfoPedidos() throws SQLException {
    StringBuilder sb = new StringBuilder();
    Connection connection = conect();

    Statement statement = statement(connection);

    ResultSet rs = statement.executeQuery(
        "SELECT P.id,P.status,C.nombre,C.apellido,L.cantidad,L.precio FROM Pedido P ,Linea L,Cliente C "
        + "WHERE P.idCliente = C.id AND L.idPedido = P.id");
    sb.append("codigo---status---nombreCompletoCliente---numProducto---Importe").append(System.lineSeparator());
    while (rs.next()) {
        sb.append(String.format("%s---%s---%s %s---%s---%s", rs.getString(1), rs.getString(2), rs.getString(3),
            rs.getString(4), rs.getString(5), rs.getString(6))).append(System.lineSeparator());
    }

    return sb.toString();
}
```

→ RESULTADO 2

```
codigo---status---nombreCompletoCliente---numProducto---Importe
1---CANCELADO---Adriano McCoveney---10---6
1---CANCELADO---Adriano McCoveney---6---8
1---CANCELADO---Adriano McCoveney---7---4
1---CANCELADO---Adriano McCoveney---10---10
1---CANCELADO---Adriano McCoveney---1---3
1---CANCELADO---Adriano McCoveney---9---4
1---CANCELADO---Adriano McCoveney---10---3
1---CANCELADO---Adriano McCoveney---7---3
1---CANCELADO---Adriano McCoveney---2---7
1---CANCELADO---Adriano McCoveney---10---1
1---CANCELADO---Adriano McCoveney---1---9
1---CANCELADO---Adriano McCoveney---6---5
2---PROCESANDO---Ramonda Guyon---1---8
2---PROCESANDO---Ramonda Guyon---8---2
2---PROCESANDO---Ramonda Guyon---10---2
```


→CAPTURA NECESARIA PARA EL PUNTO 3 (AÑADIR CLIENTE)

```
public void añadirCliente(String nombre, String apellido, String email, String fechaNacimiento, String genero)
    throws SQLException {
    Connection connection = conect();
    Statement statement = statement(connection);
    PreparedStatement ps = connection.prepareStatement(
        "INSERT INTO Cliente (nombre, apellido, email, fechaNacimiento, genero) VALUES (?, ?, ?, ?, ?)");
    ps.setString(1, nombre);
    ps.setString(2, apellido);
    ps.setString(3, email);
    ps.setString(4, fechaNacimiento);
    ps.setString(5, genero);
    ps.executeUpdate();
    ps.close();
    // Ejecuta query de eliminación, actualización o inserción (DELETE, UPDATE,
    // INSERT)
    // statement.executeUpdate("INSERT INTO Cliente (nombre, apellido, email,
    // fechaNacimiento, genero) "
    // + "VALUES (" + " , 'Ricciardiello', 'rr0@yelp.com', '1983-04-15', 'M');\n");
}
```

→RESULTADO 3

Dime una opcion:

3

Introduce el nombre:

josecito

Introduce el apellido:

vazquez

Introduce el email:

jv319069@gmail.com

Introduce el fechaNacimiento:

1999-12-12

Introduce el genero:

m

→CAPTURA NECESARIA PARA EL PUNTO 4 (ACTUALIZAR CLIENTE EXISTENTE)

```
public void actualizarCliente(String email, String emailAntiguo) throws SQLException {
    Connection connection = conect();
    Statement statement = statement(connection);
    PreparedStatement ps = connection.prepareStatement("UPDATE Cliente c Set c.email=? Where c.email=?");
    ps.setString(1, email);
    ps.setString(2, emailAntiguo);
    ps.executeUpdate();
    ps.close();
}
```

→RESULTADO 4

josecito---vazquez---jv319069@gmail.com---24---m

→CAPTURA NECESARIA PARA EL PUNTO 5 (ELIMINAR CLIENTE)

```
public void eliminarCliente(int id) throws SQLException {
    Connection connection = conect();
    Statement statement = statement(connection);
    int idPedido = 0;
    ResultSet rs = statement.executeQuery("SELECT id from Pedido p where p.idCliente = " + id);
    while (rs.next()) {
        idPedido = rs.getInt(1);
        PreparedStatement ps = connection.prepareStatement("DELETE FROM Linea l WHERE l.idPedido = ? ");
        ps.setInt(1, idPedido);
        ps.executeUpdate();
        ps.close();
    }
    PreparedStatement ps1 = connection.prepareStatement("DELETE FROM Pedido p WHERE p.idCliente = ? ");
    ps1.setInt(1, id);
    ps1.executeUpdate();
    ps1.close();
    PreparedStatement ps2 = connection.prepareStatement("DELETE FROM Cliente c WHERE c.id = ?");
    ps2.setInt(1, id);
    ps2.executeUpdate();
    ps2.close();
}
```

→RESULTADO 5

eliminado = true;

→CAPTURA NECESARIA PARA EL PUNTO 6 (AÑADIR PEDIDO)

```
public void anadirPedido(Integer id, String codigo, String status, Integer idCliente) throws SQLException {
    Connection connection = conect();
    Statement statement = statement(connection);
    PreparedStatement ps = connection
        .prepareStatement("INSERT INTO Pedido (id, codigo, status, idCliente) VALUES (?, ?, ?, ?)");
    ps.setInt(1, id);
    ps.setString(2, codigo);
    ps.setString(3, status);
    ps.setInt(4, idCliente);
    ps.executeUpdate();
    ps.close();

    // Ejecuta query de eliminación, actualización o inserción (DELETE, UPDATE,
    // INSERT)
    // statement.executeUpdate("INSERT INTO Cliente (nombre, apellido, email,
    // fechaNacimiento, genero) "
    // + "VALUES (" + " , 'Ricciardiello', 'rr0@yelp.com', '1983-04-15', 'M');\n");
}
```

→RESULTADO 6

```
Dime una opcion:
6
Dime el id:
1000
Dime el codigo
2
Dime el estado
PROCESANDO
Dime el idCliente
1
```


→ CAPTURA NECESARIA PARA EL PUNTO 7 (INCLUIR LÍNEAS A UN PEDIDO EXISTENTE)

```
public void anadirLinea(int id, String codigo, String nombreProducto, int idPedido, int cantidad, int precio)
    throws SQLException {
    Connection connection = conect();
    Statement statement = statement(connection);
    PreparedStatement ps = connection.prepareStatement(
        "INSERT INTO Linea (id, codigo, nombreProducto, idPedido, cantidad, precio) VALUES (?, ?, ?, ?,?,?)"
    );
    ps.setInt(1, id);
    ps.setString(2, codigo);
    ps.setString(3, nombreProducto);
    ps.setInt(4, idPedido);
    ps.setInt(5, cantidad);
    ps.setInt(6, precio);
    ps.executeUpdate();
    ps.close();
}
```