

EJERCICIO GUIADO. JAVA. ACCESO A BASE DE DATOS

Representación de Tablas en un JTable

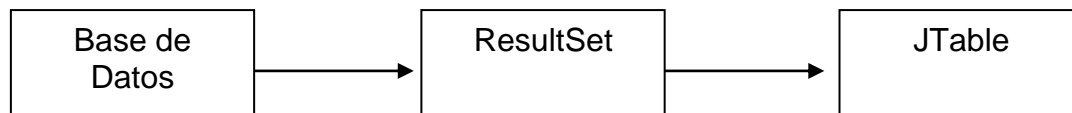
En la hoja guiada anterior vimos que existe un tipo de objeto llamado JTable que nos permite introducir tablas en nuestras aplicaciones.

Estos objetos son ideales para mostrar el contenido de una tabla de la base de datos o del resultado de una consulta.

Recuerda que para extraer datos de la base de datos realizamos una consulta SQL de tipo SELECT cuyo resultado se almacena en un objeto *ResultSet*.

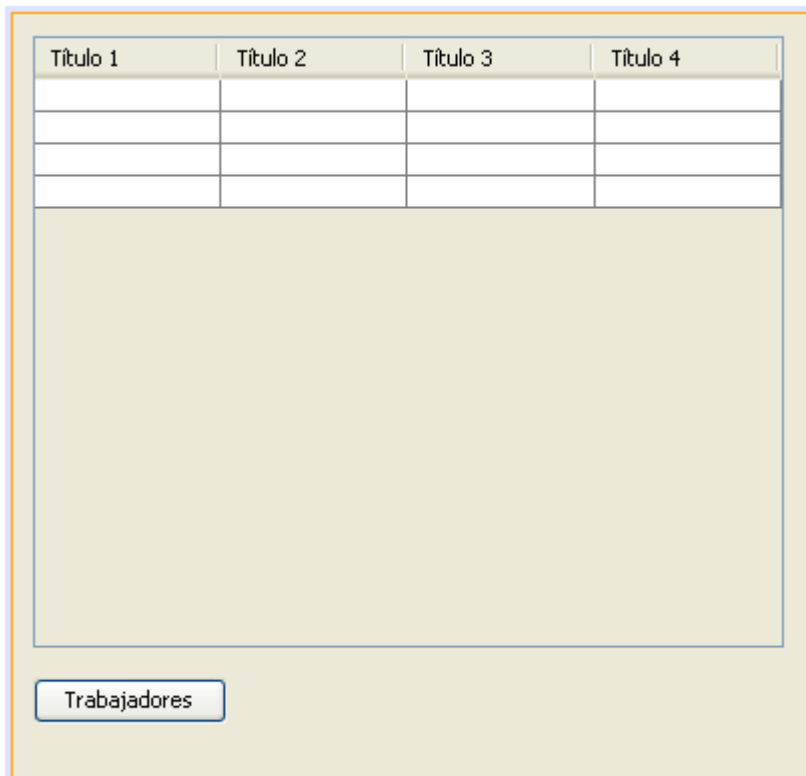
Un *ResultSet* básicamente es una tabla almacenada en memoria (y por tanto no visible) Sin embargo, en esta hoja guiada veremos como trasladar el contenido de un *ResultSet* a un JTable para que el usuario pueda visualizar los datos de forma cómoda.

El proceso será básicamente el siguiente:



EJERCICIO GUIADO Nº 1

1. Crear un proyecto java y prepararlo para que pueda acceder a la base de datos MANEMPSA.MDB:
 - a. Crear una subcarpeta *Base* y copiar dentro de ella el fichero de base de datos MANEMPSA.MDB
 - b. Declarar los dos objetos globales: *conexión* y *sentencia*.
 - c. Programar el método *PrepararBaseDatos* de forma que el programa acceda a la base de datos y se construyan los objetos *conexión* y *sentencia*.
 - d. Programar el evento *windowClosing* de la ventana principal de forma que se cierre la conexión.
2. Una vez hecho esto, añade a la ventana principal un objeto *JTable* y llámalo simplemente *tabla*.
3. Añade en la parte inferior un botón llamado *btnTrabajadores*. La ventana principal quedará así:



4. Agregaremos al proyecto un objeto "modelo de tabla" (*DefaultTableModel*) llamado *m* que sea global. También añadiremos en el constructor una llamada a un método *PrepararTabla* y programaremos dicho método de la siguiente forma:

```

public class ventanaprincipal extends javax.swing.JFrame {

    DefaultTableModel m; ←
    Connection conexion;
    Statement sentencia;

} /** Creates new form ventanaprincipal */
} public ventanaprincipal() {
    initComponents();
    PrepararBaseDatos();
    PrepararTabla(); ←
- }

{ void PrepararTabla() {

    String titulos[] = new String[4];
    m = new DefaultTableModel(null, titulos);
    tabla.setModel(m);

}

```

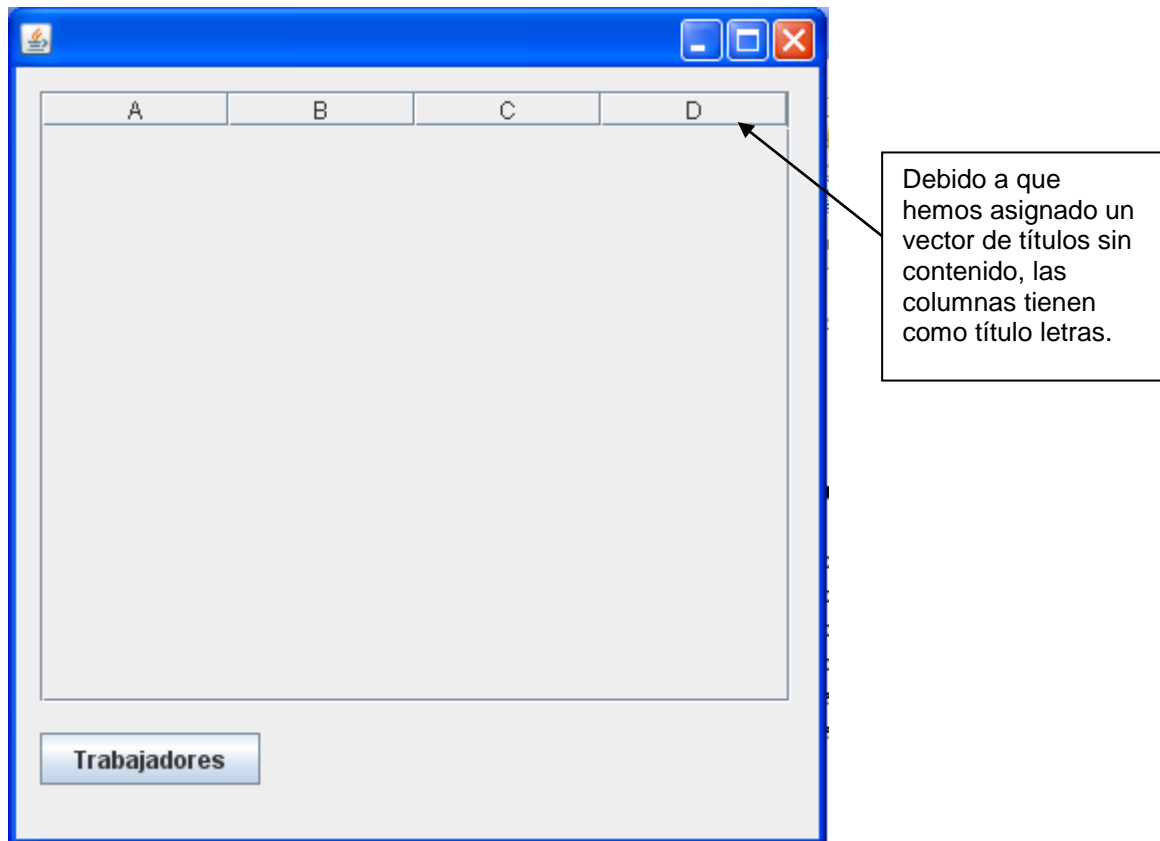
Si observas el código, verás que en el método *PrepararTabla* se crea un vector de cadenas con cuatro elementos pero que no contiene nada.

Luego se crea el *modelo de tabla* a partir de este vector vacío.

Y se asigna finalmente a la tabla.

El resultado de esto es que la tabla aparecerá vacía y sin títulos (En realidad, como títulos aparecen letras: A, B, C, etc)

5. Ejecuta el programa para ver su aspecto de momento.



6. Bien, ahora programaremos el botón *btnTrabajadores* de forma que al pulsarlo se muestre el contenido de la tabla *Trabajadores*. Programe lo siguiente:

```
private void btnTrabajadoresActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agregue su código aquí:
    try {
        //Extraer datos de la tabla y almacenarlos en el resultSet
        ResultSet r = sentencia.executeQuery("select * from trabajadores");

        //Crear el modelo y definir la cabecera de la tabla
        String titulos[] = {"DNI", "Nombre", "Apellidos", "Sueldo", "Fecha", "Matrícula"};
        m=new DefaultTableModel(null, titulos);
        tabla.setModel(m);

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error al extraer los datos de la tabla");
    }
}
```

Estudiemos el código detenidamente.

Lo primero que se hace es extraer el contenido de la tabla *Trabajadores* de la base de datos, ejecutando una consulta SQL usando el objeto *sentencia*. El resultado de dicha consulta se almacenará en un objeto *ResultSet* llamado *r*.

```
ResultSet r = sentencia.executeQuery("select * from trabajadores");
```

A continuación, se extraerá información del *ResultSet* y se introducirá en la tabla. Lo primero que se hace es crear el modelo de la tabla definiendo la cabecera. Observa el código.

```
String titulos[] = {"DNI","Nombre","Apellidos","Sueldo","Fecha","Matrícula"};  
m=new DefaultTableModel(null,titulos);  
tabla.setModel(m);
```

Como ves, lo que hacemos es definir un vector de cadenas con los títulos correspondientes a los campos de la tabla, y luego usamos este vector para crear el modelo de la tabla.

Finalmente asignamos el modelo creado a la tabla.

7. Ejecute el programa de momento. Observe como al pulsar el botón *Trabajadores* la cabecera de la tabla pasa a contener los campos indicados en el vector *títulos*.



8. Ahora ya solo nos queda recoger todo el contenido del *ResultSet* y mostrarlo en la tabla. Para ello, modifique el código del *actionPerformed* del botón *Trabajadores* para que quede como sigue:

```

private void btnTrabajadoresActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agregue su código aquí:
    try {
        //Extraer datos de la tabla y almacenarlos en el resultSet
        ResultSet r = sentencia.executeQuery("select * from trabajadores");
        //Crear el modelo y definir la cabecera de la tabla
        String titulos[] = {"DNI","Nombre","Apellidos","Sueldo","Fecha","Matricula"};
        m=new DefaultTableModel(null,titulos);

        String fila[] = new String[6];
        while(r.next()) {
            fila[0]=r.getString("DNI");
            fila[1]=r.getString("Nombre");
            fila[2]=r.getString("Apellidos");
            fila[3]=r.getString("Sueldo");
            fila[4]=r.getString("Fecha");
            fila[5]=r.getString("Matricula");
            m.addRow(fila);
        }
        tabla.setModel(m);

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,"Error al extraer los datos de la tabla");
    }
}

```

Si estudiamos el código, veremos que se extrae la tabla trabajadores entera introduciéndola en un *ResultSet* llamado r, a través de una instrucción SELECT de SQL.

Luego construimos un modelo de tabla (*DefaultTableModel*) llamado m usando un vector de títulos, que se corresponde a los campos de la tabla trabajadores.

Una vez hecho esto, creamos un vector de seis elementos, correspondiente a los seis datos que contiene cada fila, y empezamos a recorrer el *ResultSet* usando el típico:

```
while (r.next()) {
```

En cada vuelta del bucle tomamos los valores de la fila del *ResultSet* y los almacenamos en el vector (en la posición correspondiente)

Luego se añade el vector que se ha construido al modelo de la tabla.

Así pues, en cada vuelta se añadirá un registro más al modelo m. Estos registros son extraídos del *ResultSet*.

Cuando el bucle ha terminado, el modelo de la tabla contiene exactamente lo mismo que el *ResultSet*. Es decir, contiene la tabla *trabajadores*, ya que este fue el resultado de la consulta SQL ejecutada.

Ya solo tenemos que asignar el modelo de la tabla al *JTable* correspondiente, es decir, a la tabla, con lo que esta visualizará los datos del modelo.

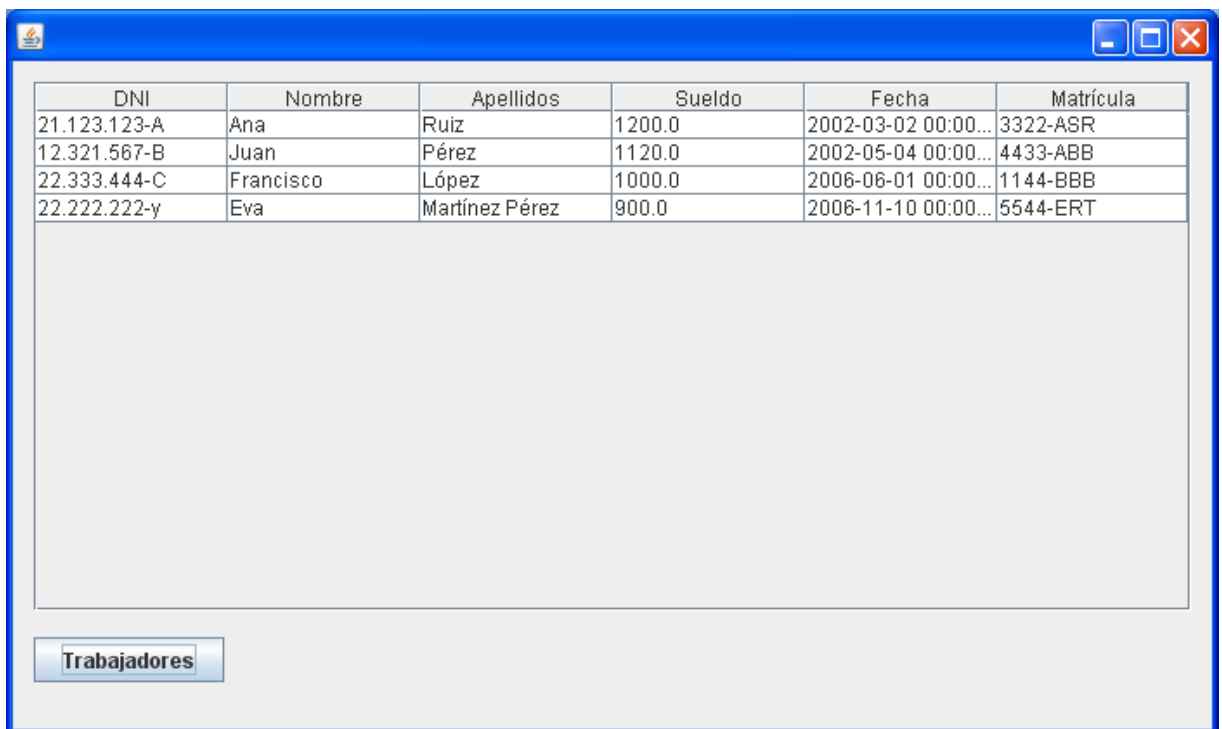
Todo este proceso se rodea con un try...catch para evitar errores inesperados.

9. Para entender el código anterior aquí tienes un pequeño gráfico explicativo:

Base de Datos → ResultSet → Vector → Modelo de Tabla → JTable

- De la base de datos extraemos datos a un *ResultSet*. (Esto se hace ejecutando una consulta SQL de tipo SELECT a través del objeto *sentencia*)
- Del *ResultSet* extraemos los datos a un modelo de tabla. (Esto se hace recorriendo el *ResultSet* y almacenando cada registro en un vector)
- Almacenamiento del vector en el modelo de tabla. (Cada vector extraído del *ResultSet* se inserta como fila en el modelo de tabla)
- Se asigna el modelo de tabla al objeto JTable. (Al hacer esto el objeto JTable muestra el contenido del modelo)

10. Ejecute el programa y observe el resultado al pulsar el botón *Trabajadores*. Como ve, el objeto *tabla* muestra entonces el contenido de la tabla *Trabajadores*.



DNI	Nombre	Apellidos	Sueldo	Fecha	Matricula
21.123.123-A	Ana	Ruiz	1200.0	2002-03-02 00:00...	3322-ASR
12.321.567-B	Juan	Pérez	1120.0	2002-05-04 00:00...	4433-ABB
22.333.444-C	Francisco	López	1000.0	2006-06-01 00:00...	1144-BBB
22.222.222-y	Eva	Martínez Pérez	900.0	2006-11-10 00:00...	5544-ERT

Trabajadores

11. Si observa la tabla verá que la fecha aparece en el formato estándar de almacenamiento:

Año-mes-día hora:min:seg

Y que los sueldos aparecen con punto decimal.

Mejore estos detalles haciendo los siguientes cambios en el botón *Trabajadores*:

```

private void btnTrabajadoresActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agrege su código aquí:
String fecha;
try {
//Extraer datos de la tabla y almacenarlos en el resultSet
ResultSet r = sentencia.executeQuery("select * from trabajadores");
//Crear el modelo y definir la cabecera de la tabla
String titulos[] = {"DNI","Nombre","Apellidos","Sueldo","Fecha","Matrícula"};
m=new DefaultTableModel(null,titulos);

String fila[] = new String[6];
while(r.next()) {
fila[0]=r.getString("DNI");
fila[1]=r.getString("Nombre");
fila[2]=r.getString("Apellidos");
fila[3]=r.getString("Sueldo").replace(".",",");
fecha=r.getString("Fecha");
fecha=fecha.substring(8,10)+"/"+fecha.substring(5,7)+"/"+fecha.substring(0,4);
fila[4]=fecha;
fila[5]=r.getString("Matricula");
m.addRow(fila);
}
tabla.setModel(m);

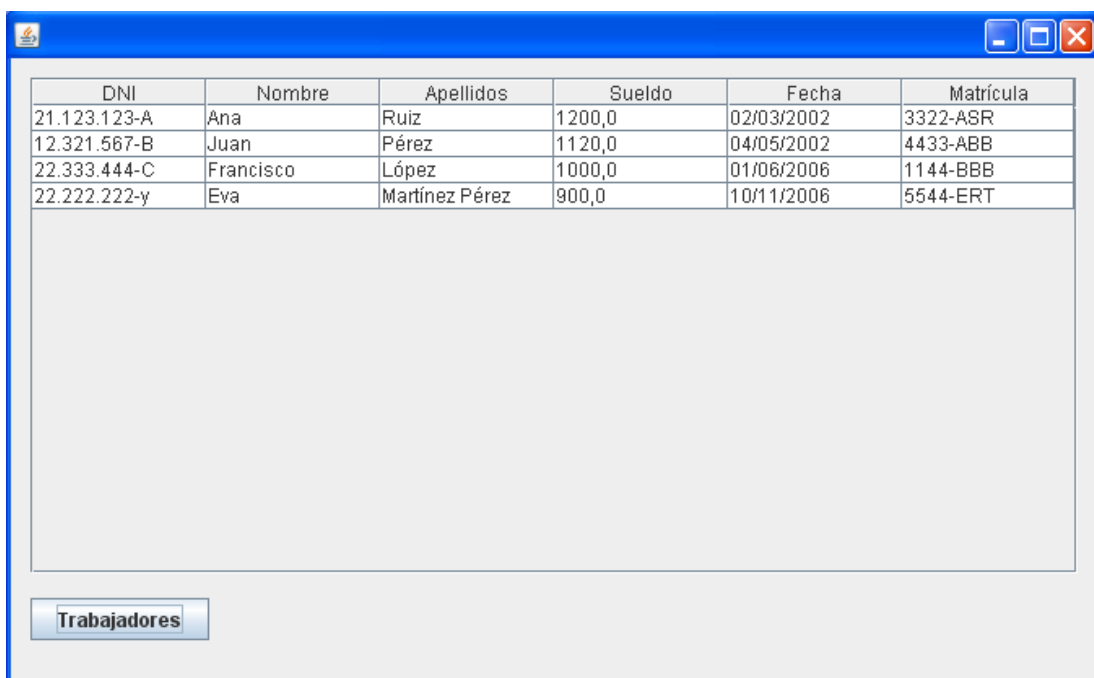
} catch (Exception e) {
JOptionPane.showMessageDialog(null,"Error al extraer los datos de la tabla");
}
}

```

Como se puede observar, antes de almacenar los sueldos en el modelo de la tabla se reemplazan el punto decimal por una coma.

Y en el caso de la fecha, se extrae de la cadena el año, mes y día y se concatenan en un formato más habitual: día/mes/año.

12. Ejecuta ahora el programa y observa como se muestran las fechas y los sueldos:



DNI	Nombre	Apellidos	Sueldo	Fecha	Matrícula
21.123.123-A	Ana	Ruiz	1200,0	02/03/2002	3322-ASR
12.321.567-B	Juan	Pérez	1120,0	04/05/2002	4433-ABB
22.333.444-C	Francisco	López	1000,0	01/06/2006	1144-BBB
22.222.222-y	Eva	Martínez Pérez	900,0	10/11/2006	5544-ERT

Trabajadores

CONCLUSIÓN

El proceso a seguir para mostrar una tabla de una base de datos en un JTable es el siguiente:

- **Realizar una consulta `SELECT` para extraer los datos de la tabla de la base de datos. El resultado se almacenará en un *ResultSet*.**
- **Recorrer el *ResultSet* almacenando cada una de sus filas en un vector y luego traspasando este vector a un modelo de tabla.**
- **Una vez que el modelo de tabla está relleno, asignar el modelo a un JTable.**