

# Uso y funcionamiento de un driver JDBC

**DigitalHouse**>  
Coding School

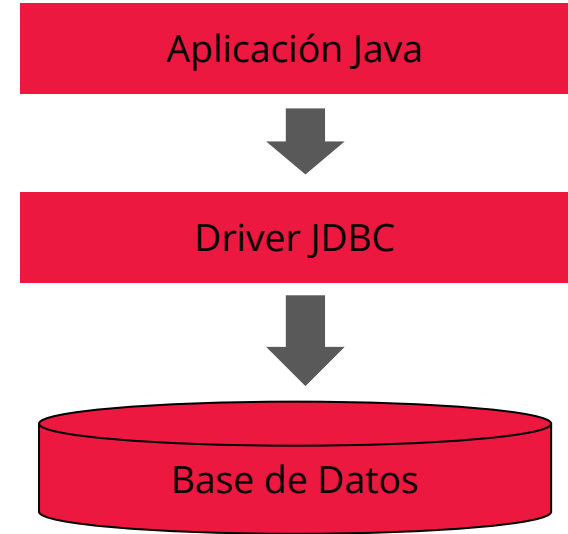


**Certified Tech  
Developer**  
The Ultimate Degree

# Drivers

Si bien cualquiera puede implementar las interfaces que JDBC provee, puede ser un trabajo arduo y extenso, ya que requiere de un gran conocimiento sobre motores de bases de datos. Por lo tanto, hay una implementación de JDBC.

Cada implementación del framework consiste en un paquete de clases, típicamente empaquetados en archivos *.jar*. Se pueden descargar de los sitios oficiales de cada motor de base de datos y son conocidos con el nombre de *driver jdbc*. Es decir que tenemos un *driver jdbc* para Oracle, uno para MySQL, etc.



# Drivers

## Uso y funcionamiento

Cada driver es un paquete de clases. Como tal, debemos incorporarlo en forma de librería a nuestro programa java. De esta forma, tendremos accesibles en el CLASSPATH, las clases que implementan las interfaces del framework. Si bien es posible hacerlo manualmente, típicamente hacemos esto mediante el IDE.

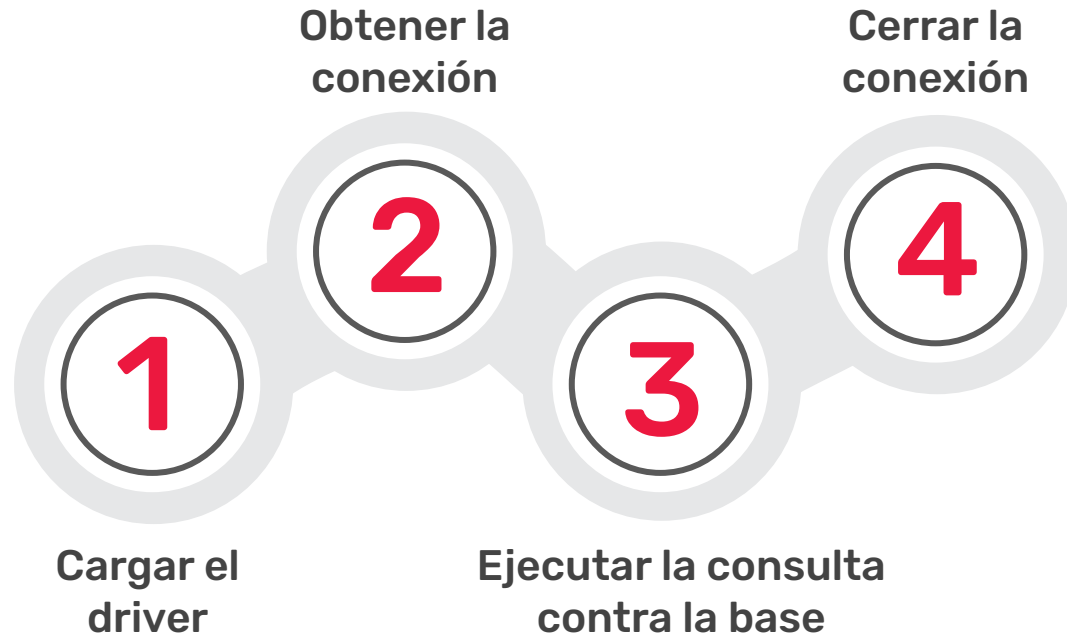
# Drivers

## Uso y funcionamiento

Una vez disponible el driver en el CLASSPATH podremos empezar a trabajar con el driver. En su expresión más básica, cada vez que queramos interactuar con la base de datos necesitamos realizar una serie de pasos. Conozcamos cuáles son...

# Drivers

## Pasos



Veamos, en código, cómo se llevan a cabo estos pasos. Supongamos una clase de prueba, con un método main en el que ponemos:

```
Connection c = null;
try {
    Class.forName("org.h2.Driver");
} catch (ClassNotFoundException e) {
    e.printStackTrace();
    System.exit(0);
}
```

```
try {  
    String url =  
"jdbc:h2:tcp://localhost//D:/base_de_datos/ejemplo";  
    c = DriverManager.getConnection(url, "sa", "sa");  
    c.setAutoCommit(false);  
} catch (SQLException e) {  
    e.printStackTrace();  
    System.exit(0);  
}
```

```
try {  
    Statement s = c.createStatement();  
    s.execute("AQUÍ_SENTENCIA_SQL");  
} catch (Exception e) {  
    e.printStackTrace();  
} finally {  
    try {  
        c.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```



DigitalHouse>  
Coding School