13 EJERCICIOS

1. Rellena la siguiente tabla en donde a la izquierda aparece el nombre de la base de datos y a la derecha debe aparecer la librería necesaria para la conexión mediante un programa Java:

Base de datos Librería Java necesaria para la conexión

SQLite sqlite-jdbc-3.7.2.jar

Apache Derby derbytools.jar

HSQLDB hsqldb.jar

H2 h2.jar

MySQL ORACLE Db4o Berkeley db4o.jar

2. Rellena la siguiente tabla resumen para que aparezca por cada base de datos estudiada el driver y la URL necesaria para el establecimiento de la conexión:

|  |  |  |
| --- | --- | --- |
| **Base de datos** | **Driver** | **URL** |
| SQLite  Apache Derby  HSQLDB  H2  MySQL ORACLE db4o | org.sqliteJDBC ..................  org.derbyJDBC  org.hsqldb  org.h2JDBC  org.db4oJDBC | jdbc:sqlite;D.:/db/SQLite/ejemplo,db jdbc:derby;/localhost  jdbc:hsqldb;ejemplo  jdbc:h2;D.:/db/SQLite/ejemplo,db  jdbc:db4o;D.:/db/SQLite/ejemplo,yap |

3. Cuál de las siguientes afirmaciones sobre JDBC NO es correcta:

a) JDBC define una API que pueden usar los programas Java para conectarse a bases de datos relaciónales y orientadas a objetos.

b) JDBC no solo provee una interfaz sino que también define una arquitectura estándar, para que los fabricantes puedan crear los drivers que permitan a las aplicaciones Java el acceso a los datos.

c) JDBC dispone de la misma interfaz para todas las bases de datos.

**d) Los tipos de conectores 3 y 4 se usan normalmente cuando el único sistema de acceso final al gestor de bases de datos es ODBC (es decir, no existen drivers disponibles para el SGBD).**

e) Los tipos de conectores 1 y 2 exigen instalación de software en el puesto cliente. El tipo 4 no exige instalación en el cliente.

44

4. ¿El siguiente código Java es correcto? Razona la respuesta:

**import** java.sql.\*;

**public class** Main {

**public static void** main(String[] args) {

Class.*forName*("com.mysql.j dbc.Driver");

Connection conexión = DriverManager.getConnection

("jdbc:mysql://localhost/ejemplo","ejemplo", "ejemplo"); Statement sentencia = conexión.createStatement();

ResultSet resul = sentencia.executeQuery ("SELECT \* FROM empleados");

**while** (resul.next())

{

System.out.println (resul.getlnt("EMP\_NO") + " \* " +

resul.getString("APELLIDO"));

}

resul.close(); sentencia.close(); conexión.cióse();

}

}

Está mal porque en la URL del driver tiene una pequeña separación y Java va a intentar buscar ese driver con separación incluida.