

JAVA STANDARD EDITION

Día 3

Oscar Chávez-Bosquez

<http://chavezbosquez.github.io/diplomado/java.html>

Hoy veremos...

1. *Refresh*
2. Cuadros de diálogo Swing
3. Manejo de archivos
4. Introducción a Base de Datos
5. Java DB

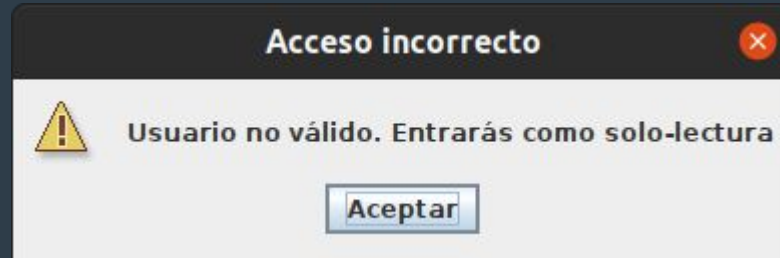
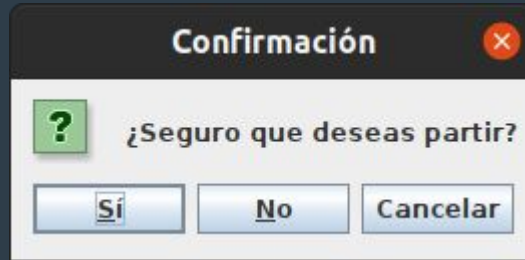
Refresh...

Digital Learning SL: [Test de Java y POO](#)

Cibertest: [Programación en Java Básico](#)

Daypo: [Test de Java, nivel básico](#)

<https://chavezbosquez.github.io/diplomado/java.html>



Cuadros de diálogo

Cuadros de diálogo Swing

Clase **JOptionPane**: cuadros de diálogo estándar en Swing.

Tipos:

- **showMessageDialog()**: Mensajes de texto.
- **showConfirmDialog()**: Cuadros de confirmación.
- **showInputDialog()**: Permiten la entrada de datos.

<https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>

FrameLoginMejorado.java

Código de `btnAceptarActionPerformed()` :

```
// Validar datos de entrada
String usuario = txtUsuario.getText();
String contra = txtContraseña.getText();
if (usuario == null || usuario.isBlank()) {
    JOptionPane.showMessageDialog(this, "Introduce un usuario", "Error",
                                JOptionPane.ERROR_MESSAGE);

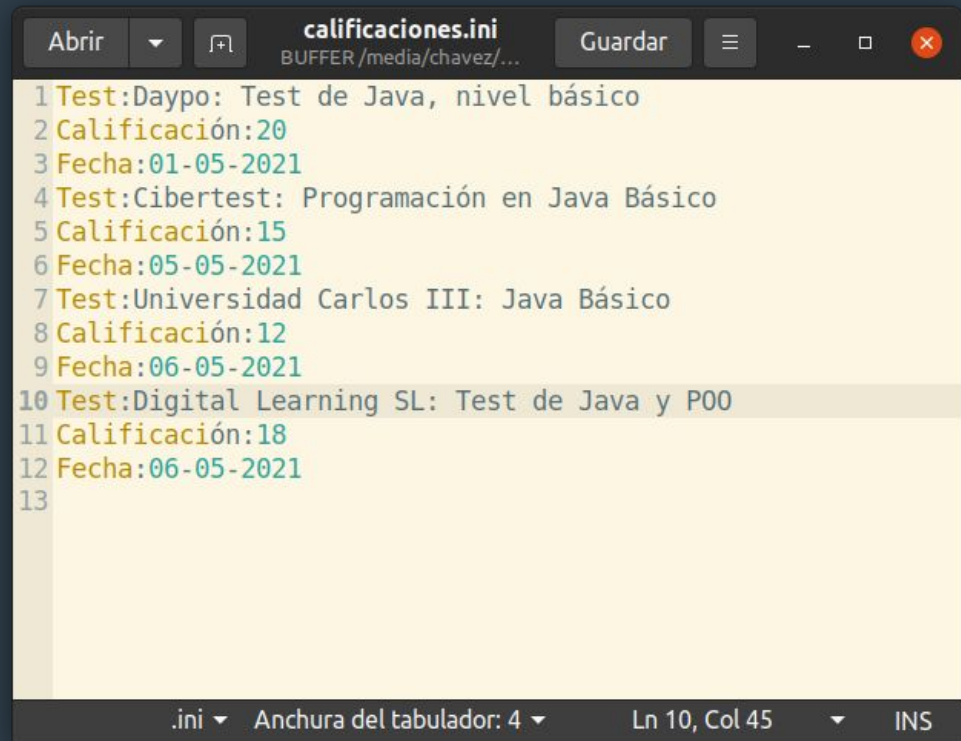
    return;
}
if (contra == null || contra.isBlank()) {
    JOptionPane.showMessageDialog(this, "Introduce una contraseña", "Error",
                                JOptionPane.ERROR_MESSAGE);

    return;
}
// Validar acceso
if (usuario.equals("chavez") && contra.equals("chavez")) {
    JOptionPane.showMessageDialog(this, "Bienvenido " + usuario, "Acceso correcto",
                                JOptionPane.INFORMATION_MESSAGE);
} else {
    JOptionPane.showMessageDialog(this, "Usuario no válido. Entrarás como solo-lectura",
                                "Acceso incorrecto", JOptionPane.WARNING_MESSAGE);
}
```

FrameLoginMejorado.java

Código de `btnCancelarActionPerformed()` :

```
int respuesta = JOptionPane.showConfirmDialog(this,  
                                                "¿Seguro que deseas partir?",  
                                                "Confirmación",  
                                                JOptionPane.YES_NO_CANCEL_OPTION);  
  
if (respuesta == 0) {  
    System.exit(0);  
}
```



The screenshot shows a text editor window with the title 'calificaciones.ini' and the path 'BUFFER /media/chavez/...'. The editor contains a list of test results, each with a line number, a 'Test' name, a 'Calificación' (score), and a 'Fecha' (date). The text is color-coded: 'Test' is blue, 'Calificación' is orange, and 'Fecha' is green. The list includes tests from Daypo, Cibertest, Universidad Carlos III, and Digital Learning SL. The status bar at the bottom indicates the file type is '.ini', the tab width is 4, the cursor is at line 10, column 45, and the encoding is 'INS'.

```
1 Test:Daypo: Test de Java, nivel básico
2 Calificación:20
3 Fecha:01-05-2021
4 Test:Cibertest: Programación en Java Básico
5 Calificación:15
6 Fecha:05-05-2021
7 Test:Universidad Carlos III: Java Básico
8 Calificación:12
9 Fecha:06-05-2021
10 Test:Digital Learning SL: Test de Java y P00
11 Calificación:18
12 Fecha:06-05-2021
13
```

.ini ▾ Anchura del tabulador: 4 ▾ Ln 10, Col 45 ▾ INS

Manejo de archivos

Manejo de archivos

Clases principales:

- **File**: *Apuntador* a un archivo o directorio.
- **FileWriter**: Escritura de archivos.
- **FileReader**: Lectura de archivos.

FrameCalificaciones.java

Código de `btnGuardarActionPerformed()` :

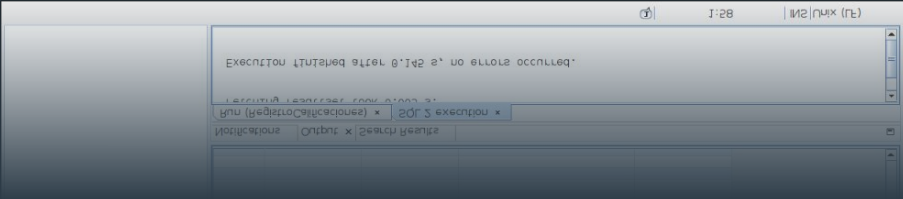
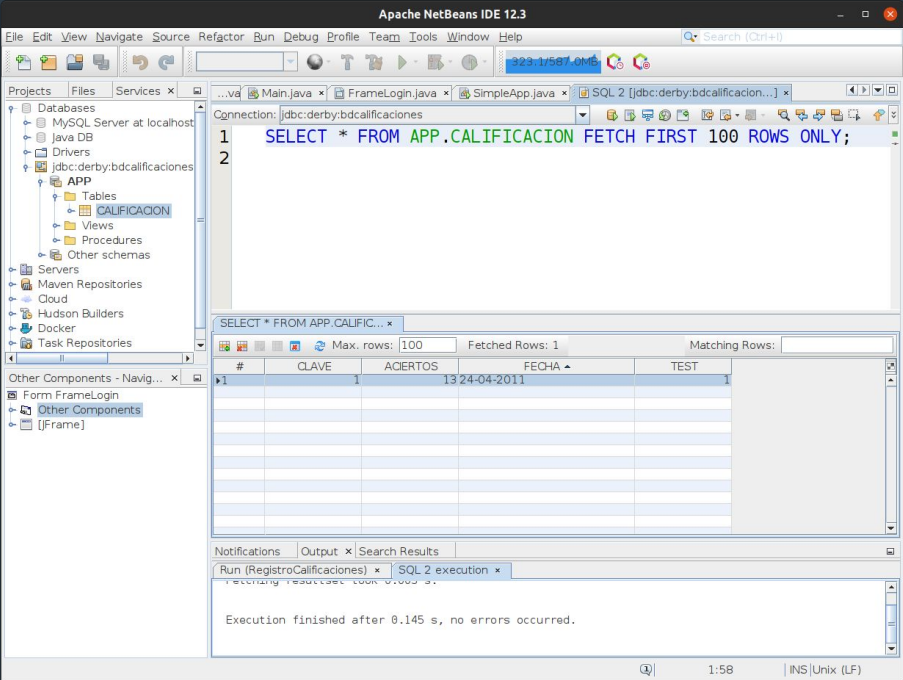
```
File archivo = new File("calificaciones.ini");
try {
    archivo.createNewFile();
    FileWriter writer = new FileWriter(archivo, true);
    writer.write("Test:" + cboTest.getSelectedItem() + "\r\n");
    writer.write("Aciertos:" + spnAciertos.getText() + "\r\n");
    DateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");
    String fecha = formatter.format(calFecha.getDate());
    writer.write("Fecha:" + fecha + "\r\n");
    writer.flush();
    writer.close();
} catch (IOException ex) {
    System.err.println("Excepción:" + ex);
}
```

FrameCalificaciones.java

Código de `btnLeerArchivoActionPerformed()` :

```
File archivo = new File("calificaciones.ini");
try {
    FileReader reader = new FileReader(archivo);
    char[] texto = new char[(int)archivo.length()];
    reader.read(texto);
    String s = new String(texto);
    JOptionPane.showMessageDialog(this, s);
    reader.close();
} catch (FileNotFoundException ex) {
    System.err.println("Excepción:" + ex);
} catch (IOException ex) {
    System.err.println("Excepción:" + ex);
}
```

Base de Datos

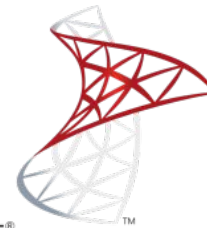


Base de Datos

- Colección organizada de información estructurada.
- Un archivo o un directorio.
- Bases de datos RELACIONALES.
- Filas y columnas en una serie de tablas.
- Lenguaje SQL.

Sistema Manejador de Base de Datos (SMBD)

ORACLE



Microsoft®
SQL Server®

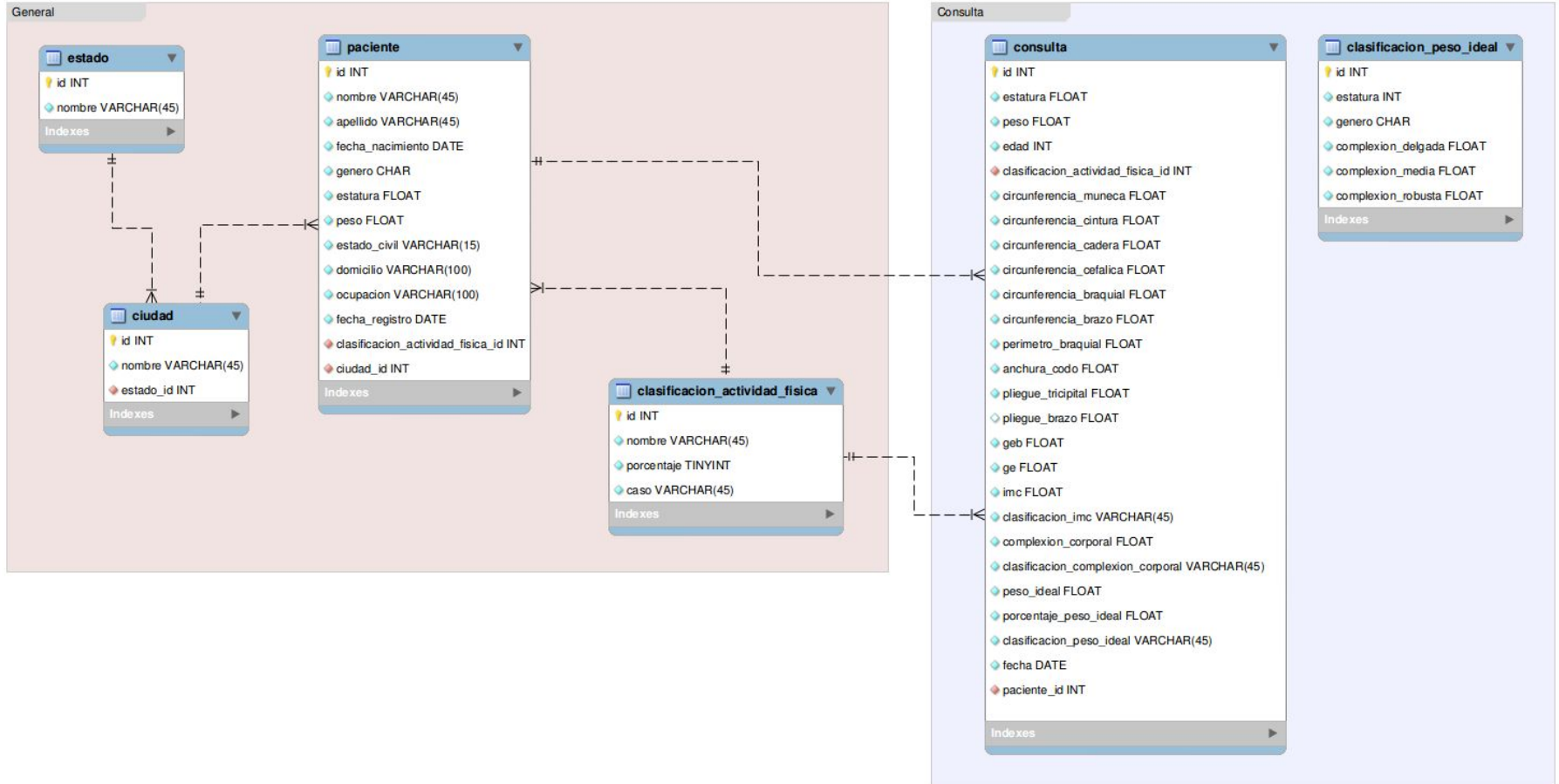


MariaDB®



MySQL™

Base de datos



Lenguaje SQL

- *Structured Query Language.*
- Lenguaje de definición de datos (DDL):
 - **SELECT** * **FROM** nombre_de_la_tabla **WHERE** condicion;
- Lenguaje de manipulación de datos (DML):
 - **CREATE TABLE** nombre_de_la_tabla;

Java DB



Java DB

Java DB is a distribution of the [Apache Derby open source database](#). It supports standard ANSI/ISO SQL through the JDBC and Java EE APIs.

[FAQs](#)[Documentation](#)

The **Apache DB Project**

<http://db.apache.org/>

http://db.apache.org/derby/derby_downloads.html

Archivo `pom.xml` (*Project Files*)



Agregar dentro de `<dependency>...</dependency>` :

```
<dependency>
  <groupId>org.apache.derby</groupId>
  <artifactId>derby</artifactId>
  <version>10.15.2.0</version>
</dependency>
<dependency>
  <groupId>org.apache.derby</groupId>
  <artifactId>derbyclient</artifactId>
  <version>10.15.2.0</version>
</dependency>
<dependency>
  <groupId>org.apache.derby</groupId>
  <artifactId>derbynet</artifactId>
  <version>10.15.2.0</version>
</dependency>
<dependency>
  <groupId>org.apache.derby</groupId>
  <artifactId>derbytools</artifactId>
  <version>10.15.2.0</version>
</dependency>
<dependency>
  <groupId>org.apache.derby</groupId>
  <artifactId>derbyLocale_es</artifactId>
  <version>10.15.2.0</version>
</dependency>
```

NetBeans 8.2



1. Descargar [Apache Derby](#)

Nota: *Revisar su versión de JDK*

2. Descomprimir el archivo zip
3. Copiar la carpeta **lib** dentro del directorio de tu proyecto

NetBeans 8.2



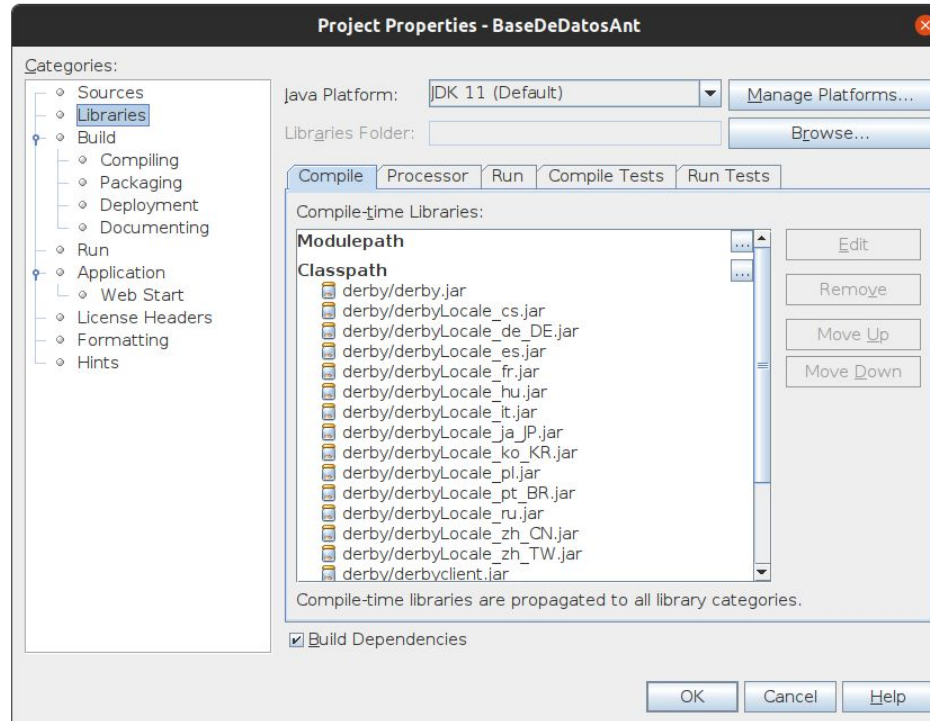
The image illustrates the steps to add a library to a project in NetBeans 8.2. It shows the context menu for the 'BaseDeDatosAnt' project, the 'Project Properties' dialog, and the 'Add JAR/Folder' dialog.

Context Menu: The context menu for the 'BaseDeDatosAnt' project is open, showing options like 'New', 'Build', 'Clean and Build', 'Run', 'Debug', 'Test', 'Run Selenium Tests', 'Run Maven', 'Set Configuration', 'Open POM', 'Reload POM', 'Open Required Projects', 'Close', 'Rename...', 'Move...', 'Copy...', 'Delete', 'Find...', 'Inspect and Transform...', 'Versioning', 'History', 'Properties', 'Build (Ant)', 'Create', 'Update', and 'Create'.

Project Properties - BaseDeDatosAnt: The 'Project Properties' dialog is open, showing the 'Libraries' category. The 'Compile-time Libraries' section is active, showing the 'Modulepath' and 'Classpath' tabs. The 'Add JAR/Folder' button is highlighted.

Add JAR/Folder: The 'Add JAR/Folder' dialog is open, showing the search results for 'lib'. The 'derby.jar' file is selected. The 'Nombre de archivo' field is set to '/home/chavez/NetBeansProjects/BaseDeDatosAnt/lib' and the 'Archivos de tipo' is set to 'Classpath Entry (folder, ZIP or JAR file)'. The 'Abrir' button is highlighted.

NetBeans 8.2



¿Funciona? Main.java

```
public static void main(String[] args) {  
    try {  
        Class.forName("org.apache.derby.jdbc.EmbeddedDriver");  
        System.out.println("Driver cargado con éxito");  
    } catch (ClassNotFoundException ex) {  
        System.err.println("Excepción:" + ex);  
    }  
}
```

Crear base de datos. Main.java

```
public static void main(String[] args) {
    try {
        Class.forName("org.apache.derby.jdbc.EmbeddedDriver");
        Connection conexion = DriverManager.getConnection("jdbc:derby:basededatos;create=true");
        DatabaseMetaData metadata = conexion.getMetaData();
        ResultSet resultado = metadata.getTables(null, null, null, new String[]{"TABLE"});
        if (!resultado.next()) {
            String sql = "CREATE TABLE calificacion(clave INT PRIMARY KEY
                        + "GENERATED ALWAYS AS IDENTITY(Start with 1, Increment by 1),"
                        + " aciertos INTEGER,"
                        + " fecha VARCHAR(20),"
                        + " test INTEGER)";

            Statement sentencia = conexion.createStatement();
            sentencia.execute(sql);
            System.out.println("Base de Datos creada");
        } else {
            System.out.println("Ya existe la BD");
        }
        resultado.close();
        conexion.close();
        DriverManager.getConnection("jdbc:derby:basededatos;shutdown=true");
    } catch (ClassNotFoundException ex) {
        System.err.println(ex);
    } catch (SQLException ex) {
        System.err.println(ex);
    }
}
```

Archivo Conexion.java

```
public Connection getConnection() {  
    try {  
        Class.forName("org.apache.derby.jdbc.EmbeddedDriver");  
        Connection conexion =  
            DriverManager.getConnection("jdbc:derby:basededatos;create=true");  
        return conexion;  
    } catch (ClassNotFoundException ex) {  
        System.err.println(ex);  
        return null;  
    } catch (SQLException ex) {  
        System.err.println(ex);  
        return null;  
    }  
}
```

Guardar los datos: Conexion.java

```
public boolean guardarDatos(int aciertos, String fecha, int test) {
    Connection conexion = this.getConexion();
    try {
        PreparedStatement psInsert = conexion.prepareStatement(
            "INSERT INTO calificacion(aciertos, fecha, test) VALUES(?,?,?)");
        psInsert.setInt(1, aciertos);
        psInsert.setString(2, fecha);
        psInsert.setInt(3, test);
        psInsert.executeUpdate();
        psInsert.close();
        conexion.close();
        return true;
    } catch (SQLException ex) {
        System.err.println(ex);
        return false;
    }
}
```


Guardar los datos: FrameCalificaciones.java

```
private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {  
    int test = cboTest.getSelectedIndex();  
    int aciertos = Integer.parseInt(spnAciertos.getText());  
    DateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");  
    String fecha = formatter.format(calFecha.getDate());  
    Conexion conexion = new Conexion();  
    if (conexion.guardarDatos(aciertos, fecha, test)) {  
        JOptionPane.showMessageDialog(this, "Datos guardados");  
    } else {  
        JOptionPane.showMessageDialog(this, "Error al guardar los datos", "Error",  
                                     JOptionPane.ERROR_MESSAGE);  
    }  
    conexion.cerrarConexion();  
}
```

Leer los datos: Conexion.java

```
public String getDatos() {  
    Connection conexion = this.getConexion();  
    try {  
        Statement sentencia = conexion.createStatement();  
        ResultSet resultado = sentencia.executeQuery(  
            "SELECT clave,aciertos,fecha,test FROM calificacion");  
        String texto = "";  
        while (resultado.next()) {  
            int id = resultado.getInt(1);  
            int aciertos = resultado.getInt(2);  
            String fecha = resultado.getString(3);  
            int test = resultado.getInt(4);  
            texto = texto + aciertos + " " + fecha + " " + test + "\n";  
        }  
        resultado.close();  
        return texto;  
    } catch (SQLException ex) {  
        System.err.println(ex);  
        return null;  
    }  
}
```

Leer los datos: FrameCalificaciones.java

```
private void btnLeerActionPerformed(java.awt.event.ActionEvent evt) {  
    Conexion conexion = new Conexion();  
    String datos = conexion.getDatos();  
    JOptionPane.showMessageDialog(this, datos);  
    conexion.cerrarConexion();  
}
```

No olviden cerrar la conexión: `Conexion.java`

```
public void cerrarConexion() {  
    try {  
        DriverManager.getConnection("jdbc:derby:basededatos;shutdown=true");  
    } catch (SQLException ex) {  
        System.err.println(ex);  
    }  
}
```