

Grado Superior en Informática

Acceso a Datos

Unidad 6: Programación de
componentes de acceso a datos

Tarea online 6



Autor: Raúl Serrano Torres.

1. Creación de componente de acceso a datos.

Para este primer apartado se ha creado una librería que permitirá acceder a la base de datos `modulosnf.sql` y recuperar la información que contiene. Dicha librería se compone de dos clases:

- `Modulo.java`: Clase que se corresponde con la tabla `modulos` de la base de datos.
- `VerRegistros.java`: Clase que consta de métodos para acceder a la base de datos y realizar consultas a la misma.

1.1. `Modulo.java`

Esta clase se crea con el fin de permitir crear objetos tipo `Modulo` que se corresponderán con las tuplas que devolverá la consulta realizada a la base de datos. Por ello, esta clase cuenta con 4 propiedades o atributos, que son los mismos que los de la tabla `módulos`:

- `siglasCiclo` (`String`).
- `siglasModulo` (`String`).
- `nombreModulo` (`String`).
- `horasSemanales` (`int`).

También cuenta con los **getters** y **setters** correspondientes a sus atributos.

1.2. `VerRegistros.java`

Esta clase es la encargada de conectar la aplicación a la base de datos `modulosnf`. Consta de 3 propiedades:

- **`bd`** (`String`): cuyo valor fijo es el nombre de la base de datos (`modulosnf` en este caso).
- **`url`** (`String`): correspondiente a la url que se usará para la conexión al servidor de la base de datos.
- **`conn`** (`Connection`): objeto tipo conexión que se servirá para conectarse a la base de datos y poder realizar consultas. Si su valor es `null`, indica que la base de datos no está conectada.

Para poder realizar la conexión a la base de datos y realizar las consultas, esta clase cuenta con una serie de métodos que veremos a continuación:

- **boolean conectarBD(String login,String password):** este método recibe como parámetros dos cadenas correspondientes al usuario y la contraseña necesarios para conectarse a la base de datos. Cuando se invoca este método, la propiedad **conn** se inicializará almacenando la conexión a la base de datos. Esta acción puede lanzar una excepción tipo **SQLException** o **ClassNotFoundException** que imprimirán por pantalla el error ocurrido.
En caso de que no ocurra ninguna excepción, el método retornará el valor que devuelve el método **comprobarConexion**, devolviendo **true** si **conn** se ha inicializado correctamente o **false** si ha habido algún problema y su valor sigue siendo **null**.
- **void desconectarBD():** desconecta la base de datos estableciendo el valor de **conn** a **null**.
- **boolean comprobarConexion():** devuelve **true** si **conn** se ha inicializado correctamente o **false** y su valor sigue siendo **null**, lo cual indica que la base de datos esta desconectada.
- **ArrayList<Modulo> obtenerContenidoBD():** devuelve un **ArrayList** de **modulos** que contiene la información de toda la tabla **modulos** de la base de datos. Esta información se consigue ejecutando la siguiente consulta:

SELECT *

FROM modulos

Cada tupla de la base de datos se guardará en un objeto **Modulo** que se añadirá al **ArrayList** que devuelve el método.

Además de estos métodos, para que esta clase funcione correctamente se ha añadido la librería que permite cargar el driver MySQL (**MySQL JDBC Driver**).

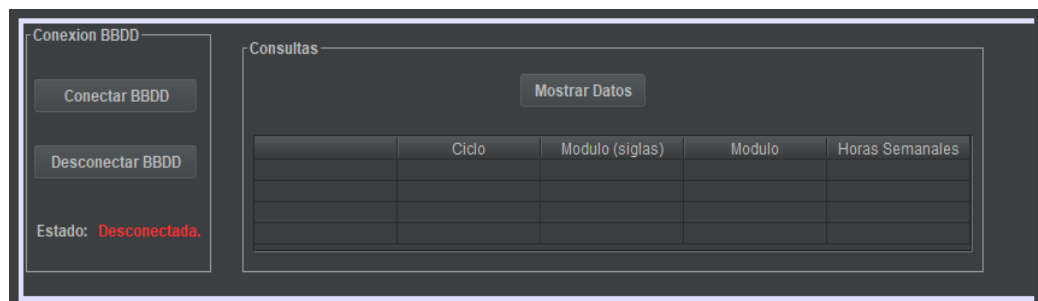
2. Programa de prueba de la librería.

En este apartado veremos una aplicación basada en Swing que pondrá a prueba la librería creada anteriormente. Dicha librería se encuentra en la carpeta **lib** que hay dentro del proyecto. En ella se guardan tanto la librería correspondiente a **MySQL** como la nueva librería llamada **VerRegistros**.

La aplicación consta de un único fichero que contiene la interfaz del programa llamado **VentanaPrincipal.java**.

2.1. VentanaPrincipal.java.

La interfaz de la aplicación presenta el siguiente aspecto.



En ella podemos ver una serie de botones para conectarse y desconectarse de la base de datos, así como para mostrar los datos de la base de datos, un indicador del estado de la conexión a la base de datos y una tabla donde se mostrará el contenido de la consulta realizada.

La clase ventana principal tan solo tiene una propiedad:

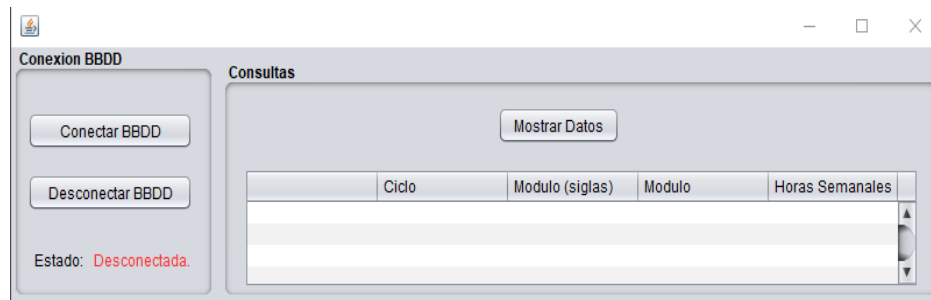
- **VerRegistros conexion:** objeto de la clase VerRegistros.java que permitirá a la aplicación conectar la base de datos, así como obtener información de la misma.

Además, cuenta con un único método cuya función es cargar los datos de la consulta a realizar en la tabla.

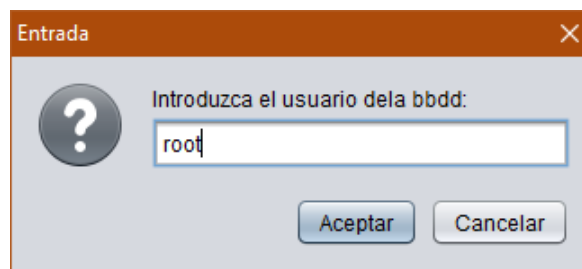
- **void cargarTabla(ArrayList<Modulo> lista):** este método recibe una lista con una serie de objetos Modulo y los carga en la tabla de la interfaz añadiendo una columna correspondiente al número de la fila. Cada modulo se corresponde con una fila de la tabla.

2.2. Funcionamiento de la aplicación.

Al ejecutar la aplicación con encontraremos la interfaz de la siguiente forma:

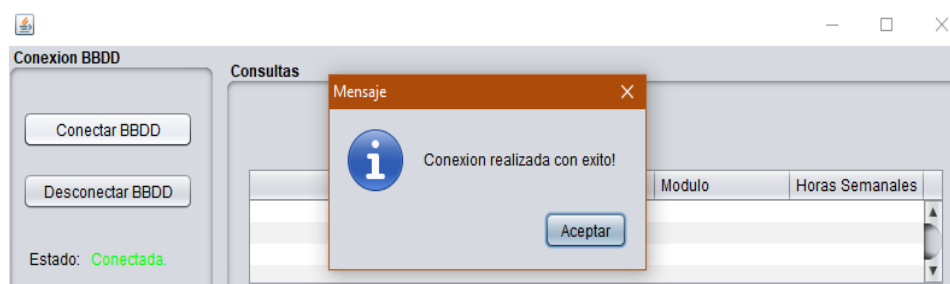


Como se puede observar, la tabla está vacía y el indicador de la parte inferior izquierda nos dice que la base de datos no está conectada. Para conectarla tendremos que presionar el botón **Conectar BD**. Al hacerlo, aparecerán dos ventanas que solicitarán información. La primera pedirá el nombre del usuario (en este caso “root”) que accederá a la base de datos y la segunda la contraseña (en este caso una cadena vacía).

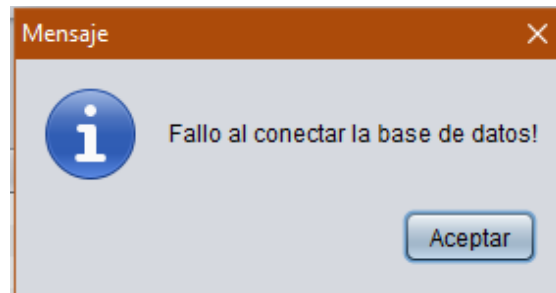


Estos datos se guardarán en variables que serán usadas como parámetros para invocar el método **conectarBD** del objeto **VerRegistros** que será llamado al introducir la contraseña. Al hacerlo, dicho método conectará la aplicación a la base de datos.

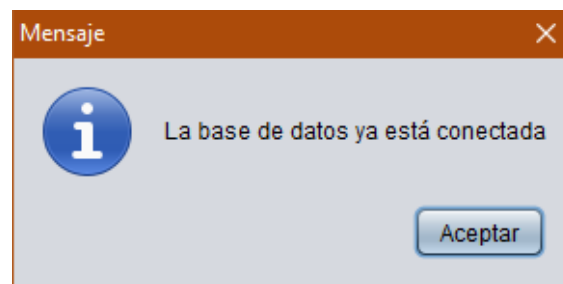
Si los datos han sido introducidos correctamente la aplicación mostrará un mensaje confirmando que la base de datos se ha conectado correctamente (si conectarBD devuelve true) y el estado de la conexión cambiará a **Conectada**.



En caso contrario mostrará un mensaje indicando que ha habido un error (si conectarBD devuelve false).



Si la base de datos ya ha sido conectada y se pulsa de nuevo este botón, aparecerá el siguiente mensaje.



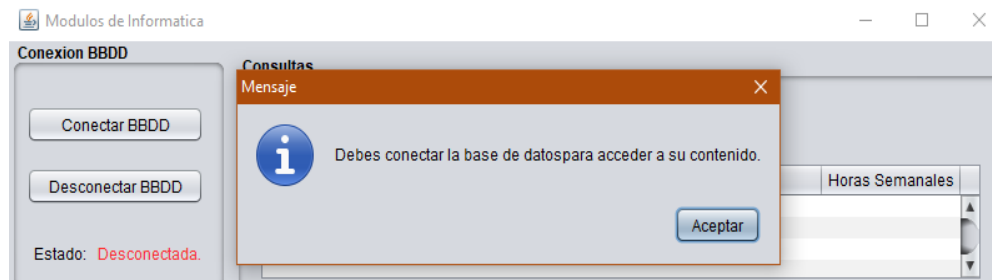
Esto es gracias a que antes de invocar al método **conectarBD** se comprueba si la base de datos está conectada llamando al método **comprobarConexion**.

```
private void btnConectarActionPerformed(java.awt.event.ActionEvent evt) {  
    if(conexion.comprobarConexion())  
        JOptionPane.showMessageDialog(rootPane, "La base de datos ya está conectada");  
    else{  
        String usuario = JOptionPane.showInputDialog("Introduzca el usuario de"  
            + " la bbdd:");  
        String contraseña = JOptionPane.showInputDialog("Introduzca la contraseña:");
```

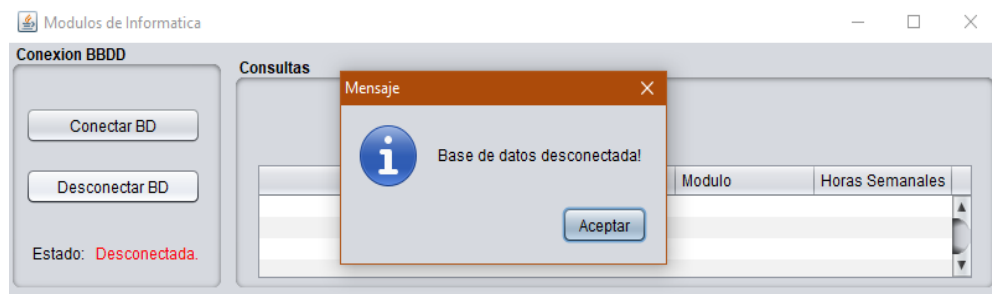
Una vez conectada, podemos proceder a mostrar los datos en la tabla pulsando el botón **Mostrar Datos**. Este botón llama al método **cargarTabla** pasándole como parámetro un objeto `ArrayList<Modulo>` resultado de llamar al método **obtenerContenidoBD** del objeto conexión. Al hacerlo, se mostrarán los datos de la consulta en la tabla.

Consultas				
<div>Mostrar Datos</div>				
	Ciclo	Modulo (siglas)	Modulo	Horas Semanales
0	DAM	AD_2021	Acceso a Datos	1
1	DAM	DI_2021	Desarrollo de Int...	2
2	DAW	DWES_2021	Desarrollo Web e...	2

Si se intentan mostrar los datos sin conectar la base de datos, aparecerá el siguiente mensaje de error. Esto es debido a que antes de tratar de obtener los datos, se comprueba si la base está conectada llamando al método **comprobarConexion** del objeto **conexión**.



El botón **Desconectar BD** desconectará la base de datos invocando el método **desconectarBD** del objeto **conexión**. Si la base de datos ha sido anteriormente conectada, aparecerá el siguiente mensaje indicando que esta ha sido desconectada. Además, el estado de la conexión cambiará a “Desconectada”.



Si la base de datos no está conectada y se intenta desconectar, aparecerá el siguiente mensaje. De nuevo, la aplicación llama al método **comprobarConexion** del objeto **conexión** para realizar la comprobación.

