

# JDBC: Conexiones con bases de datos desde Java

Luis Fernando Llana Díaz

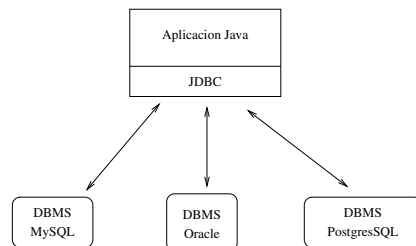
Departamento de Sistemas Informáticos y Computación  
Universidad Complutense de Madrid

10 de mayo de 2007

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## JDBC

- Conexión con una fuentes de datos.
- Realizar peticiones y actualizaciones.
- Manejar los resultados de las consultas



Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Drivers JDBC

Es necesario un *driver* dependiente del proveedor de la Base de datos.

MySQL ↔ MySQL java connector

## Creando la base de datos

Fichero `crea.sql`

```
create database valores; 1
use valores; 2
3
create table valores ( 4
    id int unsigned auto_increment primary key, 5
    nombre varchar(255), 6
    ibex_id varchar(100), 7
    url varchar(255), 8
    tipo enum("acciones","fondos") 9
); 10
11
create table datos ( 12
    id int unsigned auto_increment primary key, 13
    empresa int unsigned, 14
    fecha date, 15
    valor double 16
); 17
18
grant all on valores.* to luis identified by 'patata'; 19
```

```
~/sql$ mysql -u root < crea.sql 1
```

## Interfaz `java.sql.Connection`

- Establece conexión con la base de datos. No tiene constructor.
- Es un interfaz (está implementado en el *driver*).
- Constuye instrucciones SQL *precompiladas* (método `prepareStatement`).

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Realizando conexión con la base de datos

```
package valores;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Conexion {
    protected static Connection getMysqlConnection(String url, String db,
                                                    String user, String passwd)
        throws ClassNotFoundException,
            InstantiationException,
            SQLException,
            IllegalAccessException {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        return DriverManager.getConnection("jdbc:mysql://" + url + "/" + db +
            "?user=" + user + "&password=" + passwd);
    }
    protected static Connection getMysqlConnection()
        throws ClassNotFoundException,
            InstantiationException,
            SQLException,
            IllegalAccessException {
        return getMysqlConnection("localhost", "valores", "luis", "patata");
    }
    protected static String getFich(String fich) {
        java.net.URL url = ClassLoader.getResource(fich);
        return url.getPath();
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Ficheros de datos

Fichero de valores: `valores.csv`

```
BSCH-BSCH-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accindi_1.htm 1
Endesa-ENDESA-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accindi_1.htm 2
Repsol-YPF-REPSOL YPF-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accindi_1.htm 3
Acciona-ACCIONA-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accindi_1.htm 4
Inditex-INDITEX-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accindi_1.htm 5
Teléfono-TELEFONICA-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accindi_1.htm 6
Banco Popular-BA.POPULAR-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accindi_1.htm 7
Ebro-Puleva-EBRO PULEVA-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/igbm_3_1.htm 8
Mittal-Steel-ARCELOR MIT.-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/igbm_2_1.htm 9
Indra-INDRA A-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/igbm_6_1.htm 10
EADS-EADS-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/igbm_2_1.htm 11
Unipapel-UNIPAPEL-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/igbm_3_1.htm 12
Madrid Bolsa-MADRID BOLSA-fondos-http://www.bolsamadrid.es/esp/mercados/fondos/htmfondos/0085alfa.htm 13
Plus Madrid-PLUSMADRID-fondos-http://www.bolsamadrid.es/esp/mercados/fondos/htmfondos/0085alfa.htm 14
FonCaixa 65-FONCAIXA 65 BOLSA INDICE ESPA&#241;A-fondos-http://www.bolsamadrid.es/esp/mercados/fondos/htmfondos/0015alfa100.htm 15
```

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Cargar datos

```
package valores;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;
public class LeeValores {
    private static void insertaDatos(Connection con, BufferedReader in)
        throws IOException, SQLException {
        .....
    }
    public static void main (String [] args) throws Exception {
        Connection con = null;
        BufferedReader in = null;
        try {
            con = Conexion.getMysqlConnection();
            in = new BufferedReader(new FileReader(Conexion.getFich("sql/valores.csv")));
            insertaDatos(con,in);
        } finally {
            if (con!=null) con.close();
            if (in!=null) in.close();
        }
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java



## Cargar datos

```
private static final String sqlIns =
    "insert into valores (nombre,ibex_id,url,tipo) "+
    "values ( ?, ? , ? , ?)";
private final static int NOMBRE=1;
private final static int IBEX_ID=2;
private final static int URL_BOLSA=3;
private final static int TIPO=4;

private static void insertaDatos(Connection con, BufferedReader in)
    throws IOException, SQLException {
    String linea = in.readLine();
    PreparedStatement pstmt = con.prepareStatement(sqlIns);
    while (linea!=null) {
        Scanner scan = new Scanner(linea);
        scan.useDelimiter(".");
        String nombre = scan.next();
        String ibex_id = scan.next();
        String tipo = scan.next();
        String url = scan.next();
        System.out.println(nombre+"."+ibex_id+"."+tipo+"."+url);
        pstmt.setString(NOMBRE, nombre);
        pstmt.setString(IBEX_ID, ibex_id);
        pstmt.setString(TIPO, tipo);
        pstmt.setString(URL_BOLSA, url);
        int n = pstmt.executeUpdate();
        linea=in.readLine();
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Ficheros de datos

Fichero de valores: `valores.csv`

```
Fecha-BSCH-ENDESA-REPSOL-YPF-ACCIONA-INDITEX-Telefónica-Banco Popular-Ebro-Puleva-Mittal Steel-INDRA-AEDS-Unipapel-Madrid Bolsa-PlusMadrid-Fon Caixa 65
28/03/02-9.6-17.05-14.45-.....12.35-17.33-5.49 2
01/04/02-9.6-17.05-14.45-.....12.35-17.33-5.49 3
..... 4
27/08/04-8.15-25.16-98.50-15-19-11.83-.....11.81-17.03-4.64 5
30/08/04-7.99-15.29-17.03-50.05-19.08-11.78-.....11.89-17.1-4.66 6
12/12/06-14.01-35.76-27.71-137.95-39.59-16.06-13.7-18.62-31.5-18.35-...21.08-25.22-8715 8
..... 8
19/12/06-13.98-35.15-26.96-138.2-40.78-16.11-13.7-19.17-31.25-18.15-24.20--21.51-25957-8.35 12
20/12/06-14.08-35.16-26.92-139.9-40.95-16.28-13.68-19.28-31.86-18.42-25.45-21.4-25107-8.37 12
21/12/06-14.06-35.43-26.6-140.9-40.61-16.15-13.65-19.25-31.86-18.46-25.79--21.2-25.19-8.33 12
..... 12
27/12/06-14.16-35.72-26.5-140-41.01-16.2-13.69-19.59-31.95-19.06-26.07-21.34-21.16-13.25-8.26 12
28/12/06-14.12-35.55-26.41-141.05-40.89-16.16-13.68-19.12-32.26-18.67-26.00-21.40-21405-25.14-8.18 12
29/12/06-14.14-35.83-26.2-141.1-40.81-16.12-13.73-19.2-32-18.61-26.00-21.50-21.23-2453-8.18 12
01/01/07-14.14-35.83-26.2-141.1-40.81-16.12-13.73-19.2-32-18.61-26.00-21.50-21.18-24624-8.18 12
02/01/07-14.49-35.57-26.63-143.05-40.79-16.39-13.97-19.46-32.87-18.88-26.29-21.85-21718-25.24-8.18 12
```

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Cargar datos

```
private static void insertaDatos(Connection con, BufferedReader in)
    throws IOException, SQLException, ParseException {
    int [] id_valores = valores(con);
    String linea = in.readLine();
    linea = in.readLine();
    String sql = "insert into datos (fecha,valor,precio) values ( ?, ? , ? )";
    PreparedStatement pstmt = con.prepareStatement(sql);
    while (linea!=null) {
        Scanner scan = new Scanner(linea);
        scan.useDelimiter(".");
        String strFecha=scan.next();
        if (!strFecha.equals("")) {
            DateFormat df = new SimpleDateFormat("dd/MM/yy");
            Date fecha = df.parse(strFecha);
            System.out.println(df.format(fecha));
            for (int i = 0; i < id_valores.length ; i++) {
                String strValor = scan.next();
                if (!strValor.equals("")) {
                    double precio = Double.parseDouble(strValor);
                    pstmt.setDate(1,new java.sql.Date(fecha.getTime()));
                    pstmt.setInt(2,id_valores[i]);
                    pstmt.setDouble(3,precio);
                    int n = pstmt.executeUpdate();
                }
            }
        }
        linea = in.readLine();
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Cargar datos

```
private static String [] valores = {
    "BSCH", "ENDESA", "REPSOL YPF",
    "ACCIONA", "INDITEX", "TELEFONICA",
    "BA.POPULAR", "EBRO PULEVA", "ARCELOR MIT.",
    "INDRA A", "EADS", "UNIPAPEL", "MADRID BOLSA",
    "PLUSMADRID", "FONCAIXA 65 BOLSA INDICE ESPA&#241;A"
};
private static int [] valores(Connection con) throws SQLException {
    int [] ids = new int[valores.length];
    String sql = "select id from valores where ibex_id like ?";
    PreparedStatement pstmt = con.prepareStatement(sql);
    for (int i = 0; i < ids.length; i++) {
        pstmt.setString(1,valores[i]);
        ResultSet rs = pstmt.executeQuery();
        rs.next();
        ids[i]=rs.getInt(1);
    }
    return ids;
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19

Luis Fernando Llana Díaz Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid  
JDBC: Conexiones con bases de datos desde Java

## Realizar actualizaciones

Instrucciones: **INSERT, UPDATE, DELETE**

```
public static void pr2(Connection con) throws SQLException { 1
    String sql = "update datos set precio=precio+1"; 2
    PreparedStatement pstmt = con.prepareStatement(sql); 3
    int n = pstmt.executeUpdate(); 4
    System.out.println(n+" filas actualizadas"); 5
    SQLWarning warning = pstmt.getWarnings(); 6
    if (warning!=null) { 7
        System.out.println("AVISOS..."); 8
        while (warning!=null) { 9
            System.out.println("Message: " + warning.getMessage()); 10
            warning = warning.getNextWarning(); 11
        } 12
    } else { 13
        System.out.println("No hay avisos"); 14
    } 15
} 16
```

## Realizar Consultas

Instrucciones Select

```
select id,nombre,tip from valores 1
```

```
ResultSet rs = pstmt.executeQuery();  
rs.next();
```

rs	id	nombre	tipo
rs 1		BSCH	acciones
rs 2		Endesa	acciones
rs 3		Repsol-YPF	acciones
rs 4		Acciona	acciones
rs 5		Inditex	acciones
rs 6		Telefónica	acciones

## Ejemplos de consultas I

```
public static void pr1(Connection con) throws SQLException {
    String sql = "select * from valores";
    PreparedStatement pstmt = con.prepareStatement(sql);
    ResultSet rs = pstmt.executeQuery();
    while (rs.next()) {
        int id = rs.getInt("id");
        String nombre = rs.getString("nombre");
        System.out.println(id+"-"+nombre);
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

## Ejemplos de consultas II

```
public static void pr3(Connection con) throws SQLException {
    String sql = "select valores.nombre, datos.fecha, datos.precio "+
        " from datos left join valores on valores.id=datos.valor "+
        " where datos.fecha>='2007-03-01' "+
        " order by valores.tipo, valores.nombre, datos.fecha";
    PreparedStatement pstmt = con.prepareStatement(sql);
    ResultSet rs = pstmt.executeQuery();
    while (rs.next()) {
        String nombre = rs.getString("valores.nombre");
        Date fecha = rs.getDate("datos.fecha");
        double precio = rs.getDouble("datos.precio");
        DateFormat df = new SimpleDateFormat("dd 'de' MMMM 'de' yyy");
        NumberFormat nf = new DecimalFormat("#,000.00'€'");
        System.out.println(df.format(fecha)+"\t"+nombre+"\t"+nf.format(precio));
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16

## Ejemplos de consultas III

```
public static int getID(String ibex_id, Connection con) throws SQLException { 1
    String sql = "select id from valores where ibex_id like '"+ibex_id+"'"; 2
    PreparedStatement pstmt = con.prepareStatement(sql); 3
    ResultSet rs = pstmt.executeQuery(); 4
    if (rs.next()) { 5
        return rs.getInt(1); 6
    } else { 7
        return -1; 8
    } 9
} 10
```

## Consiguiendo datos desde [www.bolsamadrid.es](http://www.bolsamadrid.es)

- Conseguir el fichero HTML.
- Analizar el fichero.
- Añadir datos en la base de datos.

## Conseguir el fichero HTML

Fichero de valores: `valores.csv`

```
BSCH-BSCH-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accind1_1.htm 1
Endesa-ENDESA-acciones-http://www.bolsamadrid.es/esp/mercados/acciones/accind1_1.htm 2
```

```
private static String cargaPagina(java.net.URL url) 1
throws IOException{ 2
    InputStream in = url.openStream(); 3
    StringBuffer sb = new StringBuffer(); 4
    int c = in.read(); 5
    while (c!=-1) { 6
        sb.append((char)c); 7
        c = in.read(); 8
    } 9
    String res = sb.toString(); 10
    return res; 11
} 12
```

## Analizar el fichero HTML

Fichero de valores:

[http://www.bolsamadrid.es/esp/mercados/acciones/accind1\\_1.htm](http://www.bolsamadrid.es/esp/mercados/acciones/accind1_1.htm)

```
..... 1
....<IMG SRC="/images/arr-up9.gif" BORDER=0> BSCH</A></TD><TD>13,50</TD>..... 2
..... 3
....<IMG SRC="/images/arr-dw9.gif" BORDER=0> ENDESA</A></TD><TD>40,29</TD>... 4
```

```
private static double getValor(String ibex_id, String datos, String tipo) 1
throws ParseException { 2
    String regexp=null; 3
    if (tipo.equals("acciones")) { 4
        regexp=ibex_id+" *</A></TD><TD>([0-9,]+)</TD>"; 5
    } else if (tipo.equals("fondos")) { 6
        regexp=ibex_id+" *</a></TD>.*?<TD *[->]*>([0-9,]+)</TD>"; 7
    } else { 8
        throw new RuntimeException("Tipo \''+tipo+' desconocido"); 9
    } 10
    Pattern p = Pattern.compile(regexp); 11
    Matcher m = p.matcher(datos); 12
    if (m.find()) { 13
        // Double.parseDouble no vale, no analiza 13,50 14
        NumberFormat nf = NumberFormat.getInstance(); 15
        return nf.parse(m.group(1)).doubleValue(); 16
    } else { 17
        throw new RuntimeException("Acciones \''+ibex_id+' no encontrada"); 18
    }
}
```

## Insertar los datos en la Base de datos

```
private static final String sqlIns = 1
    "insert into datos (fecha,valor,precio) values ( ? , ? , ?)"; 2
private static final int FECHA=1; 3
private static final int VALOR=2; 4
private static final int PRECIO=3; 5
6
private static void insertaDatos(java.sql.Date fecha,Connection con) 7
    throws SQLException, IOException, ParseException { 8
    HashMap<String,String> paginas = new HashMap<String,String>(); 9
    String sql = "select * from valores"; 10
    PreparedStatement pstmtValores = con.prepareStatement(sql); 11
    ResultSet rs = pstmtValores.executeQuery(); 12
    PreparedStatement pstmtIns = con.prepareStatement(sqlIns); 13
    while (rs.next()) { 14
        insertaDatos(pstmtIns, 15
            fecha, 16
            rs,paginas); 17
    } 18
} 19
```

## Insertar los datos en la Base de datos

```
private static void insertaDatos(PreparedStatement pstmtIns, 1
    java.sql.Date fecha, 2
    ResultSet rs, 3
    HashMap<String,String> paginas) 4
    throws SQLException, IOException, ParseException { 5
    String url = rs.getString("url"); 6
    String datos = paginas.get(url); 7
    if (datos==null) { 8
        datos = cargaPagina(paginas,url); 9
        paginas.put(url,datos); 10
    } 11
    String tipo = rs.getString("tipo"); 12
    String ibex_id = rs.getString("ibex_id"); 13
    double valor = getValor(ibex_id,datos,tipo); 14
    int id = rs.getInt("id"); 15
    String nombre = rs.getString("nombre"); 16
    pstmtIns.setDate(FECHA,fecha); 17
    pstmtIns.setInt(VALOR,id); 18
    pstmtIns.setDouble(PRECIO,valor); 19
    int n = pstmtIns.executeUpdate(); 20
} 21
```

## Consiguiendo datos automáticamente

```
#!/bin/sh 1
# $Id: transparencias.tex,v 1.8 2007-05-10 11:37:22 luis Exp $ 2
3
JAVA_HOME=/opt/java/jdk1.5 4
CLASSPATH=/home/casa/ahorros2/classes 5
$JAVA_HOME/bin/java -cp $CLASSPATH valores.ConsigueDatos $* 6
```

```
$ crontab -l 1
2
SHELL=/bin/bash 3
MAILTO=luis 4
# Directorios donde buscar programas 5
PATH=/bin:/usr/bin:/usr/X11R6/bin:/usr/local/bin:/home/luis/bin 6
7
45 23 * * 1,2,3,4,5 /home/casa/ahorros2/scripts/consigueDatos.sh > /dev/null 2>&1
```