



POLITÉCNICA
"Ingeniamos el futuro"

CAMPUS
DE EXCELENCIA
INTERNACIONAL

Universidad Politécnica de Madrid
ETS de Ingenieros Informáticos



BASES DE DATOS

Ingeniería Informática

Matemáticas e Informática

BASES DE DATOS

Acceso a Bases de datos en Java

BASES DE DATOS

Contacto con Prof. Alejandro Rodríguez

Email: alejandro.rg@upm.es

Conectores

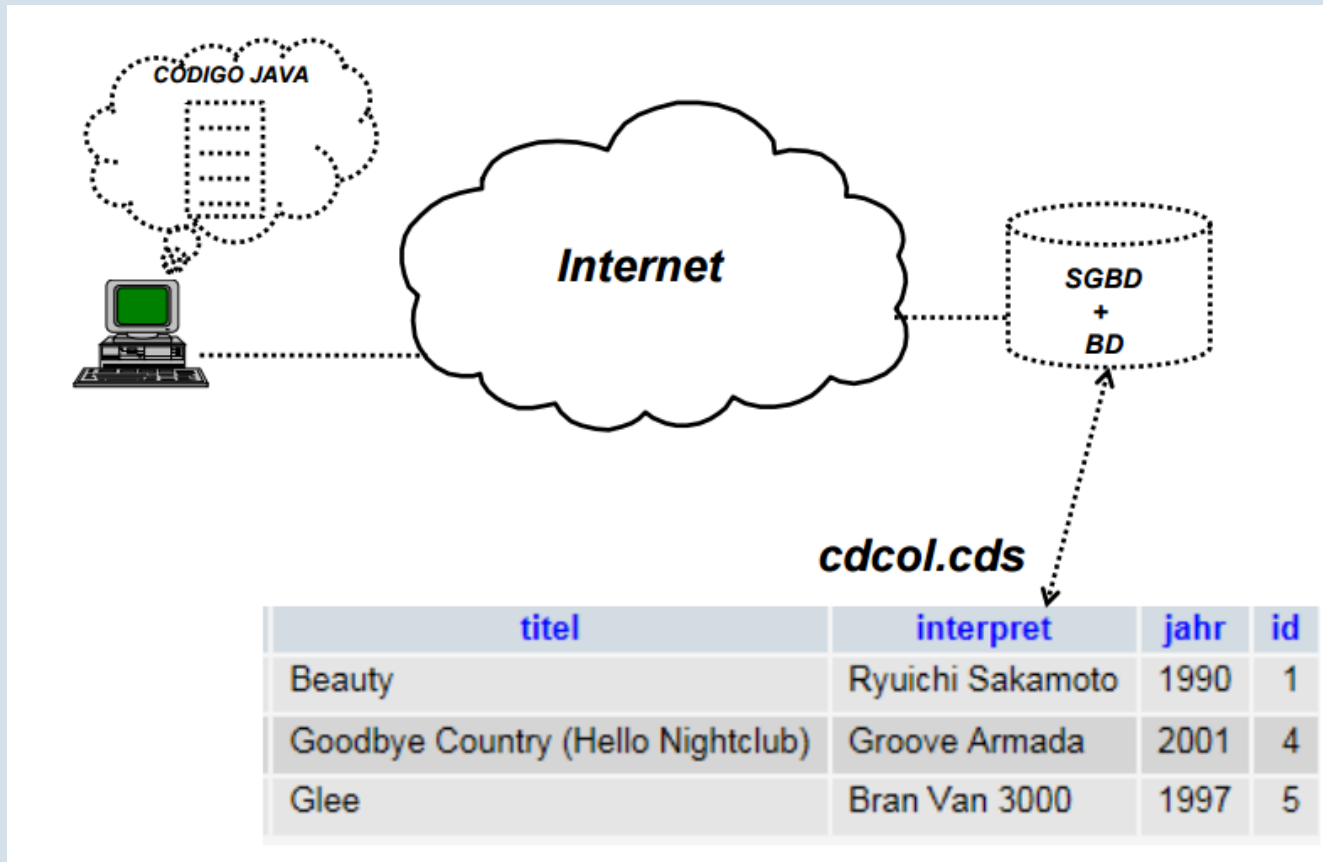
Dependiendo del lenguaje de programación tendremos una serie de conectores disponibles para trabajar con bases de datos.

JDBC

Java Database Connectivity

JDBC es un API que permite ejecutar operaciones sobre bases de datos en Java.

JDBC



JDBC

Java Database Connectivity

Como cualquier programa Java es independiente del SO o de la base de datos a la que se accede (driver-dependent).

JDBC

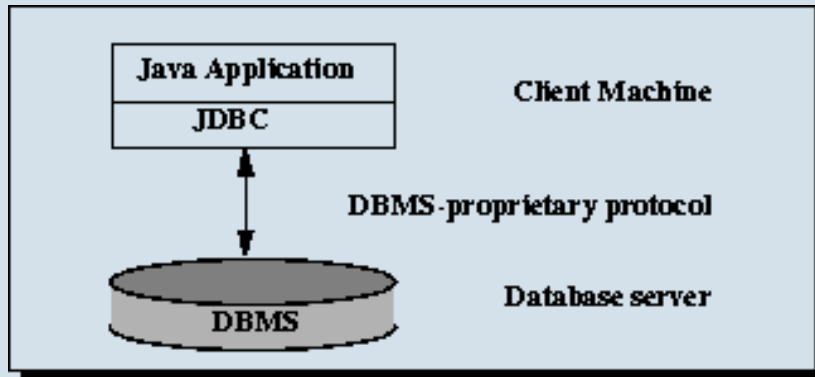
Java Database Connectivity

Como cualquier programa Java es independiente del SO o de la base de datos a la que se accede (driver-dependent).

Utiliza SQL como lenguaje de acceso y manipulación.

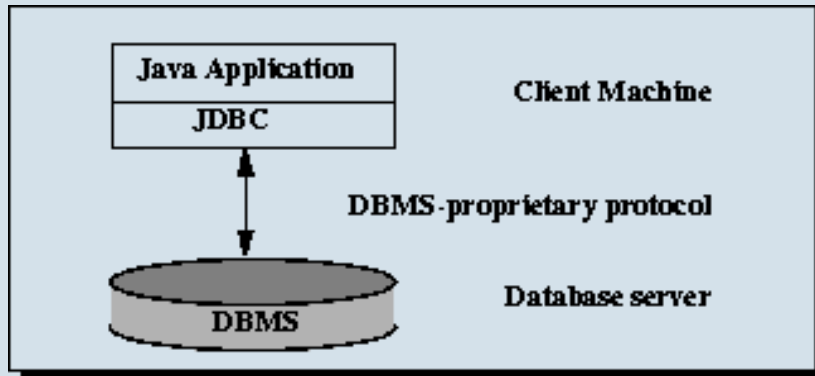
JDBC

Arquitectura 2 capas

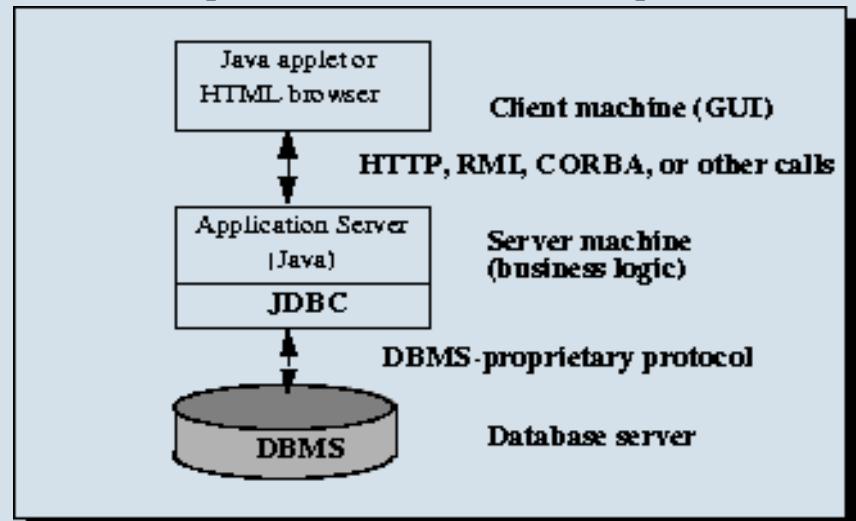


JDBC

Arquitectura 2 capas



Arquitectura 3 capas



JDBC

La conexión a un DBMS en particular se realiza junto con el driver/biblioteca de conexión apropiado.

JDBC

La mayoría de los DBMS tienen su propia implementación de driver JDBC:

- Oracle: <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>
- MySQL: <https://dev.mysql.com/downloads/connector/j/5.1.html>
- PostgreSQL: <https://jdbc.postgresql.org/>
- SQL Server: <https://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx>

JDBC

Componentes básicos de JDBC:

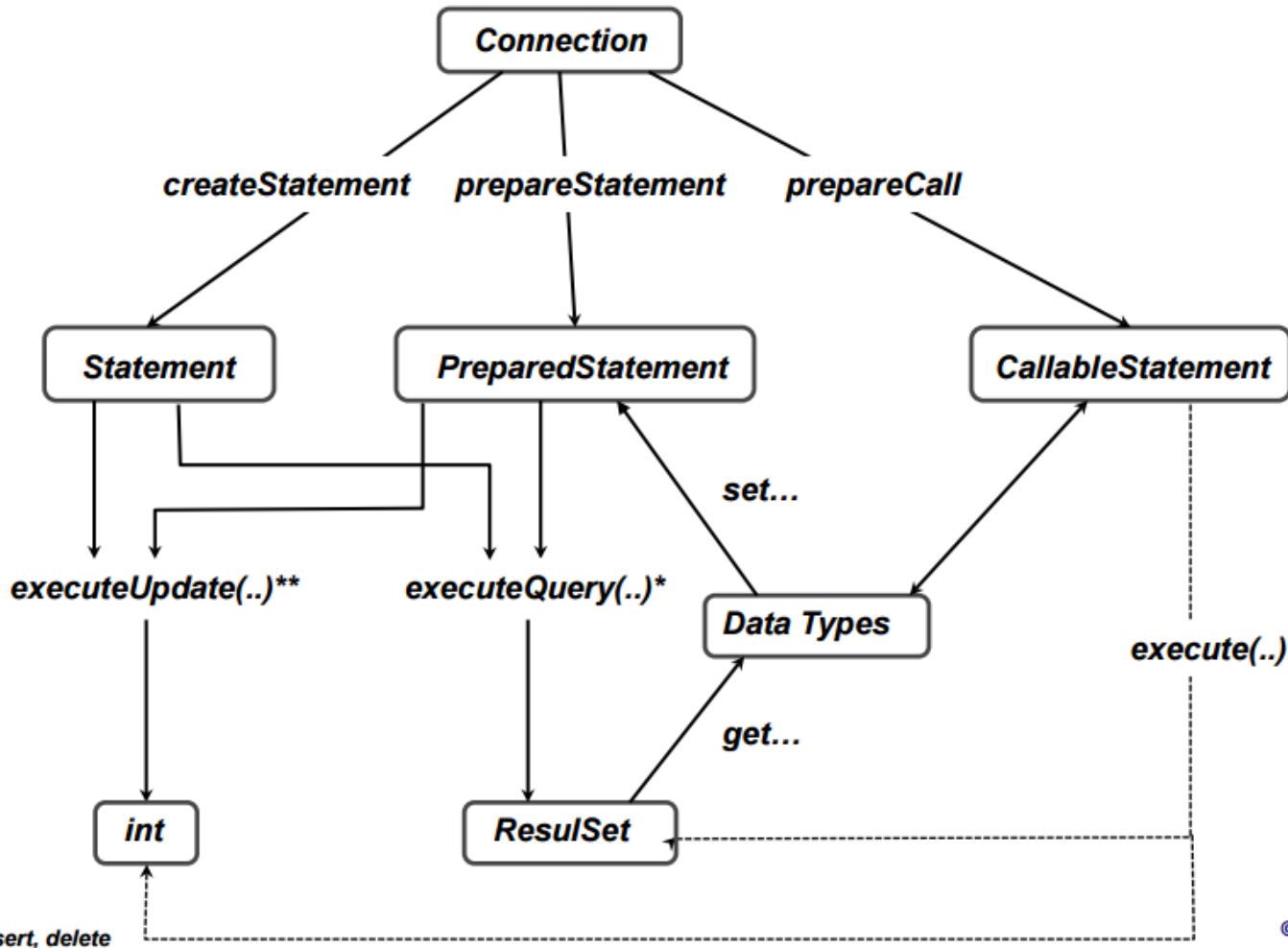
API: Proporciona el acceso programático a través de determinadas clases y métodos. Puede ejecutar sentencias SQL, obtener resultados y propagar cambios. El API JDBC es parte de la plataforma Java (SE, EE). Paquetes: `java.sql` y `javax.sql` (la mayoría de lo que se usa viene de `java.sql`)

Clases e Interfaces JDBC

Componentes básicos de JDBC:

<i>Clase</i>	<i>Descripción</i>
<i>DriverManager</i> <i>Connection</i> <i>Statement</i> <i>ResultSet</i>	<i>Para cargar un driver</i> <i>Para establecer conexiones con las bases de datos</i> <i>Para ejecutar sentencias SQL y enviarlas a las BBDD</i> <i>Para almacenar el resultado de la consulta</i>

Clases e Interfaces JDBC



Tipos de datos SQL/Java

Tipo de dato SQL

INTEGER o INT
SMALLINT
FLOAT
DOUBLE
CHARACTER(n) o CHAR(n)
VARCHAR(n)
BOOLEAN
DATE
TIME
TIMESTAMP
BLOB

Tipo de dato Java

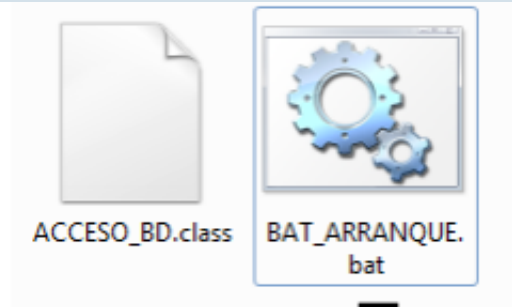
int
short
float
double
String
String
boolean
java.sql.Date
java.sql.Time
java.sql.Timestamp
java.sql.Blob

Incorporar driver JDBC en proyecto de consola

El fichero físico que contiene el driver (JAR/ZIP) debe estar en el CLASSPATH de la app.

El ejemplo siguiente es con MySQL como SGBD, pero podría ser otro.

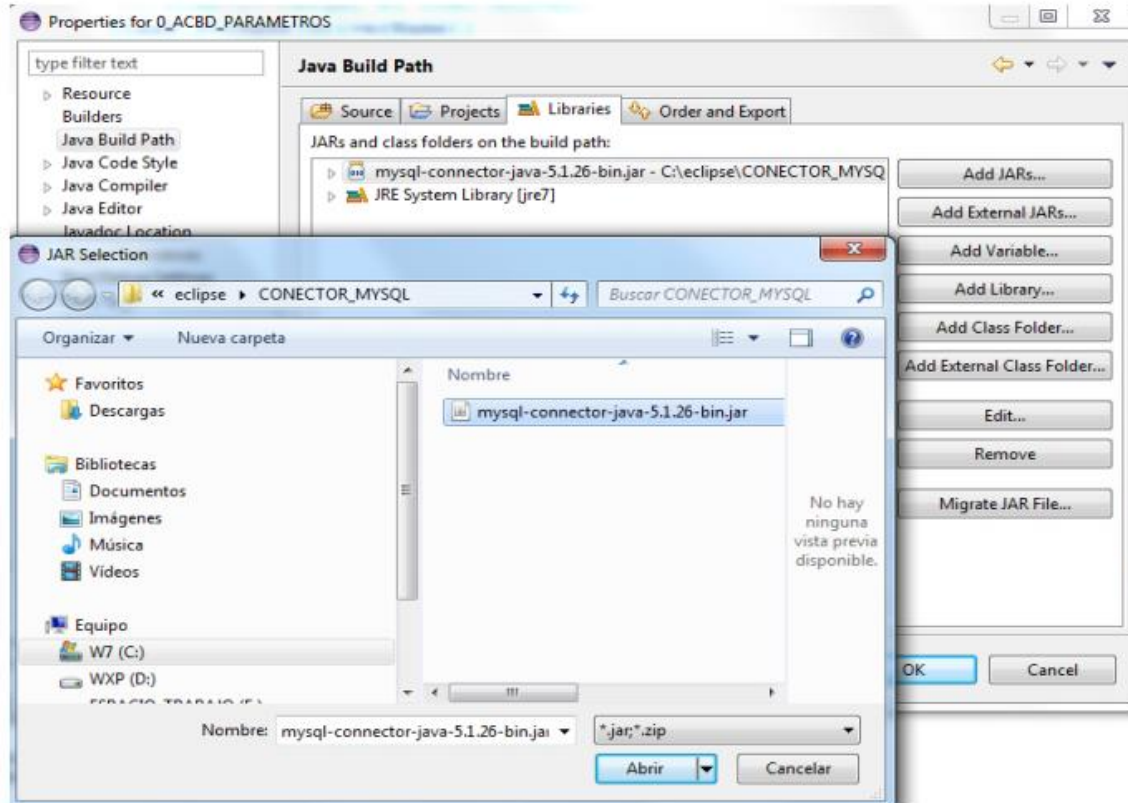
Incorporar driver JDBC en proyecto de consola



```
set CLASSPATH=%CLASSPATH%;  
C:\eclipse\CONNECTOR_MYSQL\mysql-connector-java-5.1.26-bin.jar  
  
java ACCESO_BD
```

Incorporar driver JDBC en proyecto de Eclipse

Project -> Properties -> Java Build Path -> Libraries
Click Add External JARs -> Select the JDBC driver. -> Abrir



Instalación de Driver

En el siguiente vídeo se muestra el proceso de instalación de driver y prueba:

<https://www.youtube.com/watch?v=foqBso6MHX8>

Fases de comunicación con JDBC

1. Cargar el driver
2. Establecer conexión con la BBDD (clase Connection)
3. Crear y ejecutar una sentencia SQL (clases Statement/PreparedStatement, métodos executeQuery/executeUpdate)
4. Gestionar el resultado (clase ResultSet)
5. Liberar recursos

Fase 1

```
public EJ1() {
    try {
        init();
    } catch (ClassNotFoundException e) {
        System.err.println("Clase no encontrada. ¿Se ha cargado el driver en el proyecto?");
        System.err.println("URL: http://dev.mysql.com/downloads/connector/j/");
    }
}

private void init() throws ClassNotFoundException {
    String drv = "com.mysql.jdbc.Driver";
    Class.forName(drv);
    System.out.println("Driver encontrado!");
}
```

Simplemente carga el driver. Si no funciona es porque falta cargar el JAR de JDBC.

Fase 1: Cargar el driver JDBC



Ver ejemplo – EJ1

Fase 2

```
private void init() throws Exception {  
    String drv = "com.mysql.jdbc.Driver";  
    Class.forName(drv);  
  
    String serverAddress = "localhost:3306";  
    String db = "sakila";  
    String user = "bd";  
    String pass = "bdupm";  
    String url = "jdbc:mysql://" + serverAddress + "/" + db;  
    Connection conn = DriverManager.getConnection(url, user, pass);  
    System.out.println("Conectado a la base de datos!");  
}
```

Sin modificaciones de la BD previas este código dara error por el tema del usuario y la password e incluso si no existe la BD.