

Álvaro García Fuentes  
Horas Libre Configuración  
Desarrollo de Aplicaciones Web

## **UD2 1 Instalación y configuración de JDBC**

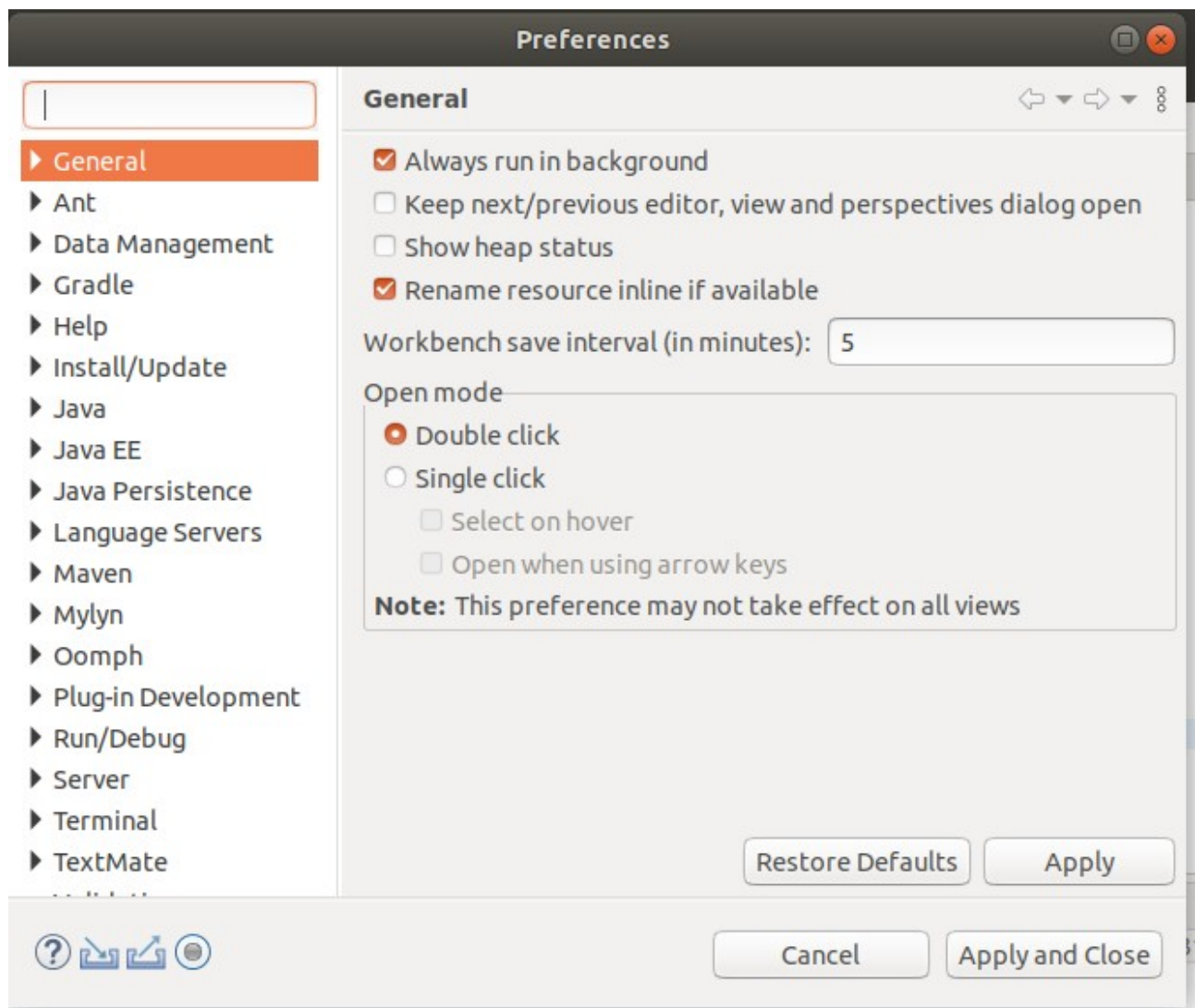
## 1.Instala el sistema de gestión de bases de datos (DBMS) MySQL.

Accedemos a la página de MySQL para descargar su software de gestión de bases de datos. Como mi S.O. es Ubuntu elijo la opción de descargar el paquete .deb para instalarlo.

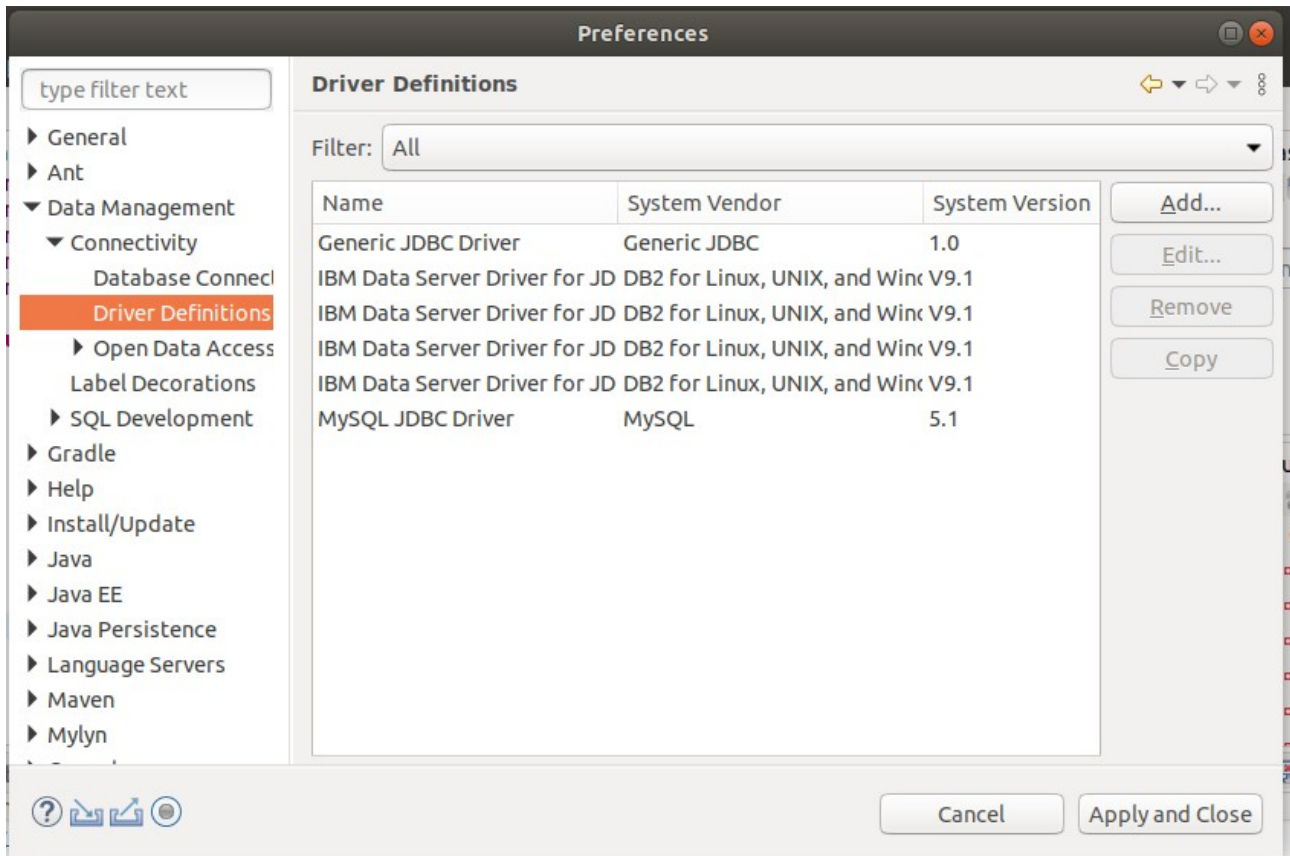
## 2. Instala el driver en el proyecto del IDE Eclipse.

Primero descargamos el driver de la página de MySQL, después lo alojamos en una carpeta de nuestra elección, y luego pasamos a configurar Eclipse para que trabaje con él:

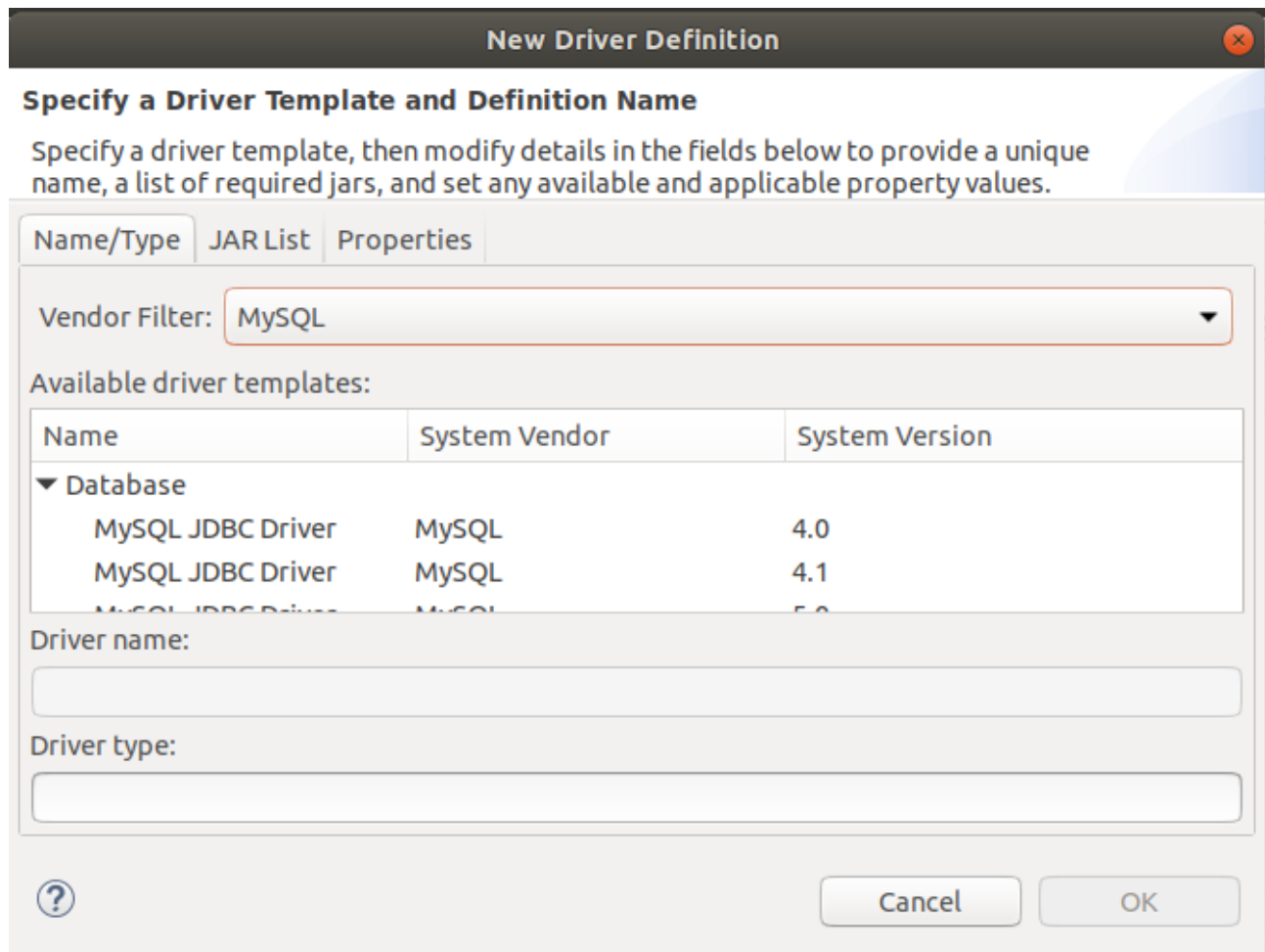
Abrimos Eclipse y vamos a la ruta Window → Preferences, ahí nos saldrá la siguiente ventana:



Aquí, iremos a la opción Data Management → Conectivity → Driver Definitions



Pulsamos el botón Add..., en la ventana que aparece seleccionamos como Vendor Filter la opción MySQL y en la lista seleccionamos la versión más reciente disponible.



The image shows a 'New Driver Definition' dialog box with a dark title bar. It has three tabs: 'Name/Type', 'JAR List', and 'Properties'. The 'Name/Type' tab is active. Inside, there's a 'Vendor Filter' dropdown set to 'MySQL'. Below it, a section titled 'Available driver templates:' contains a table with columns 'Name', 'System Vendor', and 'System Version'. The table lists MySQL JDBC drivers. Below the table are input fields for 'Driver name:' and 'Driver type:'. At the bottom are a help icon, a 'Cancel' button, and an 'OK' button.

**New Driver Definition**

**Specify a Driver Template and Definition Name**

Specify a driver template, then modify details in the fields below to provide a unique name, a list of required jars, and set any available and applicable property values.

Vendor Filter:

Available driver templates:

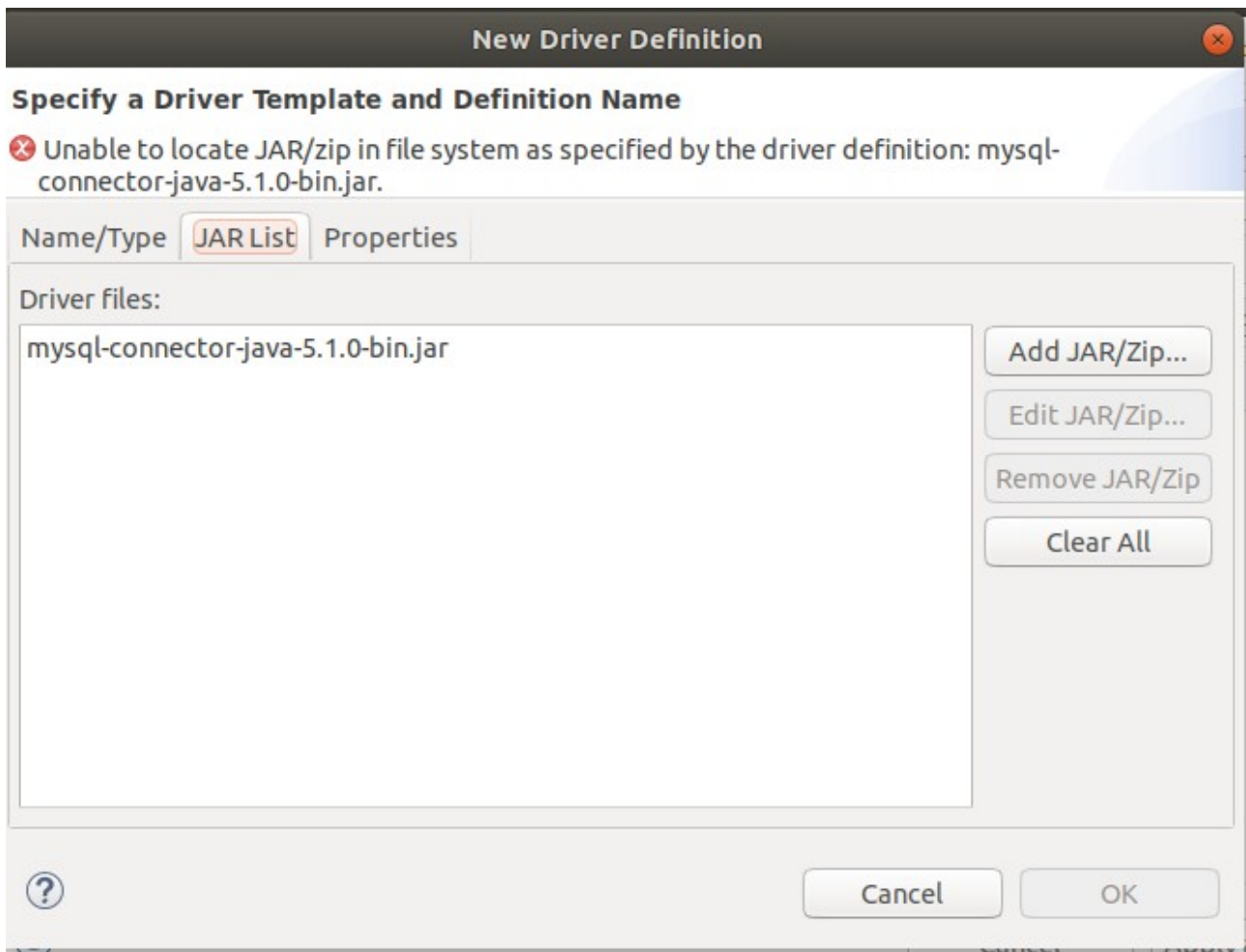
Name	System Vendor	System Version
▼ Database		
MySQL JDBC Driver	MySQL	4.0
MySQL JDBC Driver	MySQL	4.1
MySQL JDBC Driver	MySQL	5.0

Driver name:

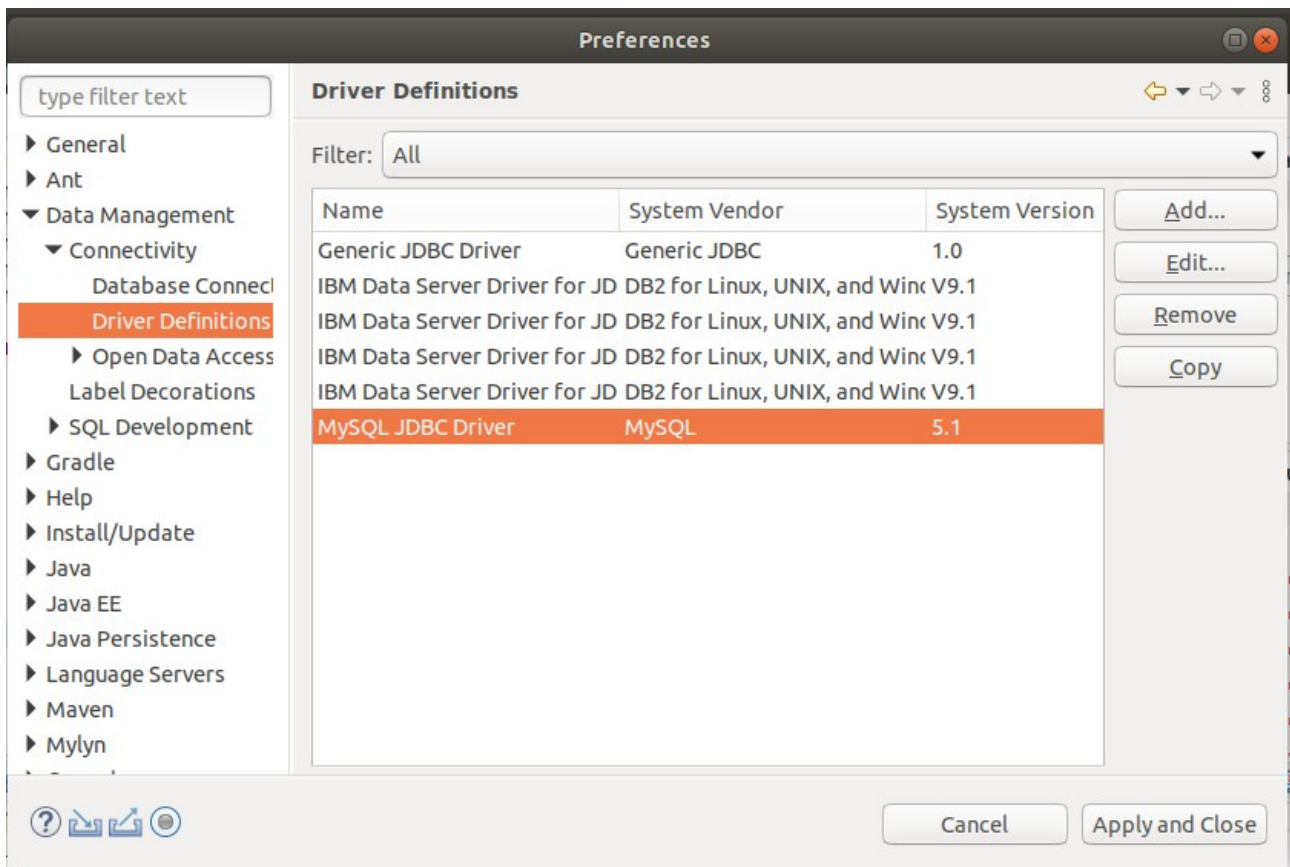
Driver type:

Ahora, seleccionamos la pestaña Jar List, y ahí seleccionamos add JAR/Zip, ahí nos pedirá que seleccionemos la ruta de nuestro driver descargado:

Álvaro García Fuentes  
Horas Libre Configuración  
Desarrollo de Aplicaciones Web



Le damos a OK y luego en la ventana principal de Preferences le damos a Apply and Close:

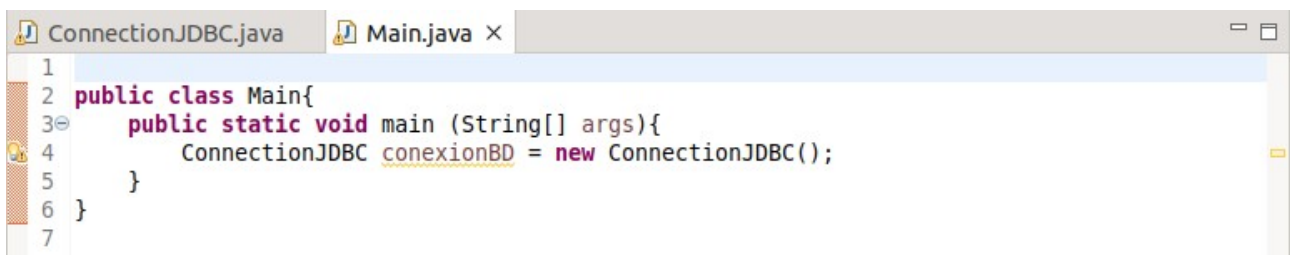


Con esto ya tenemos terminada nuestra configuración de JDBC y podemos empezar a utilizarlo.

### 3. Partiendo de `DriverManager.getConnection()` crea una conexión (objeto `Connection`) que represente la conexión establecida entre la aplicación y la BD.

Mediante Eclipse creamos un proyecto Java para conectar a nuestra Base de Datos con JDBC. Creamos el siguiente código:

Achivo main:



En este archivo se crea un objeto de clase `ConnectionJDBC`, la cual definimos en el siguiente archivo:



```
ConnectionJDBC.java x Main.java
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4 import java.sql.SQLTimeoutException;
5 import java.sql.Statement;
6
7 public class ConnectionJDBC {
8
9     private static String db_ = "HLC";
10    private static String login_ = "alvaro";
11    private static String password_ = "biballo1";
12    private static String url_ = "jdbc:mysql://127.0.0.7:3306/" + db_;
13    private static Connection connection_ = null;
14    private static Statement st_ = null;
15
16    public ConnectionJDBC(){
17        try {
18            Class.forName("com.mysql.cj.jdbc.Driver");
19            connection_ = DriverManager.getConnection(url_, login_, password_);
20
21            if( connection_ != null ) {
22                st_ = connection_.createStatement();
23                System.out.println("Conexion a base de datos " + db_ + " correcta.");
24            } else
25                System.out.println("Conexion fallida.");
26        } catch ( SQLTimeoutException e ) { System.out.print("Error de tiempo de conexion."); }
27        catch ( SQLException e ) { e.printStackTrace(); }
28        catch ( ClassNotFoundException e ) { e.printStackTrace(); }
29        catch ( Exception e ) { e.printStackTrace(); }
30    }
31 }
```

En esta práctica nuestro código no hace nada excepto establecer una conexión con la Base de Datos MySQL llamada HLC, con un login y contraseña establecidos en la misma.

#### 4. Confirma que todo ha ido bien. Tendrás que capturar las excepciones SQLException y SQLTimeoutException.

Como nuestro código no hace nada más que conectarse a la Base de Datos sin hacer ninguna otra operación, nos lanza una excepción debido a que la Base de Datos no envía ningún paquete. Sin embargo, se nos confirma que se ha realizado dicha conexión.

```
com.mysql.cj.jdbc.exceptions.CommunicationsException: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.exceptions.SQLError.createCommunicationsException(SQLError.java:174)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:93)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:828)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:448)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:241)
at mysql.connector.java@8.0.26/com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:198)
at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:677)
at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:228)
at ConnectionJDBC.<init>(ConnectionJDBC.java:19)
at Main.main(Main.java:4)
Caused by: com.mysql.cj.exceptions.CJCommunicationsException: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.
at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
at java.base/jdk.internal.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
at java.base/java.lang.reflect.Constructor.newInstance(Constructor.java:490)
at mysql.connector.java@8.0.26/com.mysql.cj.exceptions.ExceptionFactory.createException(ExceptionFactory.java:61)
at mysql.connector.java@8.0.26/com.mysql.cj.exceptions.ExceptionFactory.createException(ExceptionFactory.java:105)
at mysql.connector.java@8.0.26/com.mysql.cj.exceptions.ExceptionFactory.createException(ExceptionFactory.java:151)
at mysql.connector.java@8.0.26/com.mysql.cj.exceptions.ExceptionFactory.createCommunicationsException(ExceptionFactory.java:164)
at mysql.connector.java@8.0.26/com.mysql.cj.protocol.a.NativeSocketConnection.connect(NativeSocketConnection.java:89)
```