

**1.** En el método main, conecta con la base de datos de tres formas diferentes, es decir, construyendo el objeto Connection de tres maneras distintas. Muestra un mensaje confirmando si la conexión fue exitosa.

Parámetros de conexión:

```
String user = "Nacho";
String password = "123456";
String url = "jdbc:postgresql://localhost:5432/InstitutoFP";
```

Código:

```
//Connection 1
try (final Connection connection =
    DriverManager.getConnection(url, user, password)){

    System.out.println(connection.getCatalog() + ", conexión 1");

} catch (SQLException e){
    System.out.println("SQLException: " + e.getMessage());
    System.out.println("SQLState: " + e.getSQLState());
    System.out.println("VendorError: " + e.getErrorCode());
}

//Connection 2
try (final Connection connection2 =
    DriverManager.getConnection(url, user, password)){

    System.out.println(connection2.getCatalog() + ", conexión 2");

} catch (SQLException e){
    System.out.println("SQLException: " + e.getMessage());
    System.out.println("SQLState: " + e.getSQLState());
    System.out.println("VendorError: " + e.getErrorCode());
}

//Connection 3
try (final Connection connection3 =
    DriverManager.getConnection(url, user, password)){

    System.out.println(connection3.getCatalog() + ", conexión 3");

} catch (SQLException e){
    System.out.println("SQLException: " + e.getMessage());
    System.out.println("SQLState: " + e.getSQLState());
    System.out.println("VendorError: " + e.getErrorCode());
}
```

Ejecución:

```
C:\Users\shote\.jdk\valhalla
InstitutoFP, conexión 1
InstitutoFP, conexión 2
InstitutoFP, conexión 3
```

**2.** Si quisiéramos conectar con una base de datos Oracle, ¿qué pasos deberíamos cambiar? Añade código también.

Código:

```
Connection con = null;

try {

    String urlOracle = "jdbc:oracle:thin:@localhost:1521:XE";
    String userOracle = "usuario";
    String passwordOracle = "clave";

    con = DriverManager.getConnection(url, user, password);
    System.out.println("Conexión a Oracle exitosa: " + con.getCatalog());

} catch (SQLException e) {

    System.out.println("Error al conectar: " + e);

}
```

Ejecución:

```
Conexión a Oracle exitosa: InstitutoFP
```

**3.** Si buscas en internet, muchas fuentes utilizan la siguiente sentencia para conectar con la BBDD, `Class.forName("org.postgresql.Driver");` ¿Por qué crees que se utiliza? ¿Por qué nosotros no lo hemos usado en nuestro código?

Antes de JDBC 4 (Java 6), era obligatorio cargar manualmente el driver para registrarlo en `DriverManager`. Esto hace que la clase del driver se ejecute y se registre sola.

Desde JDBC 4 (Java 6 en adelante), los drivers incluyen "Service Provider Mechanism", lo que permite que se auto registren al estar en el classpath. Es decir, Java ya lo hace automáticamente, No es necesario llamar a `Class.forName()` en la mayoría de casos actuales.

**4. Además de user y password, ¿qué más parámetros podemos pasar en la url de conexión? String connectionUrl = "jdbc:postgresql://localhost:5432/InstitutoFP?" + "user=usuario&password=123456";**

Podríamos pasarle los siguientes parámetros extra:

ssl=true → Activar cifrado SSL.

currentSchema=nombre → Seleccionar el esquema por defecto.

serverTimezone=UTC → Establecer zona horaria