

Diseño y Desarrollo de Sistemas de Información

Grado en Ingeniería Informática

Práctica 1

Conexión a una BD mediante JDBC

Curso 2021/22
(actualizado el 13 de octubre de 2021)

Duración estimada: 1 sesión (2 horas)

OBJETIVOS

- Conocer los conceptos fundamentales de la API JDBC para realizar una conexión a una base de datos relacional (Oracle, MySQL, etc.)
- Aprender a realizar una aplicación en lenguaje *Java* capaz de conectar con una base de datos relacional

Ejercicio 1

- Crear un proyecto Maven con NetBeans en el que se incluyan las dependencias con los artefactos necesarios (drivers) para conectarse a Oracle mediante JDBC.

En este proyecto se crearán los paquetes "Modelo", "Vista", "Controlador" y "Aplicación".

Ejercicio 2

- Añadir al proyecto un fichero Java en el paquete "Modelo" con una clase llamada "**Conexion**" que tendrá los siguientes atributos y métodos:
 - Un atributo de tipo **Connection**
 - Un constructor **Conexion()**. Realiza una conexión a la base de datos RABIDA con el nombre de usuario y contraseña asignado
 - Un constructor **Conexion(String sgbd, String ip, String bd, String usuario, String password)**. Realiza una conexión con los siguientes parámetros:
 - **sgbd**. Nombre del SGBD al que se quiere conectar (Oracle, MySQL, etc.)
 - **ip**. Dirección ip en la que se encuentra el servidor
 - **bd**. Nombre de la base de datos a la que se quiere conectar (para Oracle, este parámetro será la cadena vacía)
 - **usuario y password**. Nombre y clave del usuario con el que se realiza la conexión
 - El método **void desconexion()**. Este método debe cerrar la conexión con la base de datos
 - Añadir al proyecto un fichero Java en el paquete "Vista" con una clase llamada "**VistaConsola**" que tendrá los siguientes métodos:
 - **void vistaConsolaLogin(String texto)**. Muestra por pantalla el contenido del parámetro "texto"
 - **void vistaConsolaLogin(String texto, String error)**. Muestra por pantalla el contenido de los parámetros "texto" y "error"
-

- Añadir al proyecto un fichero Java en el paquete "Controlador" con una clase llamada "**ControladorLogin**" que tendrá los siguientes atributos y métodos:
 - Un atributo de tipo "Conexion"
 - Un constructor **CotroladorLogin()**

Para comprobar el funcionamiento de esta práctica, el constructor deberá llamar a una función para conectar con la base de datos y a otra para realizar la desconexión, las cuales llamarán, a su vez, al método constructor de **Conexion** y al método **desconexión()**, respectivamente. En estas funciones se deben capturar las excepciones de tipo *SQLException* mediante bloques *try-catch*.

- Añadir al proyecto un fichero Java en el paquete "Aplicacion" con la clase principal. Lo único que hará la aplicación será crear un objeto de tipo "ControladorLogin"

Ejercicio 3 (Opcional)

La clase **Connection** tiene un método, llamado **getMetaData()**, que devuelve los metadatos de la conexión en un objeto de la clase **DatabaseMetaData**. Esta clase tiene un gran número de métodos para recuperar los metadatos. Entre ellos **getDriverName()**, **getDriverVersion()**, etc.

- Añadir a la clase **Conexion** un método llamado **DatabaseMetaData informacionBD()**, que devuelva los metadatos de la conexión.
- Añadir a la clase **VistaConsola** un método llamado **void vistaMetadatos(DatabaseMetaData dbmd)**. Esta vista mostrará la siguiente información:
 - Versión de la base de datos
 - URL de la conexión
 - Nombre del *driver*
 - Nombre del usuario con el que se ha conectado
 - Número máximo de caracteres para el nombre de usuario
 - Número máximo de columnas que se permiten en una tabla
- Añadir a la clase **ControladorLogin** una función para recuperar los metadatos de la conexión y enviarlos a la vista **vistaMetadatos**

Más información en <https://docs.oracle.com/javase/7/docs/api/java/sql/DatabaseMetaData.html>

Ejercicio 4 (Opcional)

Añadir al proyecto funcionalidad para poder conectarse a un SGBD MySQL
