ACCESO A DATOS
UD4: Bases de datos orientadas a objetos (BDOO)

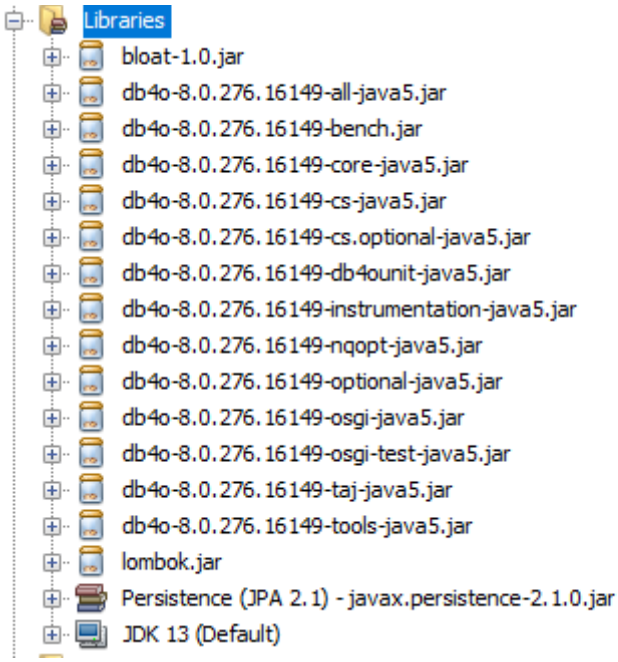
Consideraciones previas para trabajar en este tema.

Para trabajar con esta base de datos tenemos que irnos a una versión más antigua que la que tenemos. En concreto necesitaremos trabajar con la versión Apache Netbeans 11 y la versión 13 del jdk.

Podríais instalarlo en vuestro equipo anfitrión o en la máquina virtual para no interferir. Os he dejado un video de como instalarlo.

Los proyectos en esta unidad serán ant y no maven.
(Esto se debe a que no funcionan muy bien las librerias maven en el pom.xml y por eso las incorporamos con ant).

En el proyecto base ya os dejo las librerias incorporadas: las de db40, las del lombok (por si alguien lo usa y las de Persistence JPA por que en el model que hemos creado en la unidad anterior lo recogiamos a través de la base de datos. Realmente si hicieramos nosotros el modelo a mano solo necesitariamos las de db4o, pero en el base ya os dejo todas por si os hiciera falta.

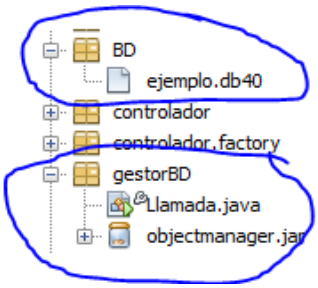


En el caso de que tuvierais que añadirlas a mano, las librerias jar las añadimos a través de Libraries-→(botón derecho) add jar y escogiendo la ruta , y la de persistence a través de Libraries-→ (botón derecho) add library e importandola.

ACCESO A DATOS
UD4: Bases de datos orientadas a objetos (BDOO)

Criterios que utilizaremos en los ejercicios que vamos a plantear a lo largo de la unidad.

En esta unidad incorporaremos 2 paquetes a nuestro proyecto además de los ya conocidos. BD , y gestorBD

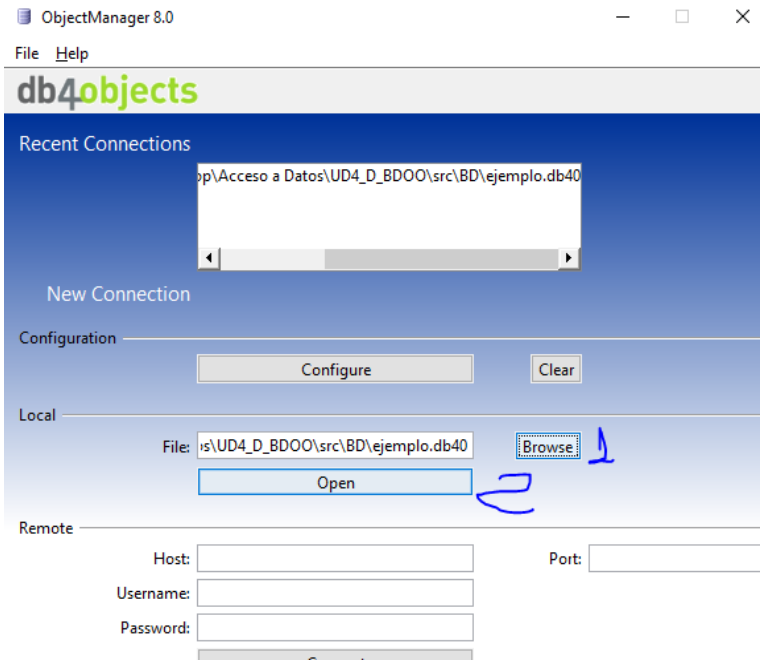


En esta unidad las base de datos se guardan como ficheros con extensión “db4o” , y para tener localizado físicamente la base de datos, las guardaremos en el paquete BD de nuestro proyecto.

Para poder visualizar el contenido de la BD utilizaremos el objectmanager.jar. Se incorpora por comodidad, ya que no sería necesario, un programa java que ejecutándolo con botón derecho y run file (Shift+F6) lo ejecuta.

Nota: observa que el la clase llamada.java llama al jdk situado en la carpeta de “c: \Archivos de programa\java\jdk-13.0.2”. Si en tu entorno de trabajo el jdk estuviera en una ruta diferente tendrás que adaptarlo.

Cuando se ejecuta llamada.java aparecerá una pantalla donde clickeando el browse tendrás que ir a la ruta donde está el fichero de la base de datos, y una vez localizado clickear open. Allí eligiendo la clase y el botón submit verás los datos.



ACCESO A DATOS
UD4: Bases de datos orientadas a objetos (BDOO)

Importante

No podrá estár ejecutándose la aplicación y el ObjectManager simultaneamente. Podría solucionarse abriendo/cerrando la base de datos por proceso, pero para seguir siempre el mismo criterio abrimos al iniciar el formulario y cerramos al salir, por eso no se va a poder abrir el ObjectManager hasta que se cierre la aplicación.

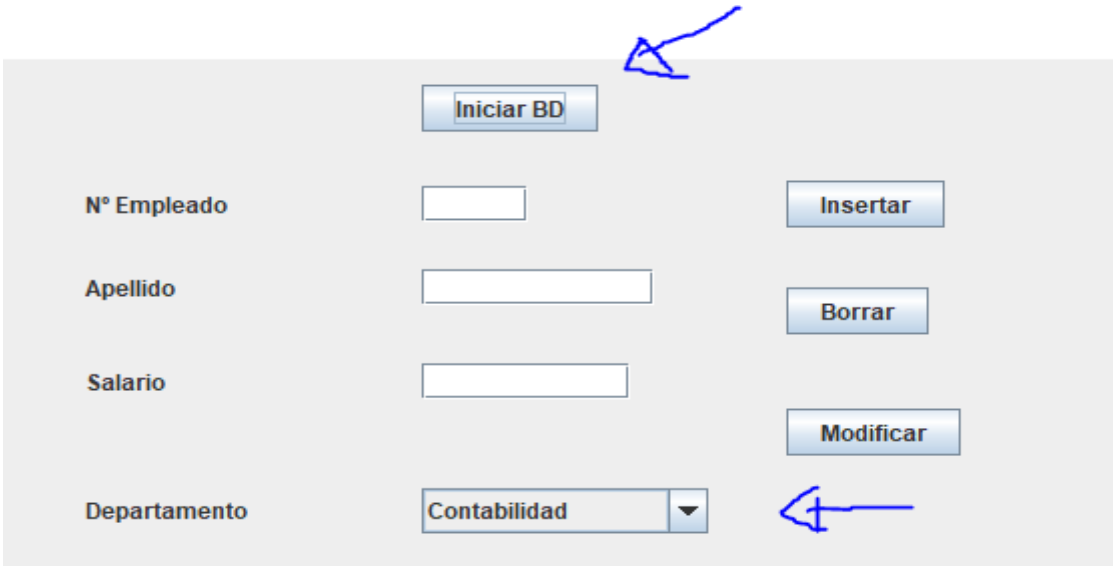
En las bdoo estas se crearán en función de como sea el modelo de las clases de trabajo. En nuestro ejemplo, por comodidad, vamos a utilizar las mismas clases que hemos utilizado en la unidad anterior.

ENUNCIADO 1: Crea un formulario que permita a través de un botón, iniciar la BD con datos para poder visualizarlos en el object manager, así como cargar el combo de departamentos. Podrás utilizar cualquiera de los 3 tipos de consulta que tenemos: QBE (QueryByExample), Soda o NQ (Native Query).

Se trata de crear un formulario tal que permita cargar datos de al menos 3 departamentos y opcionalmente al menos 1 o 2 empleados en un departamento. No sería imprescindible , pero si recomendable, que antes de insertar los departamentos compruebes que no existen previamente. (Para empezar a trabajar os recomiendo las consultas QBE pero se podrían usar cualquiera de las otras 2).
Debes verificar con el object manager que ha insertado los datos de los departamentos y de los empleados. Observa como los empleados aparecen en los departamentos como una colección, porque en el modelo por cada departamento guardamos la lista de empleados.

El diseño vamos a realizarlo para que en enunciados posteriores nos permita insertar, borrar y modificar los empleados.

El formulario podría ser tal que así:

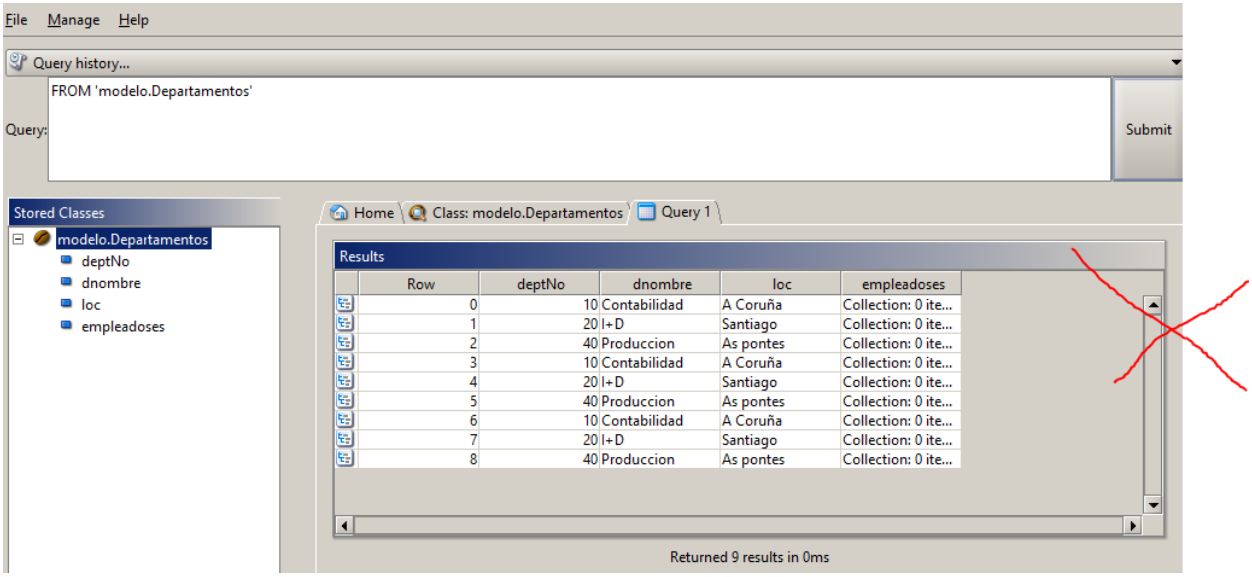


ACCESO A DATOS
UD4: Bases de datos orientadas a objetos (BDOO)

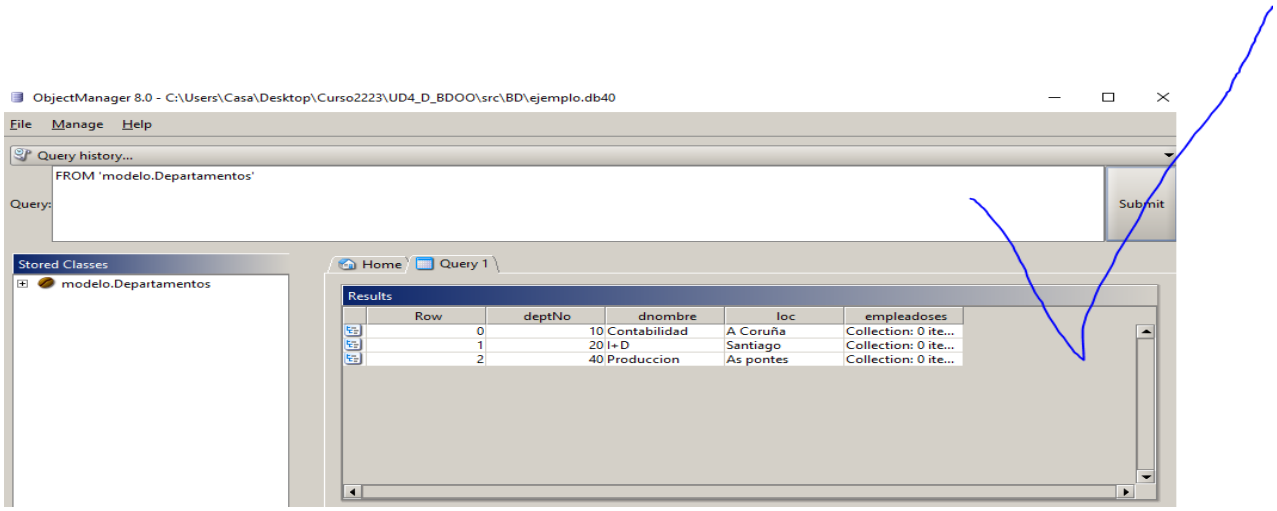
Consideraciones al enunciado

Recuerda que al trabajar con objetos y, no seguir el paradigma clásico de las bases de datos, no hay restricciones de primary key o foreign key. Si no haces la comprobación previa de que ya existe, podría insertarlos tantas veces como se realizara el proceso de iniciar la BD. Aunque la base de datos lo permite, nosotros siempre mantendremos la coherencia de los datos.

Ejemplo 1: Hemos clikeado hasta tres veces el botón de iniciar la base de datos con tres departamentos sin comprobar que existen o no previamente. Observa que el gestor permite incorporarlos, pero NO NOS INTERESA ESTA INCONSISTENCIA EN LA BD.



Ejemplo 2: Aunque le demos a iniciar la base de datos múltiples veces, comprueba si existen los datos previamente , y queda la base de datos CONSISTENTE.



Como la Base de datos es un fichero puedes borrarlo y hacer pruebas tantas veces como necesites.