

Diseño y Desarrollo de Sistemas de Información

Grado en Ingeniería Informática

Práctica 3

Acceso a Bases de Datos con Hibernate Operaciones CRUD (1ª parte)

Curso 2023/24

Duración estimada para las dos partes: 3 sesiones (4,5 horas)

OBJETIVOS

- Aprender a realizar el mapeo objeto relacional entre clases Java y tablas del modelo relacional a través de un software ORM (Hibernate)
- Aprender a realizar operaciones CRUD mediante Hibernate utilizando el patrón de diseño MVC (Modelo-Vista-Controlador)

Crear un nuevo proyecto Maven que se utilizará para el resto de las prácticas del curso y que, finalmente, será el proyecto de la asignatura.

Añadir los siguientes ficheros, clases y métodos:

1. Crear los ficheros **hibernateMariaDB.cfg.xml** e **hibernateOracle.cfg.xml**
2. Crear un paquete **Config**, en el que se crearán las clases **HibernateUtilMariaDB.java** e **HibernateUtilOracle.java**
3. En la capa **Modelo**:
 - Crear las clases **Monitor.java**, **Socio.java** y **Actividad.java** para mapear las respectivas tablas de la base de datos
 - Crear las clases **MonitorDAO.java**, **SocioDAO.java** y **ActividadDAO.java**, a las que se irán añadiendo los métodos necesarios para programar la funcionalidad del proyecto

NOTA IMPORTANTE. Los métodos de las clases DAO recibirán, por parámetro, un objeto de tipo **Session**, que corresponderá con una sesión abierta en el controlador. Ese objeto será el encargado de realizar las llamadas a las bases de datos

4. En la capa **Controlador**:
 - Crear la clase **ControladorLogin.java** con un constructor que establezca la conexión con la base de datos. La clase tendrá un atributo de tipo **SessionFactory** para establecer la conexión con los SGBD y se reprogramará la función privada **void conectarBD()** para establecer la conexión a través de Hibernate. Esta función debe poder conectarse a los SGBD Oracle y MariaDB en función de las preferencias del usuario. El constructor instanciará, además, un objeto de tipo **ControladorPrincipal** pasándole, por parámetro, el objeto de tipo **SessionFactory** creado
 - Crear la clase **ControladorPrincipal.java** con un constructor que reciba un parámetro de tipo **SessionFactory**, que será el objeto creado en la clase **ControladorLogin.java** y que mantendrá la conexión con la base de datos. Este controlador tendrá un atributo de tipo **Session** y un atributo de tipo **Transaction** para realizar las operaciones con las bases de datos

- Desde este constructor se llamará a una función que mostrará un menú de opciones con algunas operaciones de consulta sobre la base de datos

 - 1. **Información de los socios (HQL).** Muestra todos los campos de todos los socios
 - 2. **Información de los socios (SQL Nativo).** Igual que la opción 1
 - 3. **Información de los socios (Consulta nombrada).** Igual que las opciones 1 y 2
 - 4. **Nombre y teléfono de los socios.** Muestra el nombre y el teléfono de todos los socios (la consulta debe recuperar sólo estos dos campos de la base de datos)
 - 5. **Nombre y categoría de los socios.** Muestra el nombre y la categoría de los socios de una determinada categoría. El programa solicitará la categoría por teclado
 - 6. **Responsable de una actividad.** Muestra todos los campos del Monitor responsable de una actividad. El programa solicitará el código de la actividad por teclado
 - 7. **Socios de una actividad.** Muestra el nombre de los socios inscritos en una actividad. El programa solicitará el código de la actividad por teclado
-
5. En la capa **Vista**:
 - Crear las vistas necesarias para mostrar la información solicitada