

Diseño y Desarrollo de Sistemas de Información

Grado en Ingeniería Informática

Práctica 1

Conexión a una BD mediante JDBC

Curso 2023/24

Duración estimada: 2 sesiones (3 horas)

OBJETIVOS

- Conocer los conceptos fundamentales de la API JDBC para realizar una conexión a una base de datos relacional (Oracle y MariaDB)
- Aprender a realizar una aplicación en lenguaje *Java* capaz de conectar con una base de datos relacional, haciendo uso del patrón MVC

Ejercicio 1

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

En el proyecto se crearán los paquetes (capas) "**Modelo**", "**Vista**", "**Controlador**" y "**Aplicacion**".

Ejercicio 2

- En la capa **Modelo**, añadir al proyecto un fichero Java con una clase llamada "**Conexion**", con los siguientes atributos y métodos:
 - Un atributo de tipo **Connection**
 - El constructor **Conexion (String sgbd, String ip, String servicio_bd, String usuario, String password)**. Realiza una conexión con los siguientes parámetros:
 - **sgbd**. Nombre del SGBD al que se quiere conectar (Oracle o MariaDB)
 - **ip**. Dirección IP del servidor
 - **servicio_bd**. Nombre del servicio (Oracle) o de la base de datos (MariaDB)
 - **usuario y password**. Nombre y clave del usuario con el que se realiza la conexión
 - Un método **Connection getConnection()**. Este método devuelve el atributo de tipo **Connection**
 - Un método **void desconexion()**. Este método cierra la conexión con la base de datos
 - En la capa **Vista**, añadir al proyecto un fichero Java con una clase llamada **VistaMensajes** que tendrá los siguientes métodos:
 - **void mensajeConsola (String texto)**. Muestra por pantalla el contenido del parámetro "texto"
 - **void mensajeConsola (String texto, String error)**. Muestra por pantalla el contenido de los parámetros "texto" y "error"
-

- En la capa **Controlador**, añadir al proyecto un fichero Java con una clase llamada **ControladorLogin** que tendrá los siguientes atributos y métodos:
 - Un atributo de tipo **Conexion**
 - Un atributo de tipo **VistaMensajes**
 - El constructor **CotroladorLogin()**
- (a) La función del constructor es establecer la conexión con la base de datos. Para ello, se programará una función privada **Conexion conectarBD()**, que realizará la conexión y devolverá un objeto de tipo **Conexion**. Esta función mostrará un mensaje, mediante un objeto de tipo **VistaMensajes**, para saber si la conexión ha sido correcta o se ha producido algún error.

NOTA: para facilitar la ejecución del programa, los datos de la conexión (el SGBD, dirección IP, usuario, password, etc.) se escribirán en el propio código.
- (b) Una vez establecida la conexión, el programa deberá realizar la desconexión mediante una función privada **void desconectarBD()**, que deberá mostrar un mensaje indicando el resultado de la operación.

En estas dos funciones se deben capturar las excepciones mediante bloques **try-catch**.

- En la capa **Aplicación**, añadir a la clase principal el código necesario para instanciar un objeto de clase **ControladorLogin**

Ejercicio 3 (Opcional)

La clase **Connection** tiene un método, llamado **getMetaData()**, que devuelve 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.
- En la capa **Vista**, añadir un fichero Java con una clase llamada **VistaInfoBD**, que tendrá un método **void infoMetadatos (DatabaseMetaData dbmd)**. Esta vista mostrará la siguiente información:
 - Nombre de la base de datos
 - Versión de la base de datos
 - URL de la conexión
 - Nombre del *driver*
 - Versión del *driver*
 - Nombre del usuario con el que se ha conectado
 - Lista de términos SQL que tiene la base de datos que no pertenecen al SQL estándar
- Añadir a la clase **ControladorLogin** una función para recuperar los metadatos de la conexión y enviarlos a la vista **VistaInfoDB** para que los muestra mediante el método **infoMetadatos()**

Toda la información en

<https://docs.oracle.com/en/java/javase/11/docs/api/java.sql/java/sql/DatabaseMetaData.html>
