



Java

Tema 8

Conexión a Base de Datos

JDBC

1



Contenido

- 1 - Introducción a JDBC**
- 2 - Conectando a la base de datos con JDBC**
- 3 - Excepciones SQL**



Introducción a JDBC

3



Introducción a JDBC

- ❑ El API JDBC (Java DataBase Connectivity) es un conjunto de interfaces y clases Java para la **conexión y acceso a cualquier base de datos**.
- ❑ Incluido en los paquetes `java.sql` y `javax.sql`.
- ❑ Los SGBD comerciales proporcionarán un **fichero JAR** con el Driver JDBC, es decir las implementaciones de las interfaces del API.
- ❑ Ejemplo de driver JDBC de MySQL (**se ha de añadir a nuestro proyecto en Libraries**):

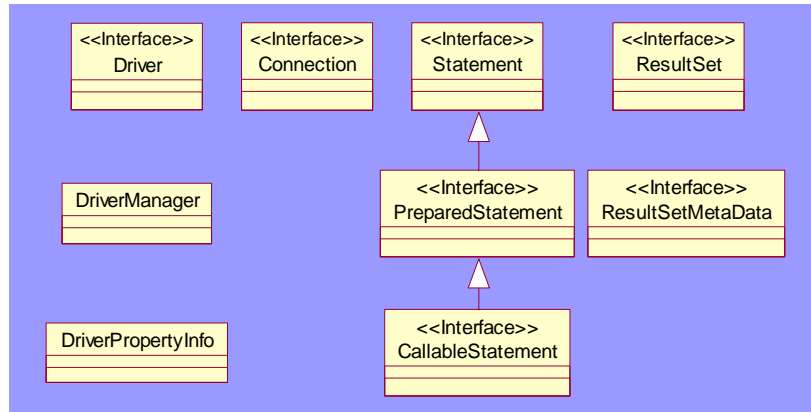
MySQL: `com.mysql.jdbc.Driver`

4



Introducción a JDBC

□ Algunas clases e interfaces del API JDBC :



5



Conectando a la Base de datos con JDBC

6



Pasos para crear una aplicación JDBC

- 1) Conectarse a la BD usando un objeto **Connection**
- 2) Ejecutar sentencias SQL usando un objeto **PreparedStatement** (ó Statement sólo si no hay parámetros)
- 3) Si es necesario, procesar registros usando un objeto **ResultSet**
- 4) Liberar recursos cerrando los objetos **Connection** y **PreparedStatement** → Si usamos **try(...)** se cerrarán automáticamente

7



Conectarse a la base de datos: La clase DriverManager

- El método estático **getConnection()** de la clase **DriverManager** establece una conexión entre una base de datos y su driver JDBC. Devuelve un objeto **Connection** que representa una conexión a una base de datos:

```
Connection conexión =  
    DriverManager.getConnection("URL", "user", "password");
```

- ✓ URL (uniform resource locator / localizador uniforme de recursos) identifica el driver, el servidor de la base de datos, el puerto y cadena de conexión de la bd
- ✓ user y password identifican el usuario y contraseña

8



Conectarse a la base de datos: DriverManager.getConnection()

// Ejemplo de conexión a MySQL

```
try {  
    Connection conexion = DriverManager.getConnection  
    ("jdbc:mysql://localhost:3306/parques","root","root");  
    driver MySQL  servidor  puerto  SID de BD  usuario  contraseña  
    └──────────────────────────────────┘  
                                URL  
  
} catch (SQLException e) {  
    System.out.println(e.getMessage());  
}
```

9



Excepciones SQL

10



Excepciones SQL

❑ `SQLException`: proporciona la siguiente información sobre el error:

- ✓ El método `getMessage()` devuelve un String describiendo el error generado por la BD
- ✓ El método `getErrorCode()` devuelve un entero con el código de error de la BD

11