|  |
| --- |
| **CFGS DESARROLLO DE APLICACIONES MULTIPLATAFORMA (DAM)** |
| Tarea 04 |
| Acceso a Datos |

|  |
| --- |
| César Grande Conde  29-11-2022 |

1. Instalación de la base de datos NetBeans

Creamos un nuevo proyecto en netbeans y nos vamos al apartado de Services y en Databases creamos una nueva conexión y cargamos el conector, en mi caso será para mysql

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Introducimos los datos necesarios, host, port,… En mi caso serán los siguientes:

* Host: localhost
* Port: 3306
* Database: idiomas
* User name: root
* Password: cesar

Y en el botón Test connection comprobamos si nos conectamos correctamente a la base de datos

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

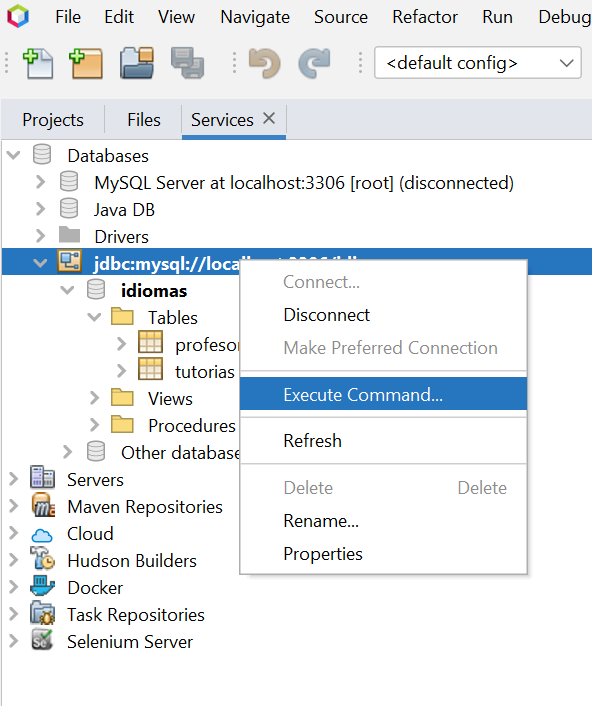
Descripción generada automáticamente

Una vez verificado la correcta conexión damos a next > next > Finishy como se muestra nos ha creado una conexión a la base de datos para poder trabajar con ella desde netbeans

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Desde netbeans podemos realizar consultas para ello ejecutamos Execute Command



Y en la ventana que nos aparece a la derecha, introducimos la consulta o el comando que deseamos, en nuestro caso copiamos las líneas sql que vienen en la tarea

Texto

Descripción generada automáticamente

Comprobamos que nos ha creado bien la base de datos con una nueva execute command

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

1. Configuración de Hibernate

Una vez creada la base de datos vamos a crear un nuevo proyecto de tipo **Maven y Java Application**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Descargo las funcionalidades que me pide netbeans

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Configuro el proyecto como se muestra en la imagen

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez creado el proyecto debemos de crear las dependencias en el fichero pom.xml

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para ello debemos ir al repositorio de Maven en el siguiente enlace <https://mvnrepository.com/> y buscar las siguientes dependencias:

1. Mysql connector

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Y seleccionamos la versión 8.0.28

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Y copiaremos en nuestro fichero pom.xml el siguiente código maven

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Hibernate será la segunda dependencia que debemos de incluir. Ahora debemos de **elegir la misma versión del hibernate y del hibernate-entitymanager** y la versión que sabemos que funciona bien es la 5.6.14.Final

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Igual que antes copiamos en el fichero pom.xml el código maven

1. Hibernate entity manager será la tercera dependencia que debemos de incluir

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Seleccionamos la versión 5.6.14.Final y pegamos el código maven en pom.xml

1. Hibernate JPA (Java Persistence API) será la última dependencia que debemos de incluir

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

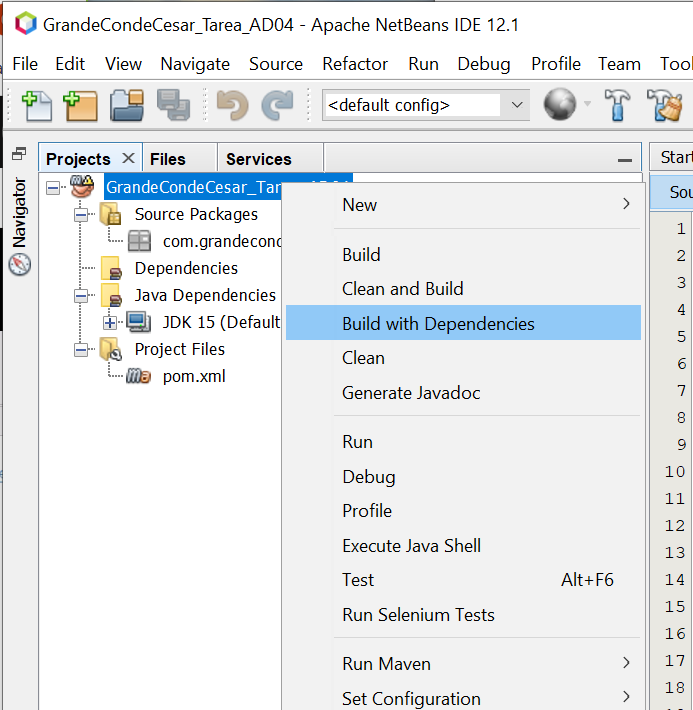
Descripción generada automáticamente

Seleccionamos la versión 1.0.2.Final y copiamos el código maven en el fichero pom.xml. Y con estas dependencias, el archivo pom.xml estaría terminado quedando de la siguiente manera

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Guardamos y después damos el boton Build withDependencies para que nos las cree.

****

Y nos crea dentro de la carpeta Dependencies las dependencias mencionadas anterior mente

Texto

Descripción generada automáticamente con confianza media

El siguiente paso es crear el archivo persistence.xml para ello creamos nuevo Persistence Unit y le damos a la estrategia de creación None porque si seleccionas Create o Drop and Create puede dar problemas

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En la siguiente imagen la dejamos por defecto. En persistence Library no sale Hibernate, por ese motivo dejamos EclipseLink(JPA 2.1) el cual después cambiaremos a mano

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En la pestaña Source donde marco tenemos que cambiarlo por

*org.hibernate.jpa.HibernatePersistenceProvider*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Comprobamos que tengamos bien los demás datos (url, usuario, driver y password) y guardamos

Texto

Descripción generada automáticamente

Ahora vamos a crear la creación automática de los POJOS para ello en nuevo, seleccionamos Entity Classes from Database

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ahora para la conexión que tengo activa me muestra las tablas que tengo disponible, añadimos todas

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Dejamos que nos genere las Query annotations de persistencia

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En la siguiente pestaña configuramos el mapeo de la siguiente forma

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Y nos genera las clases java que son nuestras tablas de la base de datos con los @Entity

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Una vez creada las clases volvemos al archivo pom.xml y nos fijamos que nos ha creado dependecias innecesarias las cuales borraremos para que no creen conflictos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Nuestro siguiente paso será configurar adecuadamente las clases con los relaciones, la base de datos de la tarea cuenta con una relación 1:N que en hibernate lo denomina OnToMany. La clase profesores tiene una relación 1:N y la configuramos para que se actualice en cascada.

Texto

Descripción generada automáticamente con confianza media

import javax.persistence.CascadeType;

Y en la clase tutorías quedara con una relación N:1, se relacióna con la columna llamada codProfe atraves del objeto llamado profesor de la clase Profesores

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

import javax.persistence.FetchType;

Una vez creada las relaciones comprobamos que el constructor, getters, setters y métodos creados sean correctos de las dos clases (Profesor y Tutorias)

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ahora en el main crearemos gestor de persistencia con las clases EntityManager y EntityManagerFactory

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Sentencias DML

Voy a leer las dos tablas de la base de datos, aunque inicialmente la tarea no lo pide, simplemente para asegurarme que lo hace bien, para ello uso dos métodos verProfes() y verTutorias()

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

La consulta a ambas tablas la hace correctamente como se observa en la siguiente imagen

Texto

Descripción generada automáticamente

El siguiente paso será realizar una inserción a la tabla Tutorias, para ello realizaremos mediante el método insertarTutorias(), debemos de crear antes un profesor ya que no puede existir una tutoría sin profesor asignado

Texto

Descripción generada automáticamente con confianza baja

Obteniendo el siguiente resultado tras la inserción

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

El siguiente paso será el borrado sobre la tabla Tutorias, para ello usaremos el método borrarTutorias(). En la tarea no indica que haya que pedir por pantalla la tutoría que quiera borrar por ese motivo le voy a indicar directamente que quiero borrar la tutoría 0001

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Tabla Tutorías antes de borrar la 0001(izquierda) y tabla una vez borrada (derecha)

Tabla

Descripción generada automáticamente Tabla

Descripción generada automáticamente

1. Obtener un listado sobre las tablas profesores y tutorías que visualice codProfe, nombre, apellido, departamento, diaSemana y horaTutoria

Utilizaremos el método obtenerListado() el cual nos da el siguiente resultado

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

el cual nos da el siguiente resultado

Tabla

Descripción generada automáticamente