



Conexión a base de Datos JDBC

Realizado por:

José Miguel Cabrera Sampedro,
Ana López Atero,
Iván Mejías Caro.

Curso: 1º DAW A

Módulo: Programación.

Profesor: José Manuel Sevillano.

Fecha: 19/05/2023



¿Qué es JDBC?

Son las siglas de *Java Data Base Connectivity*, un conjunto de herramientas que permiten la conexión desde Java a una base de datos permitiendo su manipulación.

Se trabaja con los paquetes **Java.sql** y **Javax.sql**. La manipulación de datos, se realiza a través de su clase principal *DriverManager* que implementa las interfaces *ResultSet*, *Connection*, *Statement* y *DataSource*.

La conexión a la base de datos se realiza a través de un *driver* proporcionado por el fabricante.

El proceso para un acceso a BBDD

1. Establecer conexión con la BBDD.

Añadimos a nuestra librería del proyecto el *driver* de MySQL.

<https://repo1.maven.org/maven2/com/mysql/mysql-connector-j/8.0.32/mysql-connector-j-8.0.32.jar>

Se crea un objeto de tipo *Connection* utilizando el *driver* del fabricante, en nuestro caso *MySQL*.

Hemos utilizado cadena de conexión: “**jdbc:mysql://localhost:3306/tienda**”, “**root**”, “**rootpass**”.

2. Crear un objeto *Statement* y *PreparedStatement*.

La clase *Statement* y *PreparedStatement* contienen métodos que nos permiten manipular sentencias SQL.

Para crear un objeto *PreparedStatement* utilizamos la conexión realizada en la clase *Connection* y llamamos al método *prepareStatement* (Sentencia SQL en forma de *String*).

3. Ejecutar sentencia SQL

Para crear el objeto *Statement* utilizamos el objeto creado de tipo *Connection* y hacemos uso del método *createStatement()* que devuelve un tipo *Statement*. Con este objeto *Statement* hacemos la ejecución final de la sentencia. Utilizamos:

- *executeQuery* cuando queremos ejecutar consultas.
- *executeUpdate* cuando son sentencias DML.

Si ejecutamos una sentencia de consulta, esta devuelve un objeto de tipo *resultSet*, que almacena las filas obtenidas, para posteriormente interactuar con ella (recorrer los datos, almacenar en variables, mostrar, etc.).



4. Leer resultSet (Tabla virtual)

Para leer un *resultSet*, hacemos uso en nuestro objeto *resultSet* de los métodos, *getString()* y *next()*.