Desarrollo de Aplicaciones Multiplataforma

Actividad 02

Base de datos con JDBC y ORM

Acceso a datos

Actividad

Base de datos con JDBC y ORM.

**Objetivos**

Usar gestores de bases de datos incrustados e independientes.

Establecer la conexión.

Definir la estructura de la base de datos.

Desarrollar aplicaciones que modifican el contenido de la base de datos.

Definir los objetos destinados a almacenar el resultado de las consultas.

Desarrollar aplicaciones que hacen consultas

Eliminar los objetos una vez finalizada su función.

Gestionar las transacciones.

Instalar la herramienta ORM.

Configurar la herramienta ORM.

Definir los archivos de mapeo.

Aplicar mecanismos de persistencia a los objetos.

Desarrollar aplicaciones que modifican y recuperan objetos persistentes.

Desarrollar aplicaciones que realizan consultas utilizando el lenguaje SQL.

Administrar transacciones.

|  |
| --- |
| **¿Cómo lo hago?** |
| 1. Rellena los datos que se piden en la tabla “Antes de empezar”. 2. Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc. 3. Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar. 4. Lee detenidamente todos los pasos a continuación. Realiza todas las acciones indicadas y responde a todas las preguntas planteadas. 5. Deberás realizar una aplicación **JAVA** que incluya todas las clases y métodos necesarios para dar respuesta a todos los ítems marcados en el enunciado. 6. *La entrega de la actividad estará formada por el proyecto que hayas creado en* ***NetBeans*** *como solución de la actividad, comprimido en formato* ***.zip****.* 7. Recuerda nombrar el archivo siguiendo estas indicaciones:  * Ciclo\_Módulo o crédito\_Tema\_ACT\_número actividad\_Nombre y apellido   + Ejemplo: AF\_M01\_T01\_ACT\_01\_Maria Garcia |

|  |  |
| --- | --- |
| **Antes de empezar…** | |
| Nombre |  |
| Apellidos |  |
| Módulo/Crédito |  |
| UF (solo ciclos LOE) |  |
| Título de la actividad |  |

Deseamos construir una aplicación que permita acceder a los empleados a sus incidencias en un sistema de gestión de incidencias. La aplicación se debe construir utilizando el **ORM Hibernate**.

La base de datos que vamos a utilizar, tiene las siguientes tablas:

* Tabla **incidencias**: Albergará las incidencias de los empleados. Contendrá para cada una la fecha-hora de generación, el origen, el destino, detalle de la incidencia y el tipo de la misma.
* Tabla **empleados**: Contendrá los datos de inicio de sesión de los empleados al sistema: nombre de usuario y contraseña. Además, incluirá los datos del empleado. Tendrá datos como el nombre completo y teléfono de contacto, como mínimo.
* Tabla **historial**: Registrará los eventos de inicio de sesión (I), consulta de incidencias destinadas para un empleado (C) y creación de incidencias urgentes (U). Para ello dispondrá de tres campos: el tipo de evento (“I”, “E” y “U” respectivamente), la fecha-hora del evento y el nombre de usuario del empleado que lo ha generado.

Todas las tablas tendrán, sus correspondientes identificadores en forma de clave primaria.

Con todo esto en mente, tenemos que realizar lo siguiente:

1. Crear las clases POJO correspondientes para representar de forma adecuada los datos de las tablas explicadas en la lista anterior (Incidencia, Empleado, Historial). ***(0.50 puntos)***
2. Crear un backend apropiado para la gestión de profesores. Llamar a la clase ‘**IncidenciasORM**’. Mostrará un menú que deberá permitir: ***(3.00 puntos)***
   1. Insertar un empleado nuevo en la B.D.
   2. Validar la entrada de un empleado (suministrando usuario y contraseña)
   3. Modificar el perfil de un empleado existente.
   4. Cambiar la contraseña de un empleado existente.
   5. Eliminar un empleado.
3. Modificar el backend anterior de manera que gestione también las incidencias, muestre un menú con las siguientes opciones: ***(3.00 puntos)***
   1. Obtener un objeto Incidencia a partir de su Id.
   2. Obtener la lista de todas las incidencias.
   3. Insertar una incidencia a partir de un objeto de clase Incidencia definido adecuadamente según los campos que presenta (incluido el empleado que la origina y el empleado destino).
   4. Obtener las incidencias para un empleado a partir de un objeto de clase Empleado.
   5. Obtener las incidencias creadas por un empleado concreto.
4. Crear un backend apropiado para la gestión del historial. Llamar a la clase ‘**HistorialORM**’. Mostrará un menú que deberá permitir: ***(3.00 puntos)***
   1. Insertar los eventos mediante sendos métodos.
   2. Obtener la fecha-hora del último inicio de sesión para un empleado concreto.
   3. Obtener el ranking de los empleados por cantidad de incidencias urgentes creadas (más incidencias primero).
   4. Obtener la posición dentro del ranking para un empleado concreto.
5. Construir una aplicación Java que demuestre el uso de los diferentes elementos programados. Llamar a dicho programa ‘**TestORM**’. ***(0.50 puntos)***

Este programa debe mostrar al usuario un menú con las opciones que sean necesarias para poder probar cada una de las funcionalidades.