EJERCICIO GUIADO. JAVA. ACCESO A BASE DE DATOS

Recapitulando. Ejecución de consultas desde la aplicación.

Para ejecutar una consulta sobre la base de datos desde tu aplicación java se tiene que ejecutar una instrucción usando el objeto *sentencia* de la siguiente forma:

```
ResultSet r = sentencia.executeQuery("consulta_sql");
```

La *consulta_sql* es una cadena que tiene forma de consulta SQL bien escrita (sin fallos). Por ejemplo:

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

En este código que se acaba de mostrar, se crea una cadena llamada *consulta*. Se le asigna a esta cadena una instrucción SQL para extraer todos los trabajadores, y finalmente se ejecuta dicha instrucción SQL usando el objeto *sentencia*.

La cadena *consulta* puede ser construida a través de concatenaciones de cadenas. Observa atentamente este ejemplo. Es parecido al anterior pero tiene una ligera diferencia:

```
//Supongamos que txtTabla es un cuadro de texto

String consulta;
consulta="select * from " + txtTabla.getText();
ResultSet r = sentencia.executeQuery(consulta);
```

En este caso, la cadena consulta es la concatenación de…

`"select * from "`

con

lo que contenga el cuadro de texto txtTabla - es decir, `txtTabla.getText()`

La gran ventaja de esto, es que el usuario podrá escribir el nombre de una tabla cualquiera dentro del cuadro de texto *txtTabla* y al ejecutarse este código se extraerá el contenido de dicha tabla.

Es decir, el usuario podrá *decidir* lo que quiere consultar. El usuario afecta a la construcción de la consulta SQL, y por tanto, tiene cierto control sobre esta consulta.

La construcción de consultas SQL a partir de la concatenación de cadenas y datos proporcionados por el usuario es algo muy usado en los programas de base de datos. En esta hoja guiada se insistirá en esta idea.

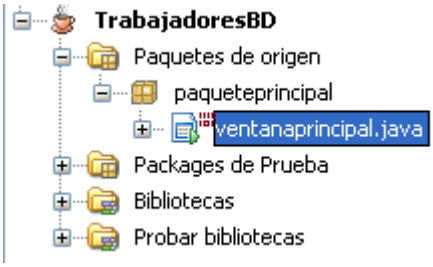
EJERCICIO GUIADO Nº 1

PLANTEAMIENTO

Se quiere realizar una aplicación de base de datos que nos muestre información sobre los trabajadores de la empresa MANEMPSA.

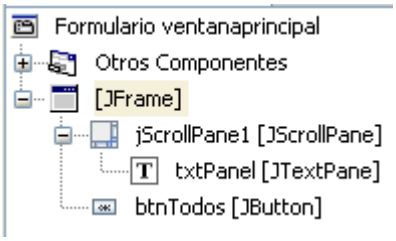
Esta aplicación le dará al usuario la capacidad de elegir la información que quiere extraer de la base de datos. Es decir, el usuario tendrá cierto control sobre las consultas que se realicen.

1. Entre en NetBeans. Crea un nuevo proyecto llamado *TrabajadoresBD*. Dentro de este proyecto crea un paquete principal llamado *paqueteprincipal* y dentro de él un JFrame llamado *ventanaprincipal*:



2. Añade a la ventana un JTextPane y un botón de momento:

El botón se llamará *btnTodos* y el JTextPane se llamará *txtPanel*.



3. Para que este programa pueda trabajar con la base de datos MANEMPSA tendrá que prepararlo haciendo lo siguiente:

    - Crear la subcarpeta Base y copiar en ella el fichero de base de datos MANEMPSA.MDB que tiene en la carpeta Mis Documentos.

    - Añadir al programa los objetos *conexión (Connection) y sentencia (Statement)* como globales.

    - Crear el procedimiento *PrepararBaseDatos* y llamarlo desde el constructor.

    - Cerrar la conexión desde el evento *windowClosing*

    Realice estos cuatro pasos que se han indicado antes de continuar.

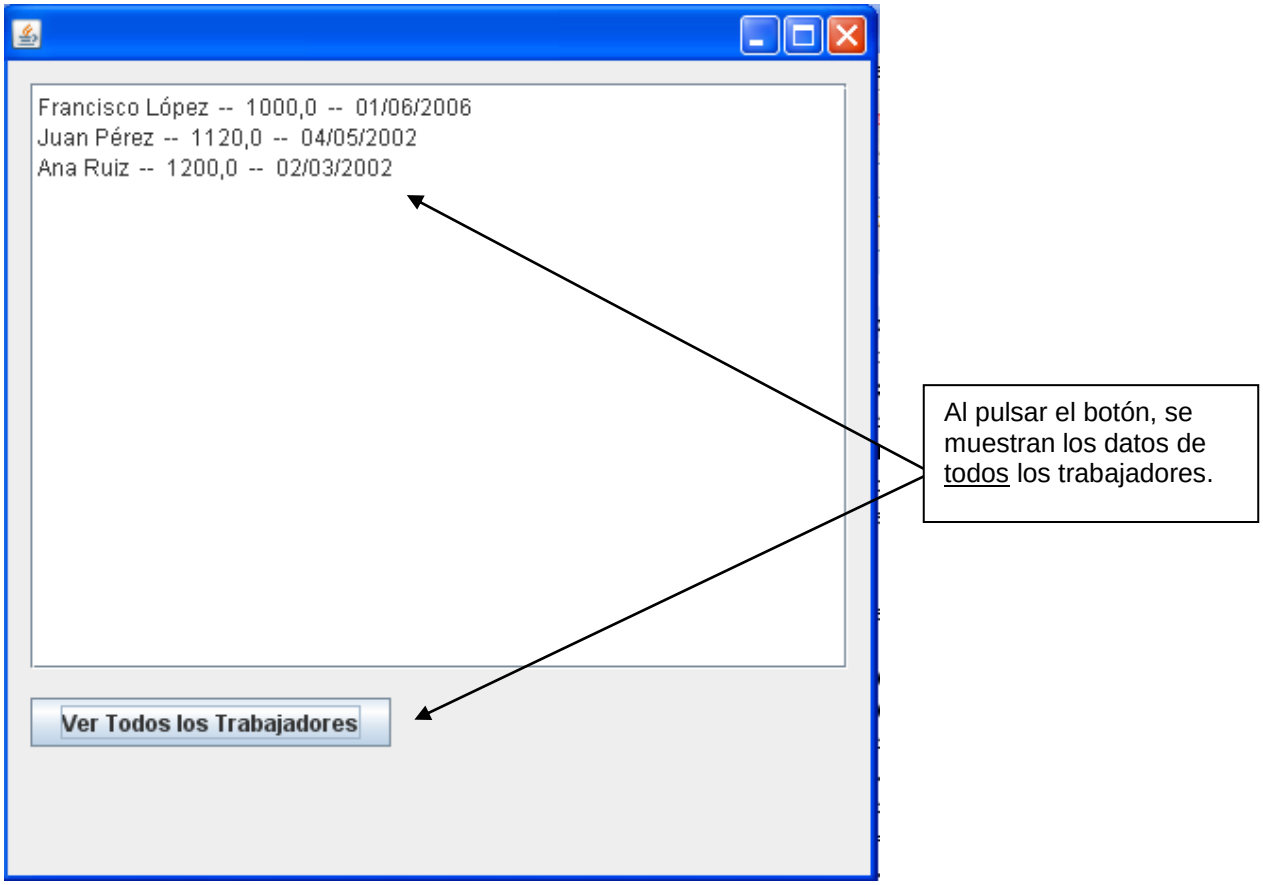
4. Ya se puede programar el botón *btnTodos.* Se pretende que al pulsar este botón aparezca en el panel *txtPanel* el contenido de la tabla *trabajadores*. Para ello, programe el siguiente código dentro del *actionPerformed* del botón *btnTodos*:

```
    private void btnTodosActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agrege su codigo aqui:
        String info="";
        String cadnombre; //cadena para el nombre
        String cadapell; //cadena para los apellidos
        String cadsueldo; //cadena para el sueldo
        String cadfecha; //cadena para la fecha

        try {
         ResultSet r=sentencia.executeQuery("select * from trabajadores order by sueldo");

         r.beforeFirst();
         while (r.next()) {
            cadnombre=r.getString("nombre");
            cadapell=r.getString("apellidos");
            cadsueldo=r.getString("sueldo");
            cadsueldo = cadsueldo.replace(".",",");
            cadfecha=r.getString("fecha");
            cadfecha=cadfecha.substring(8,10)+"/"+cadfecha.substring(5,7)+"/"+
                     cadfecha.substring(0,4);
            info=info+cadnombre+" "+cadapell+"  --  "+cadsueldo+"  --  "+cadfecha+"\n";
         }
         txtPanel.setText(""); //vacio el panel de texto
         txtPanel.setText(info); //meto la cadena info

        } catch(Exception e) {
            JOptionPane.showMessageDialog(null,"Error al consultar la tabla trabajadores");
        }
    }
```

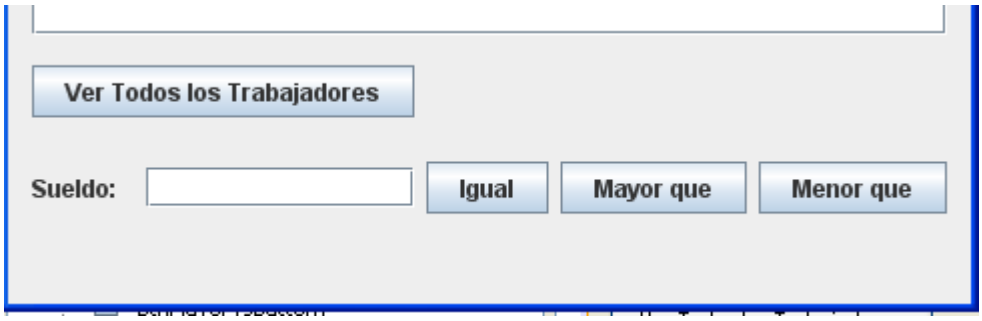
Si analiza este código, verá que es igual al que hemos realizado en hojas anteriores. Básicamente lo que hace es lo siguiente:

- Ejecuta la consulta "*`select * from trabajadores order by sueldo`*"
- Luego extrae del *ResultSet* los campos *nombre, apellidos, sueldo y fecha*.
- Hay que indicar que al campo *sueldo* se le cambia el punto decimal por la coma decimal.
- Por otro lado, la fecha se transforma para que tenga el formato dia/mes/año
- Finalmente se muestra el resultado de la consulta en el JTextPane: txtPanel. (Antes de mostrarlo se borra todo lo que hubiera en el panel)

5. Ejecuta el programa y prueba el funcionamiento de este botón:



6. Se va a mejorar el programa. Añada un cuadro de texto llamado *txtSueldo* y luego tres botones llamados respectivamente *btnMayor, btnMenor* y *btnIgual*. La parte inferior de la ventana quedará así:



7. Se pretende que estos botones funcionen de la siguiente forma:

   - El usuario introducirá un sueldo en el cuadro de texto *txtSueldo*.
   - Luego, si pulsa el botón *Igual*, aparecerá en el panel todos los trabajadores que tengan un sueldo igual al introducido.
   - En cambio, si pulsa el botón *Mayor*, aparecerá en el panel todos los trabajadores que tengan un sueldo mayor que el introducido.
   - Y si pulsa el botón *Menor*, aparecerá en el panel todos los trabajadores que tengan un sueldo menor que el introducido.

   Se empezará programando el botón *Igual*.

8. Programe en el *actionPerformed* del botón *btnIgual* lo siguiente. *(Nota: El código siguiente es prácticamente igual al anterior, solo se hace un pequeño cambio. Puede copiar y pegar y luego hacer la modificación que se indica)*

```
    private void btnIgualActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agrege su codigo aqui:
        String info="";
        String cadnombre; //cadena para el nombre
        String cadapell; //cadena para los apellidos
        String cadsueldo; //cadena para el sueldo
        String cadfecha; //cadena para la fecha

        try {
         String consulta;
         consulta = "select * from trabajadores where sueldo = "+txtSueldo.getText();
         ResultSet r=sentencia.executeQuery(consulta);

         r.beforeFirst();
         while (r.next()) {
            cadnombre=r.getString("nombre");
            cadapell=r.getString("apellidos");
            cadsueldo=r.getString("sueldo");
            cadsueldo = cadsueldo.replace(".",",");
            cadfecha=r.getString("fecha");
            cadfecha=cadfecha.substring(8,10)+"/"+cadfecha.substring(5,7)+"/"+
                     cadfecha.substring(0,4);
            info=info+cadnombre+" "+cadapell+"  --  "+cadsueldo+"  --  "+cadfecha+"\n";
         }
         txtPanel.setText(""); //vacio el panel de texto
         txtPanel.setText(info); //meto la cadena info

        } catch(Exception e) {
            JOptionPane.showMessageDialog(null,"Error al consultar la tabla trabajadores");
        }
    }
```

Estudiemos detenidamente el código remarcado:

```
String consulta;
consulta = "select * from trabajadores where sueldo = "+txtSueldo.getText();
ResultSet r = sentencia.executeQuery(consulta);
```

Aquí se crea una variable de texto llamada *consulta* y luego se concatena en ella la cadena:

```
select * from trabajadores where sueldo =
```

con

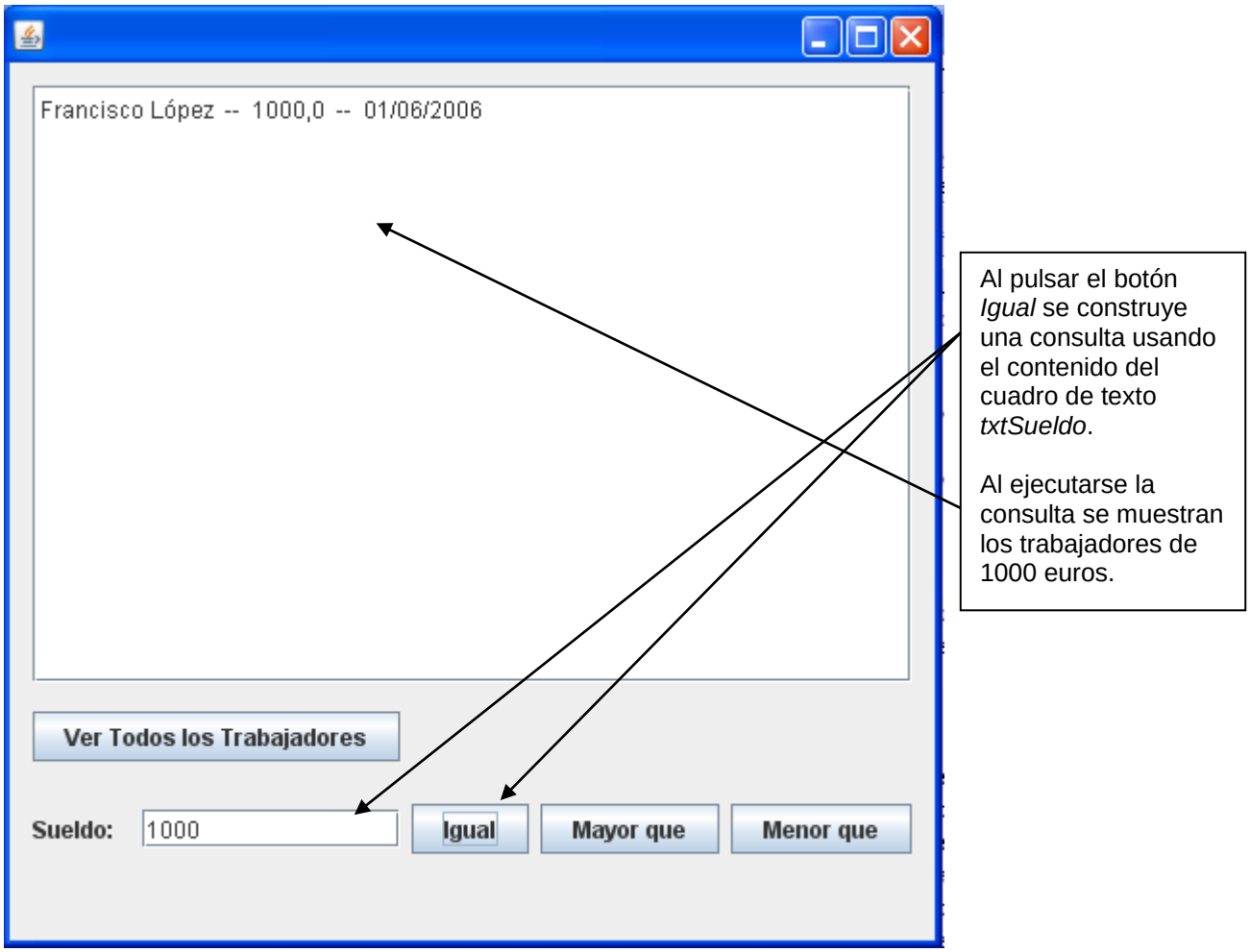
lo que contenga el cuadro de texto sueldo – es decir txtSueldo.getText()

Si el cuadro de texto del sueldo contuviera un 1000, entonces la cadena resultante sería:

```
select * from trabajadores where sueldo = 1000
```

Es decir, se construye una consulta que busca los sueldos de 1000 euros.

9. Prueba a ejecutar el programa. Escribe un valor 1000 en el cuadro de texto y luego pulsa el botón *Igual*. El resultado será que aparecen solo los trabajadores que tengan 1000 de sueldo.



10. Programa ahora el botón *Mayor que* de la siguiente forma (El código es prácticamente igual al anterior, así que puedes usar *copiar y pegar*. Se indica con una flecha la pequeña diferencia)

```
    private void btnMayorActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agrege su codigo aqui:
        String info="";
        String cadnombre; //cadena para el nombre
        String cadapell; //cadena para los apellidos
        String cadsueldo; //cadena para el sueldo
        String cadfecha; //cadena para la fecha

        try {
         String consulta;
         consulta = "select * from trabajadores where sueldo > "+txtSueldo.getText();
         ResultSet r=sentencia.executeQuery(consulta);

         r.beforeFirst();
         while (r.next()) {
            cadnombre=r.getString("nombre");
            cadapell=r.getString("apellidos");
            cadsueldo=r.getString("sueldo");
            cadsueldo = cadsueldo.replace(".",",");
            cadfecha=r.getString("fecha");
            cadfecha=cadfecha.substring(8,10)+"/"+cadfecha.substring(5,7)+"/"+
                     cadfecha.substring(0,4);
            info=info+cadnombre+" "+cadapell+"  --  "+cadsueldo+"  --  "+cadfecha+"\n";
         }
         txtPanel.setText(""); //vacio el panel de texto
         txtPanel.setText(info); //meto la cadena info

        } catch(Exception e) {
            JOptionPane.showMessageDialog(null,"Error al consultar la tabla trabajadores");
        }
    }
```

Como se puede observar, el código es igual. Simplemente cambia el operador en la cadena que se concatena. Ahora se usa un *mayor que*.

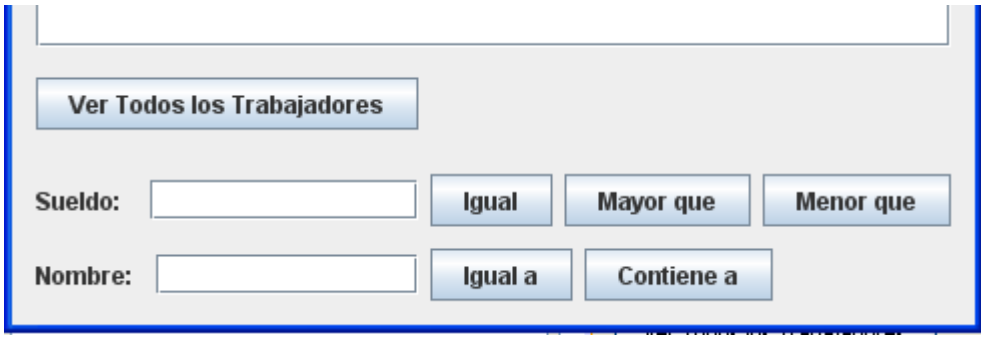
Es decir, si el usuario introdujera un 1000 en el cuadro de texto del sueldo, el resultado de la concatenación en la variable consulta sería la siguiente cadena:

```
select * from trabajadores where sueldo > 1000
```

11. Prueba a ejecutar el programa introduciendo el valor 1000 en el sueldo y luego pulsando el botón *Mayor que*. El resultado será que aparece en el panel de texto los trabajadores que cobran más de 1000 euros.

12. Como práctica, programe el botón *Menor que* de forma que muestre aquellos trabajadores que cobren menos de la cantidad introducida en el cuadro de texto *txtSueldo*.

13. Vamos a seguir mejorando el programa. Añada ahora el siguiente cuadro de texto y los siguientes botones:



El cuadro de texto se llamará *txtNombre* mientras que el botón "*Igual a* " se llamará *btnNombreIgual* y el botón "*Contiene a*" se llamará *btnContiene*.

14. Estos botones funcionarán de la siguiente forma:

    - El usuario introducirá un nombre en el cuadro de texto *txtNombre*.
    - Si luego pulsa el botón *Igual a*, entonces aparecerán todos aquellos trabajadores que tengan exactamente dicho nombre.
    - Si en cambio pulsa el botón *Contiene a*, entonces aparecerán todos aquellos trabajadores cuyo nombre contenga la palabra que se haya escrito en el cuadro de texto.

15. Empezaremos programando el botón *Igual a*. Para ello, escriba el siguiente código dentro del botón (este código es parecido a los anteriores, puede usar copiar y pegar y luego realizar las modificaciones pertinentes)

```
    private void btnNombreIgualActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agrege su codigo aqui:
        String info="";
        String cadnombre; //cadena para el nombre
        String cadapell; //cadena para los apellidos
        String cadsueldo; //cadena para el sueldo
        String cadfecha; //cadena para la fecha
        try {
         String consulta;
         consulta = "select * from trabajadores where nombre = '"+txtNombre.getText()+"'";
         ResultSet r=sentencia.executeQuery(consulta);

         r.beforeFirst();
         while (r.next()) {
            cadnombre=r.getString("nombre");
            cadapell=r.getString("apellidos");
            cadsueldo=r.getString("sueldo");
            cadsueldo = cadsueldo.replace(".",",");
            cadfecha=r.getString("fecha");
            cadfecha=cadfecha.substring(8,10)+"/"+cadfecha.substring(5,7)+"/"+
                     cadfecha.substring(0,4);
            info=info+cadnombre+" "+cadapell+"  --  "+cadsueldo+"  --  "+cadfecha+"\n";
         }
         txtPanel.setText(""); //vacio el panel de texto
         txtPanel.setText(info); //meto la cadena info

        } catch(Exception e) {
            JOptionPane.showMessageDialog(null,"Error al consultar la tabla trabajadores");
        }
    }
```

Observa la modificación del código. Puedes ver que la consulta SQL se consigue concatenando tres cadenas (se han puesto en color para facilitar la comprensión):

Primera cadena: `select * from trabajadores where nombre = '`
Segunda cadena: *`lo que contenga el cuadro de texto: txtNombre.getText()`*
Tercera cadena: `'`

Supongamos que el cuadro de texto contiene la palabra *Ana*, el resultado de la concatenación sería:

```
select * from trabajadores where nombre = 'Ana'
```

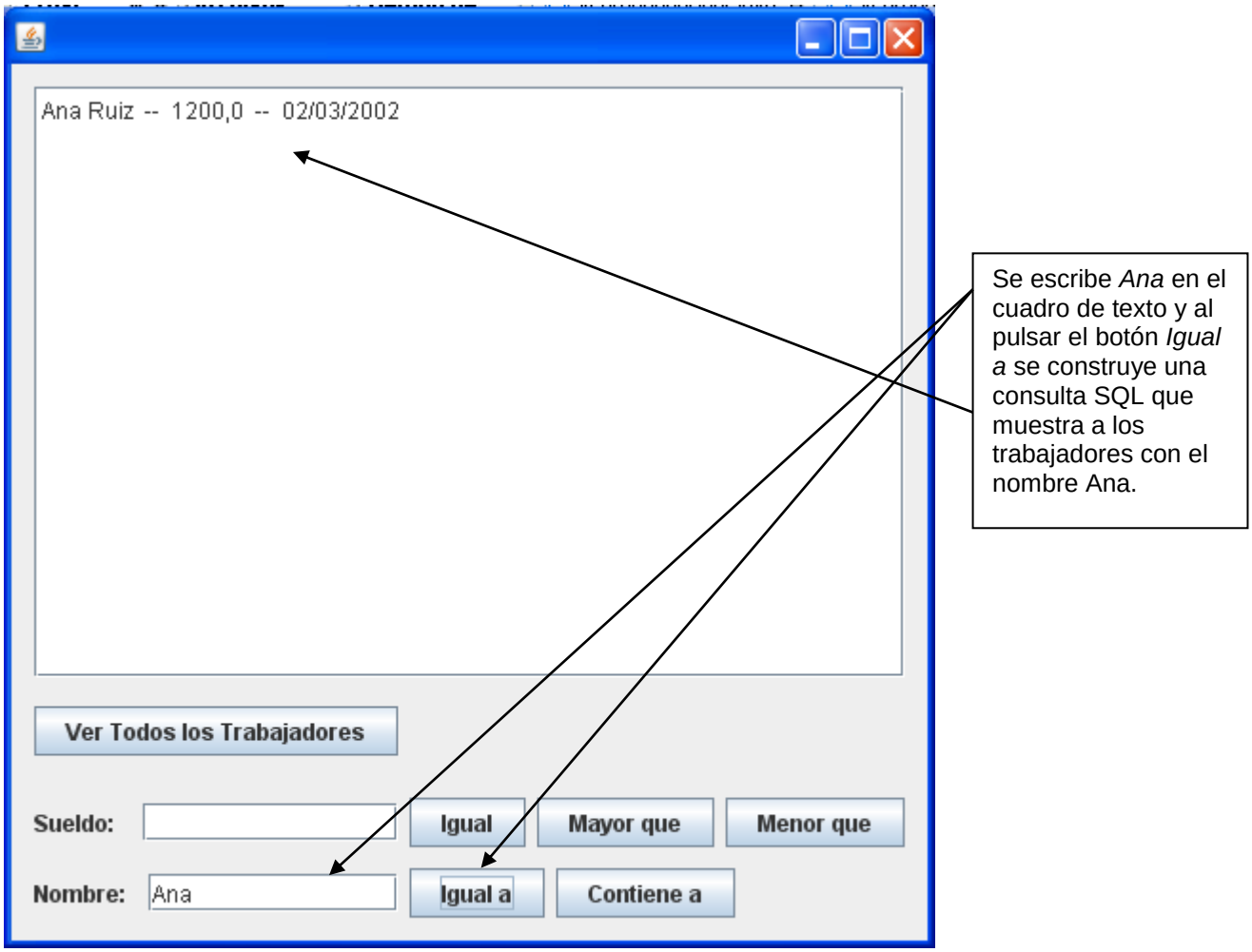
Es decir, el resultado de la concatenación sería una consulta SQL que muestra aquellos trabajadores que tengan de nombre Ana.

NOTA:

Recuerda que cuando se especifica un valor de tipo texto en una consulta SQL, es necesario que esté rodeado de comillas simples (‘)

Esa es la razón por la que se tienen que concatenar dos ‘

16. Ejecuta el programa y prueba a escribir en el cuadro de texto del nombre Ana. Luego pulsa el botón *Igual a*…



17. Ahora programaremos el botón *Contiene a* de forma que el usuario escriba un texto en el cuadro del nombre, y al pulsar el botón *Contiene a* aparezcan todos aquellos trabajadores cuyo nombre contenga el texto escrito.

    Por ejemplo, si el usuario introduce el texto "*an*" en el cuadro, al pulsar *Contiene a* aparecerán los trabajadores que se llamen: *Ju**an**, **An**tonio, **An**tonia, M**an**olo, **An**a, etc* (todos contienen el texto 'an')

18. Para ello programe lo siguiente en el *actionPerformed* del botón *Contiene a* (es un código muy parecido al anterior, solo tiene que realizar una pequeña modificación)

```
    private void btnContieneActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agrege su codigo aqui:
        String info="";
        String cadnombre; //cadena para el nombre
        String cadapell; //cadena para los apellidos
        String cadsueldo; //cadena para el sueldo
        String cadfecha; //cadena para la fecha
        try {
         String consulta;
         consulta = "select * from trabajadores where nombre like '%"+txtNombre.getText()+"%'";
         ResultSet r=sentencia.executeQuery(consulta);

         r.beforeFirst();
         while (r.next()) {
            cadnombre=r.getString("nombre");
            cadapell=r.getString("apellidos");
            cadsueldo=r.getString("sueldo");
            cadsueldo = cadsueldo.replace(".",",");
            cadfecha=r.getString("fecha");
            cadfecha=cadfecha.substring(8,10)+"/"+cadfecha.substring(5,7)+"/"+
                     cadfecha.substring(0,4);
            info=info+cadnombre+" "+cadapell+"  --  "+cadsueldo+"  --  "+cadfecha+"\n";
         }
         txtPanel.setText(""); //vacio el panel de texto
         txtPanel.setText(info); //meto la cadena info

        } catch(Exception e) {
            JOptionPane.showMessageDialog(null,"Error al consultar la tabla trabajadores");
        }
    }
```

En este caso, para crear la consulta se concatenan tres cadenas (se indican en color para facilitar la comprensión)

Primera cadena: `select * from trabajadores where nombre like '%`
Segunda cadena: *`lo que contenga el cuadro de texto: txtNombre.getText()`*
Tercera cadena: `%'`

Supongamos que escribimos en el cuadro de texto del nombre la palabra *an*, el resultado de la concatenación sería el siguiente:

```
select * from trabajadores where nombre like '%an%'
```

La condición *`nombre like '%an%'`* significa que contenga la palabra *an*.

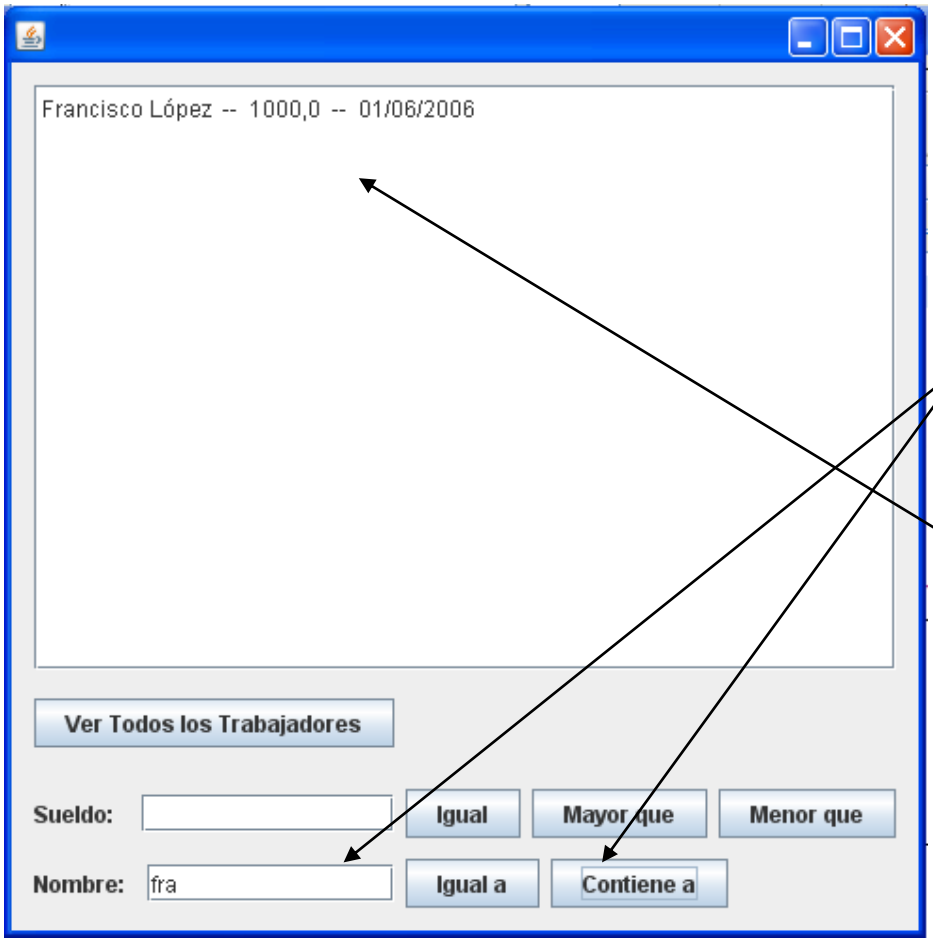
NOTA:

Cuando se estudiaron las consultas SQL, se vio que el operador *like* funcionaba a través de asteriscos. Es decir, la forma "correcta" de indicar la condición anterior sería la siguiente:

`nombre like '*an*'`

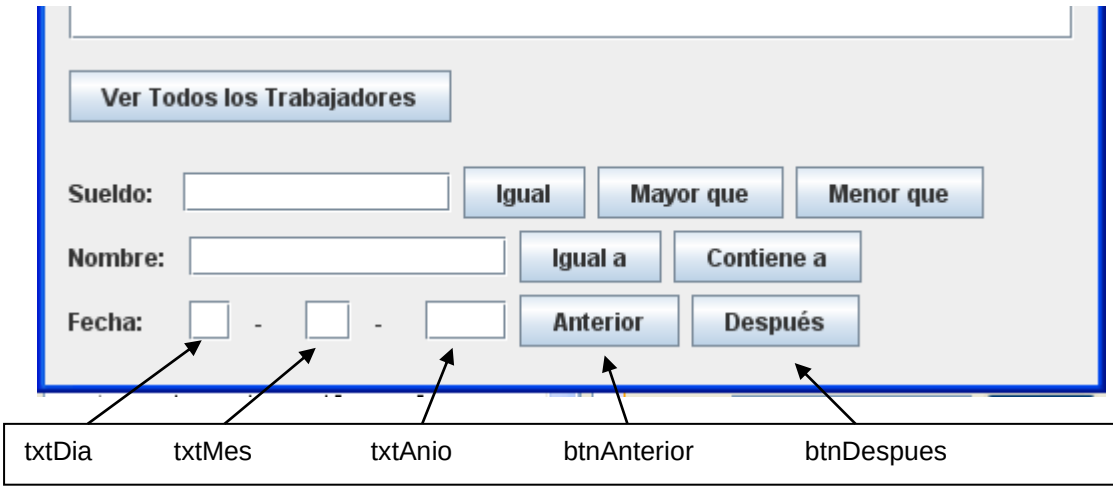
Sin embargo, hay que decir que el asterisco se debe usar solamente cuando estamos manipulando una base de datos de Access desde dentro. En el caso de que queramos acceder a ella desde una aplicación java se usarán porcentajes % en vez de asteriscos.

19. Ejecuta el programa y prueba a escribir en el cuadro de texto del nombre el texto *fra*. Luego pulsa el botón *Contiene a* y comprueba el resultado. Deben aparecer todos aquellos trabajadores cuyo nombre contenga el texto *fra.* Por ejemplo, *Francisco*.



Se escribe *fra*, se pulsa el botón *Contiene a*, y entonces el programa construye una consulta que muestra los trabajadores cuyo nombre contenga *fra*.

20. Sigamos mejorando el programa. Añade estos cuadros de texto y estos botones:



21. El objetivo de estos elementos añadidos es el siguiente:

    - El usuario introducirá una fecha (día, mes y año) en los cuadros de texto *txtDia, txtMes, txtAnio*.

    - Luego, si pulsa el botón *Anterior*, aparecerán los trabajadores que hayan entrado en la empresa antes de la fecha indicada.

    - Si pulsa el botón *Después*, en cambio, aparecerán los trabajadores que hayan entrado en la empresa después de la fecha indicada.

22. Empezaremos programando el botón *Anterior*. Accede a su *actionPerformed* e incluye el siguiente código (es un código similar al anterior, solo tienes que realizar un pequeño cambio):

```
    private void btnAnteriorActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agrege su codigo aqui:
        String info="";
        String cadnombre; //cadena para el nombre
        String cadapell; //cadena para los apellidos
        String cadsueldo; //cadena para el sueldo
        String cadfecha; //cadena para la fecha
        try {
         String consulta;
         consulta = "select * from trabajadores where fecha < #";
         consulta = consulta + txtMes.getText()+"-"+txtDia.getText()+"-"+txtAnio.getText();
         consulta = consulta + "#";
         ResultSet r=sentencia.executeQuery(consulta);

         r.beforeFirst();
         while (r.next()) {
            cadnombre=r.getString("nombre");
            cadapell=r.getString("apellidos");
            cadsueldo=r.getString("sueldo");
            cadsueldo = cadsueldo.replace(".",",");
            cadfecha=r.getString("fecha");
            cadfecha=cadfecha.substring(8,10)+"/"+cadfecha.substring(5,7)+"/"+
                     cadfecha.substring(0,4);
            info=info+cadnombre+" "+cadapell+"  --  "+cadsueldo+"  --  "+cadfecha+"\n";
         }
         txtPanel.setText(""); //vacio el panel de texto
         txtPanel.setText(info); //meto la cadena info
        } catch(Exception e) {
            JOptionPane.showMessageDialog(null,"Error al consultar la tabla trabajadores");
        }
    }
```

Presta mucha atención al código remarcado. En él, se construye una consulta que permite mostrar aquellos trabajadores cuya fecha de entrada en la empresa sea anterior a la indicada en los cuadros de texto.

Esto se consigue concatenando varias cadenas (se indican a continuación en distintos colores para facilitar la comprensión)

Primera cadena: `select * from trabajadores where fecha < #`
Segunda cadena: *lo que contiene el cuadro de texto del mes: txtMes.getText()*
Tercera cadena: `/`
Cuarta cadena: *lo que contiene el cuadro de texto del día: txtDia.getText()*
Quinta cadena: `/`
Sexta cadena: *lo que contiene el cuadro de texto del año: txtAnio.getText()*
Séptima cadena: `#`

Por ejemplo, supongamos que:

- en el cuadro *txtDia* se introdujo un 20.
- en el cuadro *txtMes* se introdujo un 12.
- En el cuadro *txtAnio* se introdujo un 2005.

Entonces, la cadena resultante de la concatenación será:

```
select * from trabajadores where fecha < #12/20/2006#
```

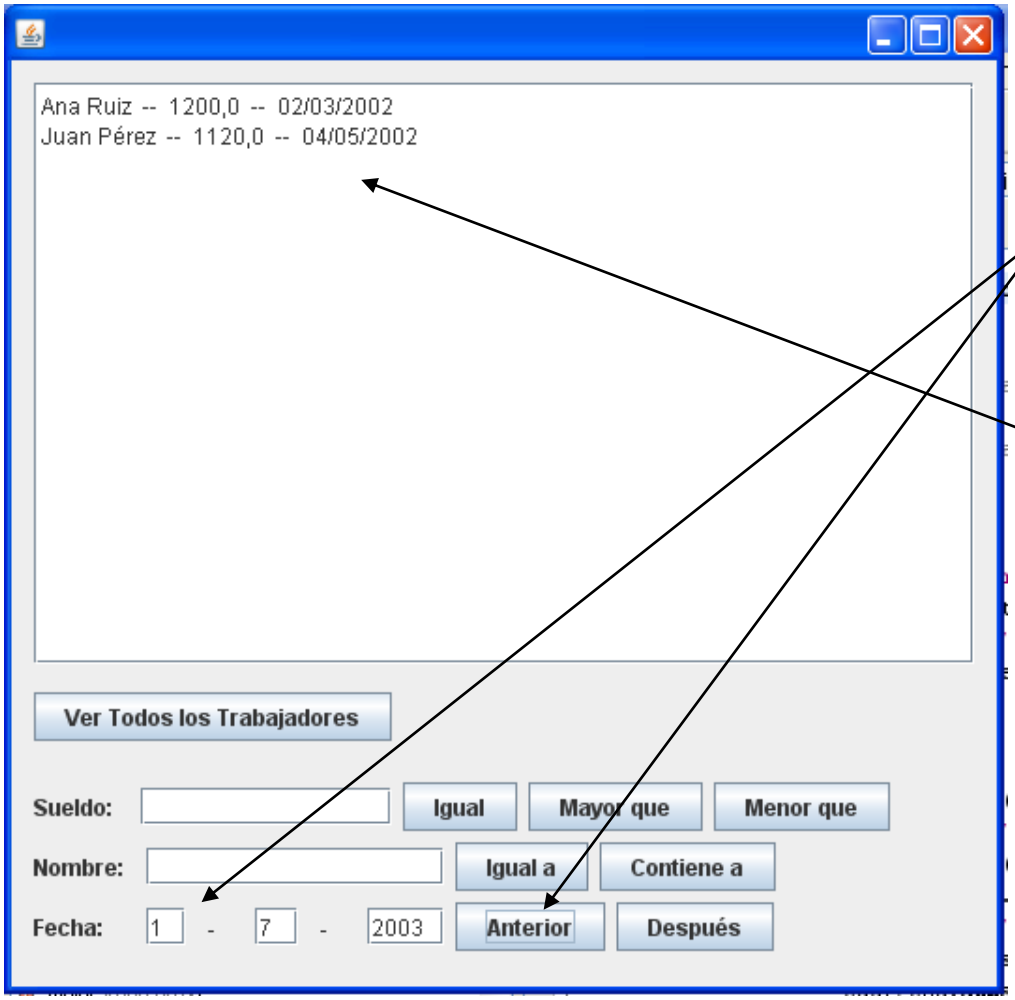
Es decir, la cadena resultante es una consulta SQL que busca los trabajadores cuya fecha de entrada en la empresa sea anterior al 20 del 12 del 2006

NOTA:

Recuerda que para que Access entienda las fechas al hacer una consulta SQL, es necesario indicarlas en el formato `mes/dia/año`.

Por otro lado, recuerda que las fechas deben estar rodeadas por almohadillas #

23. Ejecuta el programa y prueba a introducir una fecha en las casillas correspondientes. Luego pulsa el botón *Anterior* y observa como aparecen los trabajadores que entraron antes de la fecha indicada.



Introduce un día, mes y año en los cuadros correspondientes.

Al pulsar el botón *Anterior*, el programa mostrará los trabajadores que hayan entrado antes de esa fecha.

24. Como ejercicio se propone que programe el botón *Después*. Al pulsar este botón, se mostrarán los trabajadores que hayan entrado antes de la fecha indicada en los cuadros de texto. El código es prácticamente igual al código del botón *Anterior*.

CONCLUSIÓN

A través del objeto *sentencia* podemos ejecutar una consulta SQL en una base de datos.

Esta consulta SQL viene expresada como una cadena.

Se puede construir una consulta SQL a través de la concatenación de varias cadenas. Estas cadenas pueden ser datos introducidos por el usuario en cuadros de textos u otros controles.

Al hacer que el usuario pueda participar en la construcción de la consulta aportando sus propios datos, le damos la posibilidad de que él decida la información que se quiere extraer de la base de datos. De esta manera se consigue que el programa sea más potente.

A tener en cuenta lo siguiente acerca de las consultas SQL ejecutadas desde una aplicación java para acceder a una base de datos de Access:

- Los valores tipo texto se indicarán entre comillas simples ‘
- El operador *like* se usa con porcentajes (%) en vez de con asteriscos (*)
- Las fechas van rodeadas por almohadillas (#)
- Las fechas tienen que indicarse con el formato mes/dia/año