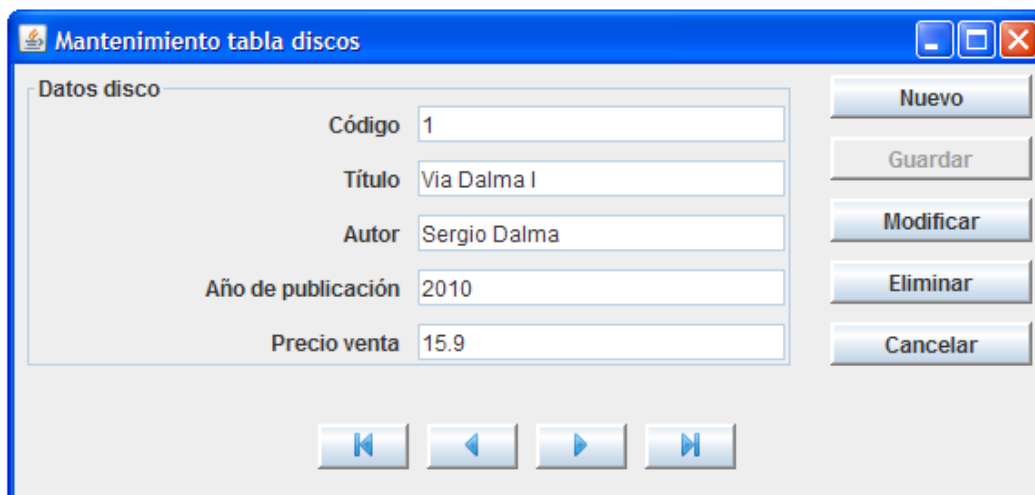


Práctica 18: Bases de datos

- 1.- Crea una base de datos en MySQL llamada **Discrografica**, añade a dicha editorial el usuario '**admin**', con clave '**1234**', y permiso total sobre la base de datos.
- 2.- Desarrollar un programa que permita crear una tabla llamada '**Discos**' en la base de datos '**Discografica**', La información que almacenará esta tabla será código, nombre de disco, autor del mismo, año de publicación y precio de venta al público.

Asimismo, también se añadirán los datos que encontrarás en el fichero de texto discos.txt, a la citada tabla. La estructura del fichero discos.txt, es un registro por línea de fichero, y cada campo del registro estará separado por un *.

- 3.- Desarrollar un interfaz gráfico para el mantenimiento y consulta de la tabla discos. Para ello seguiremos el Modelo-Vista-Controlador. El aspecto de la aplicación podrá ser similar al de la siguiente figura:



Se sugiere seguir los siguientes pasos:

- Empezar por desarrollar el **modelo**, para ello:
 - Desarrollar la clase Disco, esta clase almacenará la información de un disco (código, título, autor, año publicación y precio de venta). Proporcionará los siguientes métodos: constructor, get's y toString.
 - Desarrollar la clase Modelo, dicha clase tendrá como variables de instancia:

```
private Connection con; // Objeto con la conexión a la BD
private Statement stmt; // Objeto que permite ejecutar sentencias SQL
private ResultSet rs; // Resultados de la consulta
```

y como métodos:

- ✓ Constructor: se encargará de dar valor a nuestras variables de instancia, es decir, establecer la conexión, crear el objeto Statement (se aconseja que sea bidireccional, que refleje los cambios de la BD, y que el ResultSet sea actualizable), y por último crear la consulta.
- ✓ Método que establezca la conexión con la base de datos.
- ✓ Método que cierre la conexión con la base de datos.
- ✓ Método que cierre el "Statement".

- ✓ Método que cree la consulta de la base de datos.

```
public void creaConsulta() throws SQLException
{
    String sqlString = "SELECT * FROM discos";
    rs = stmt.executeQuery(sqlString);
}
```

- ✓ Método que devuelve el primer registro de la base de datos (*first*).

```
public Disco getPrimero() throws SQLException
```

- ✓ Método que devuelve el último registro de la base de datos (*last*).

- ✓ Método que devuelve el siguiente registro de la base de datos (*next*).

- ✓ Método que devuelve el registro anterior de la base de datos (*previous*).

- ✓ Método que añade un nuevo disco a la base de datos.

```
public void insertaDisco(Disco ob) throws SQLException
```

- ✓ Método que permite modificar un disco en la base de datos.

- ✓ Método que añade un nuevo disco a la base de datos.

- ✓ Método que elimina un disco de la base de datos cuyo código llega como parámetro.

- ✓ Método que muestre una información más detallada sobre una excepción tipo SQLException.

```
public void printSQLException(SQLException ex)
```

- Hacer una clase que pruebe el funcionamiento en modo consola de nuestro modelo.

- Una vez que tenemos el modelo, vamos a continuar por desarrollar la **vista**, cuyo aspecto, ya vimos anteriormente, aunque se puede cambiar a gusto del alumno. Tan sólo mencionar que cuando vayamos a asignar control a los componentes, se aconseja, por separar el funcionamiento posteriormente en el controlador, dar control de ratón a los botones de navegación (primero, siguiente, ...), y control de acción a los botones de mantenimiento (nuevo, eliminar, ...)

- Por último sólo nos queda hacer el **controlador**, cuyo funcionamiento será el siguiente:

- Al cargar la aplicación los botones estarán todos activos excepto el botón “Guardar”, tal y como se muestra en la figura de la página anterior. Asimismo los JTextField donde se muestra la información deberán estar “no editables”.

- Al pulsar sobre los botones de navegación, se seguirá el siguiente comportamiento



Al pulsar sobre este botón se mostrará el primero de los registros de nuestra tabla discos.



Al pulsar sobre este botón se mostrará el registro anterior al actual.



Al pulsar sobre este botón se mostrará el registro siguiente al actual.



Al pulsar sobre este botón se mostrará el último de los registros de nuestra tabla discos.

- Si pulsamos sobre el botón **Nuevo**, se borrará el contenido de todos los JTextField, se pondrán editables, para poder introducir la información del nuevo disco, y sólo quedarán activos los botones “Guardar” y “Cancelar”.

- El botón “**Guardar**”, o “**Guardar cambios**”, nos permitirá recoger los datos del formulario en el caso de un nuevo disco o en el caso de una modificación.
 - Si estamos en el caso de un alta, recogeremos todos los datos del formulario, y añadiremos un nuevo registro a la tabla discos.
 - Si estamos en el caso de una modificación, habremos cambiado previamente el texto al botón “Guardar”, poniendo “Guardar cambios”, recogeremos todos los datos del formulario, excepto el código, y modificaremos dicho registro con la nueva información en nuestra tabla discos.

En ambos casos, al terminar la operación correspondiente, lanzaremos un mensaje indicando, “Disco almacenado” o “Disco modificado”, según corresponda, y por último reiniciaremos el formulario, es decir, volveremos a la situación de partida (todos los botones habilitados excepto el de guardar, los JTextField no editables, y se visualizará el primer registro).

- El botón “**Modificar**”, nos permitirá modificar el registro que se está visualizando, al pulsar el botón, se convertirán en editables todos los JTextField, excepto el del campo código, y solo se dejarán activos los botones “Guardar Cambios” (previamente le habremos cambiado la etiqueta al botón Guardar) y “Cancelar”.
- El botón “**Eliminar**”, nos permitirá eliminar el registro que se está visualizando, al pulsar sobre el botón nos aparecerá un dialogo de confirmación de borrado, si la respuesta a este dialogo es afirmativa, entonces se borrará el registro en cuestión, si se cancela el borrado, todo se quedará como estaba antes de haber pulsado el botón Eliminar.
- Por último el botón “**Cancelar**”, se encargará de reiniciar el formulario, es decir, volverlo a la situación de partida.

Para finalizar se propone que para cerrar la conexión a la base de datos, y el objeto Statement, un buen momento podría ser cuando se cierra la aplicación (al pulsar el botón de cierre).