**Introducción**

* **¿Qué es ObjectDB?**
  + Es un manejador de bases de datos orientado a objetos.
  + Provee de todos los servicios básicos para el manejo de bases de datos utilizando la filosofía orientada a objetos para el almacenamiento
  + Los objetos pueden ser almacenados y recuperados desde la base de datos
  + No es necesario definir tablas o convertir los objetos en registros usando SQL o ORM
  + Con ObjectDB se siguen 2 estándares fundamentales
    - JPA
    - JDO
  + Características
    - ObjectDB es un software multiplataforma
    - Multiplataforma en sistemas con Java SE 5 o superior.
    - Se puede integrar en
      * aplicaciones web Java EE y Spring
      * contenedores se servlets (TomCat, Jetty)
      * servidores de aplicaciones Java EE (GlassFish, JBoss)
    - Probado en varias máquinas virtuales de java (HotSpot, JRockit, IBM J9)
    - Tamaño máximo de la base de datos es de 128TB (131.072 GB)
    - Número de objetos en una base de datos es ilimitado (excepto por el tamaño)
    - Todos los tipos persistirles de JPA y JDO son soportados por ObjectDB, incluyendo las clases de entidad definidas por el usuario, clases insertables, colecciones javas estándar, tipos de datos básicos y cualquier otra clase serializarle.
    - Cada objeto en la base de datos tiene un identificador único.
    - ObjectDB soporta 2 lenguajes de consulta.
      * JDOQL
      * JPQL
      * También soporta Criteria
* **Instalación y descarga**
  + **¿Cómo instalar ObjectDB?**
    - Método. Zip

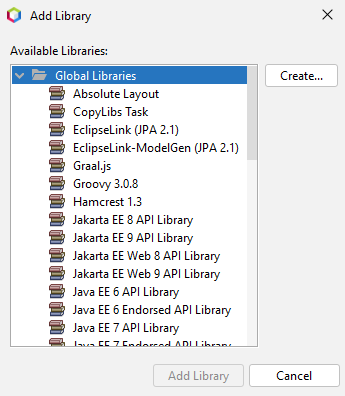
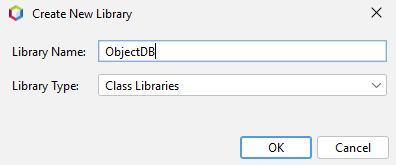
ObjectDB se distribuye como un archivo .zip, se descarga desde [aquí](https://www.objectdb.com/download), extraemos el archivo .zip y ya estaría, ya que no es necesario ejecutar ningún programa de instalación o configuración, se puede desinstalar ObjectDB eliminando el directorio de ObjectDB



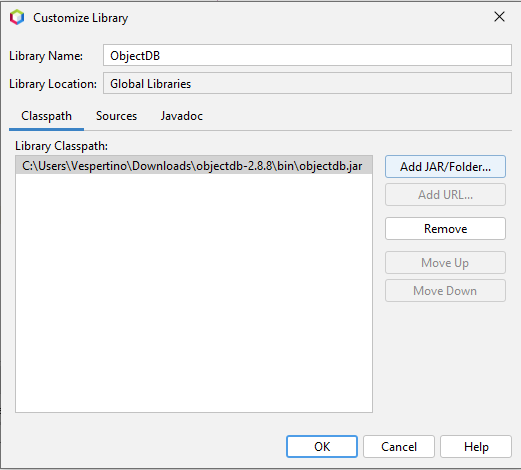
Nos creamos un proyecto con Ant



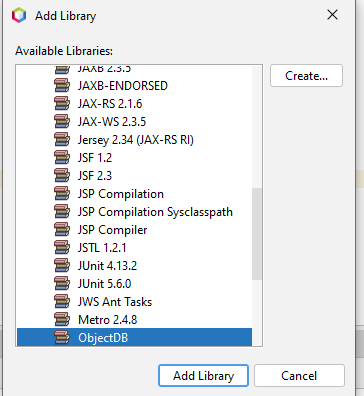
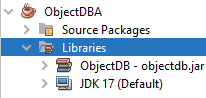
Creamos una nueva librería de ObjectDB



Añadimos el jar que previamente nos habíamos descargado



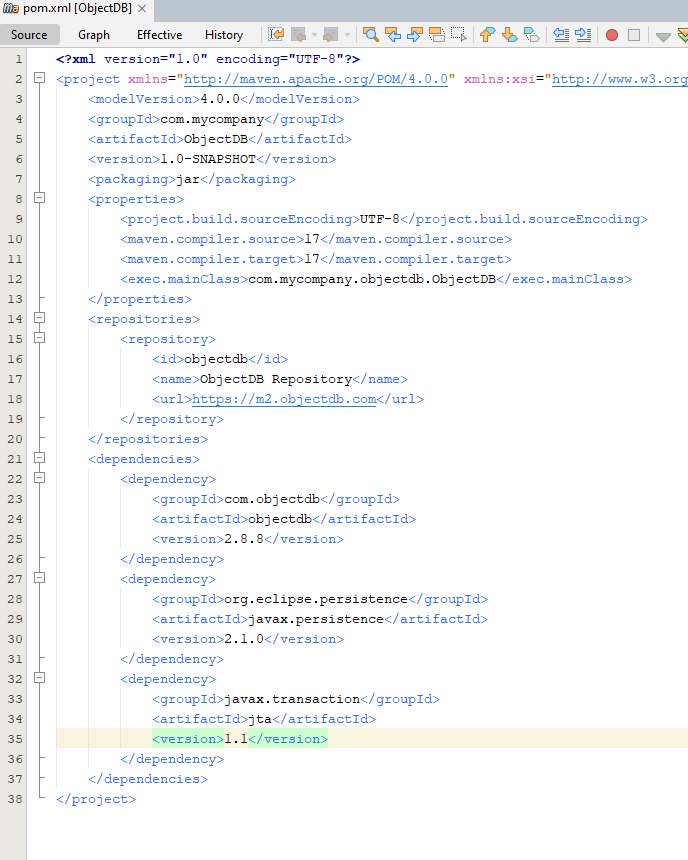
Añadimos la librería



* + - Método Maven

Añadir al pom.xml de nuestro proyecto el siguiente repositorio y las dependencias

Creamos un proyecto con Maven y añadimos el repositorio y las dependencias al pom.xml del proyecto



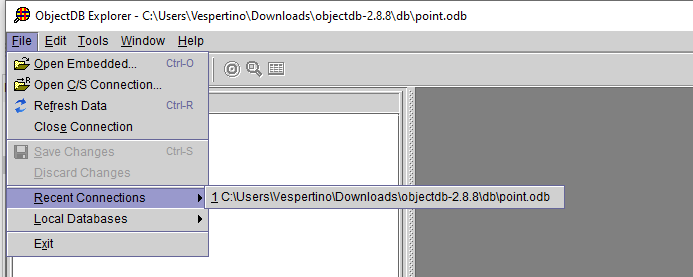
<https://prezi.com/cxciarvn6cbi/onjectdb/>

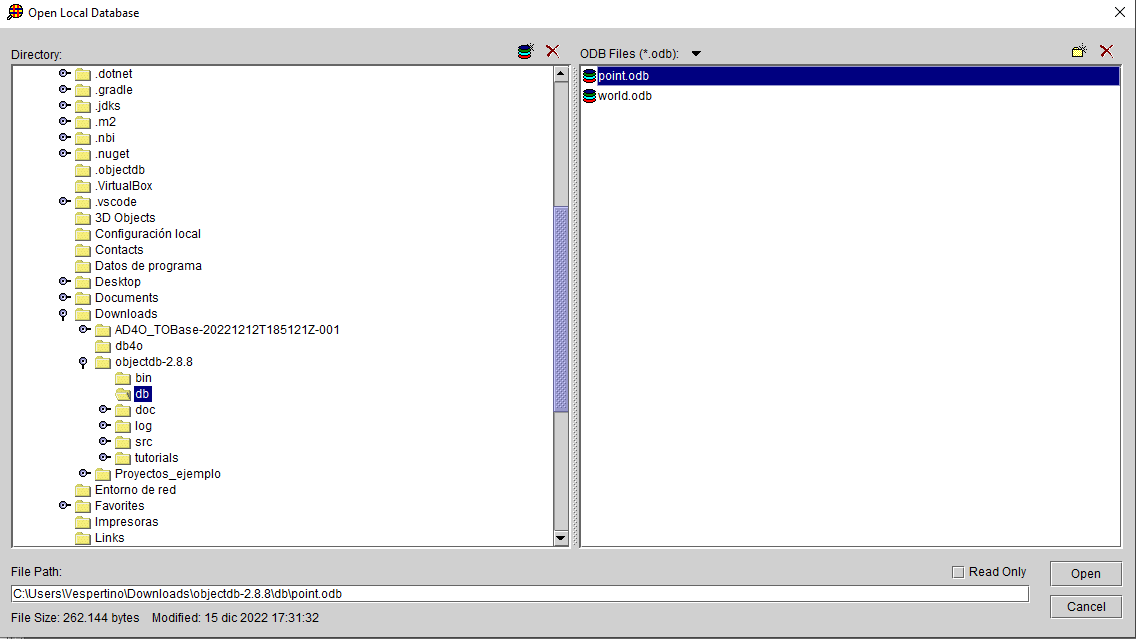
* **Requerimientos mínimos**
  + Java 5 o posterior
  + Es multiplataforma
  + ObjectDB podrá ejecutarse en cualquier plataforma que admita Java Standard Edition.
  + Se puede utilizar el kit de desarrollo de java (JDK) o el entorno de tiempo de ejecución de Java (JRE)
* **Introducción de datos utilizando la GUI del sistema**

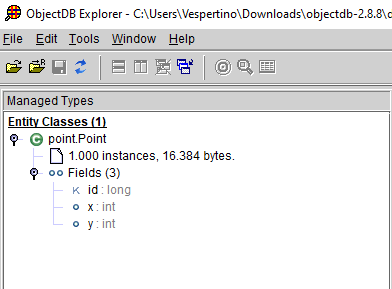
La GUI de ObjectDB es ObjectDB Explorer



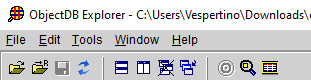


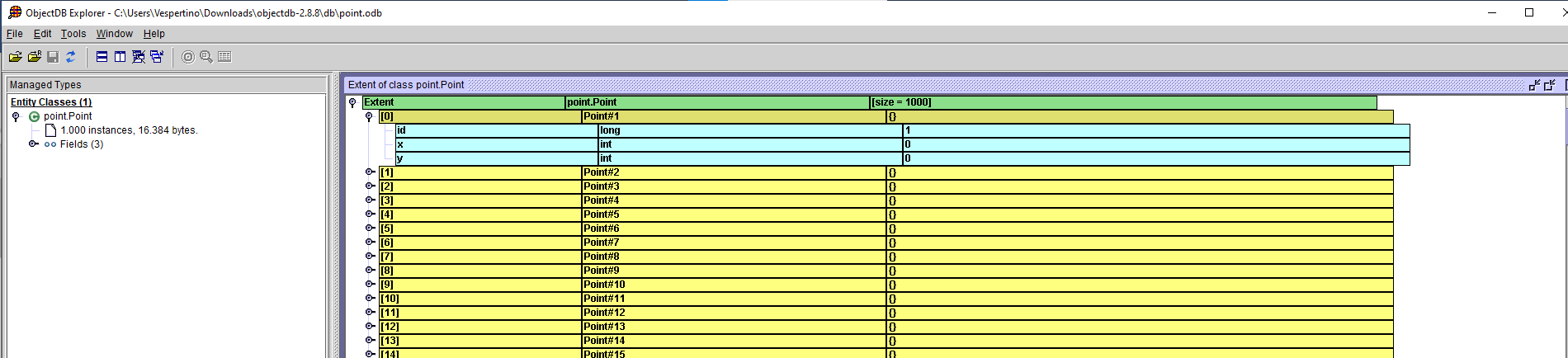




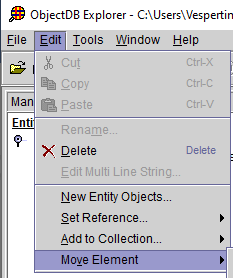


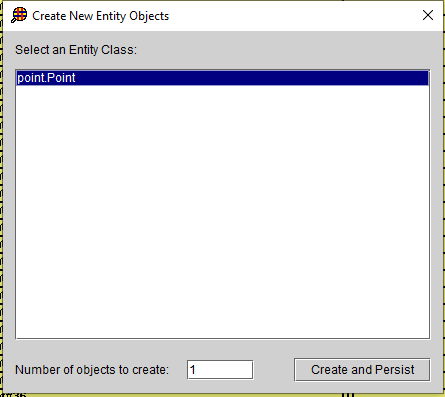
* **CONSULTAR**

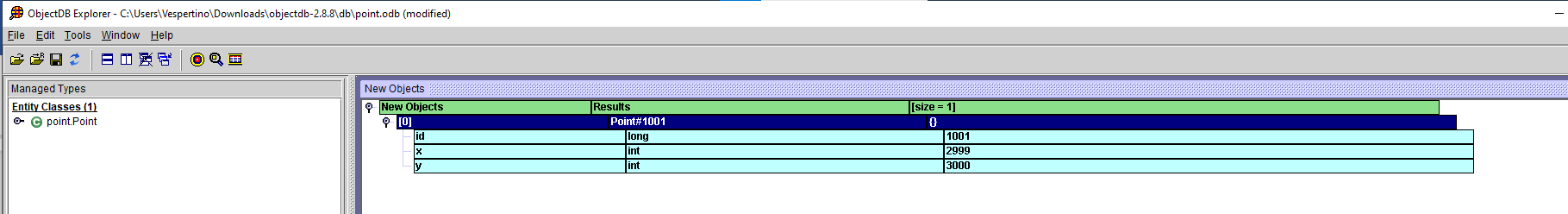




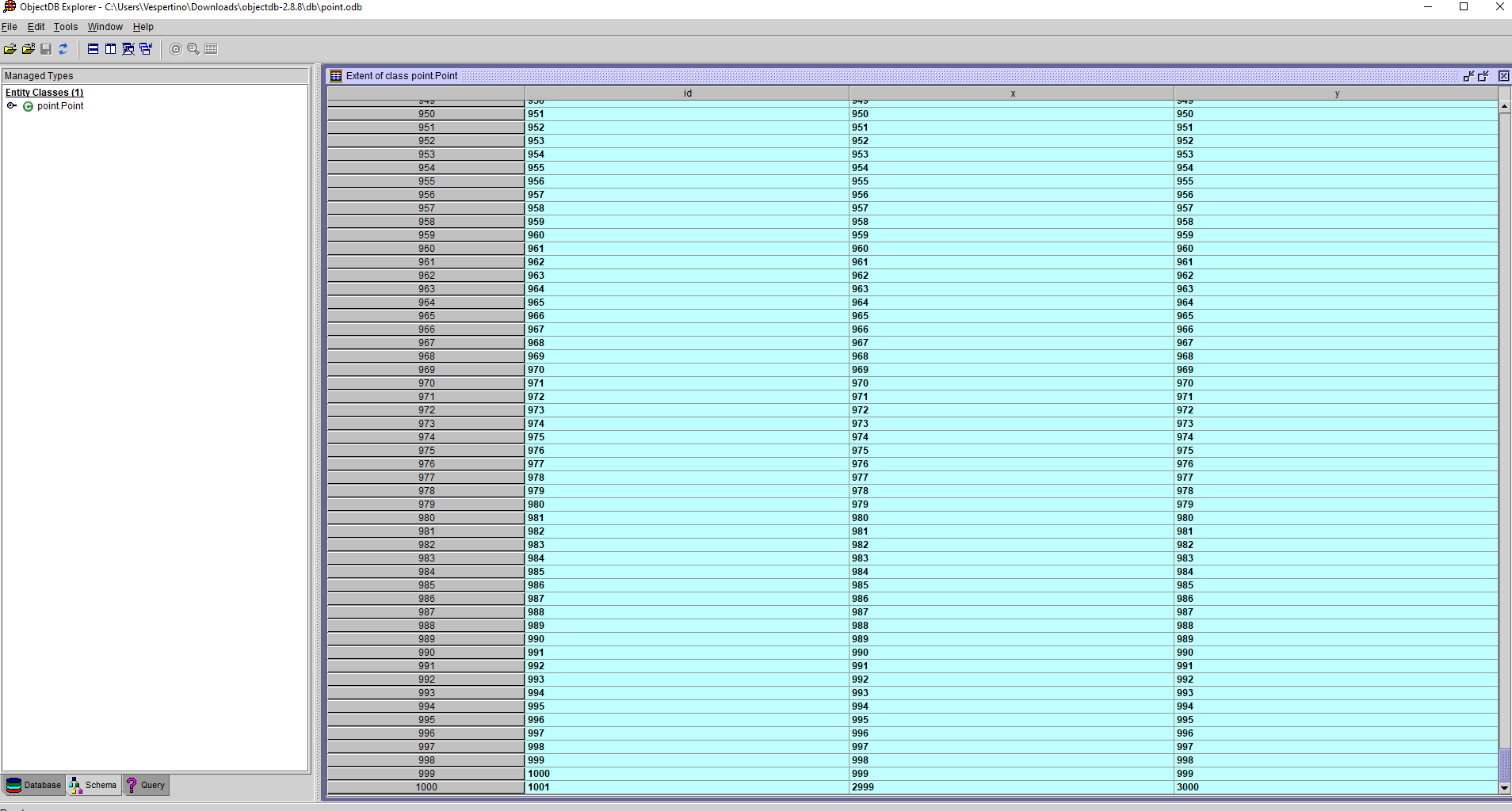
* **INSERTAR**







Ahora lo guardamos, refrescamos y comprobamos que se ha insertado

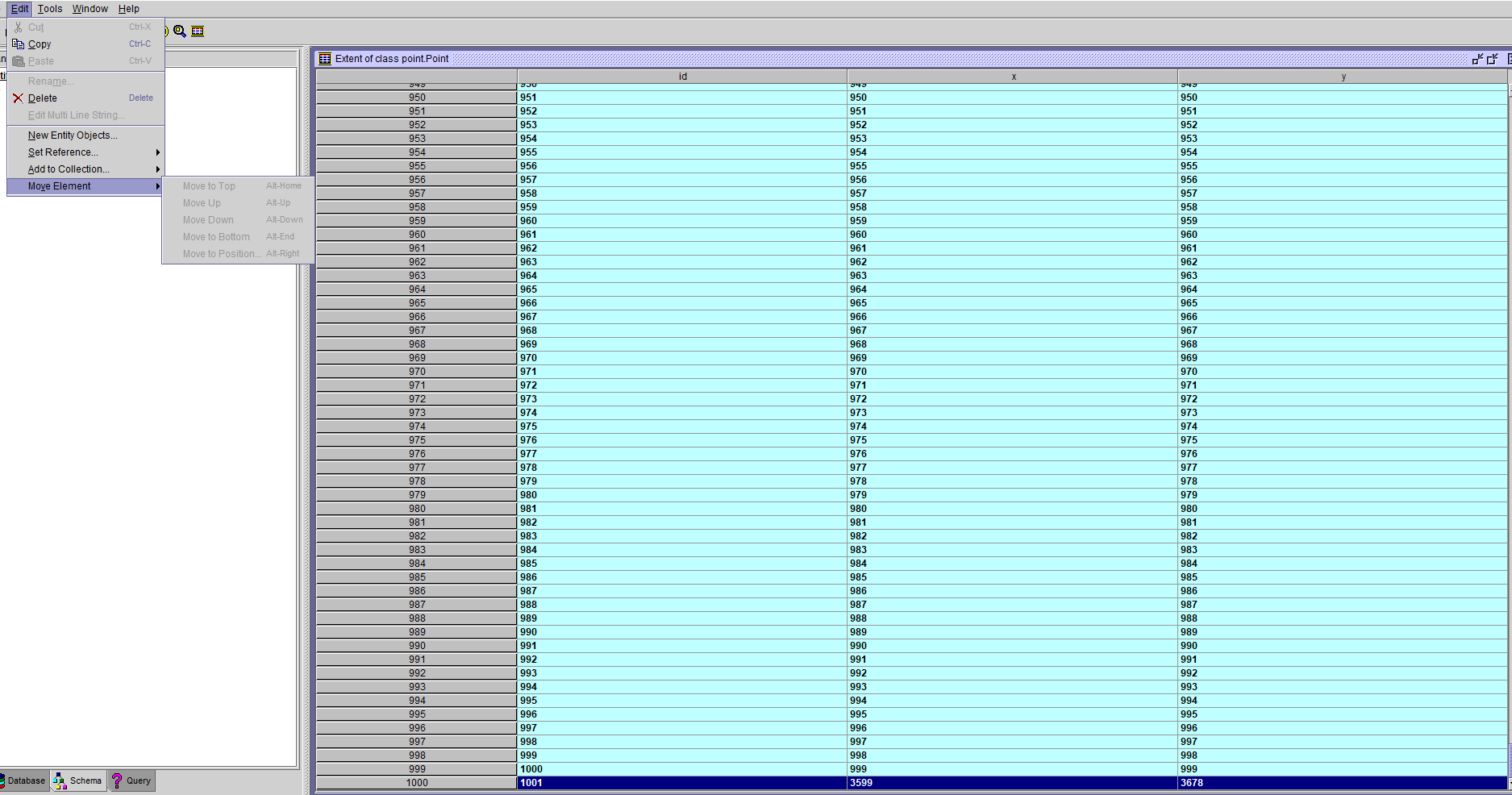


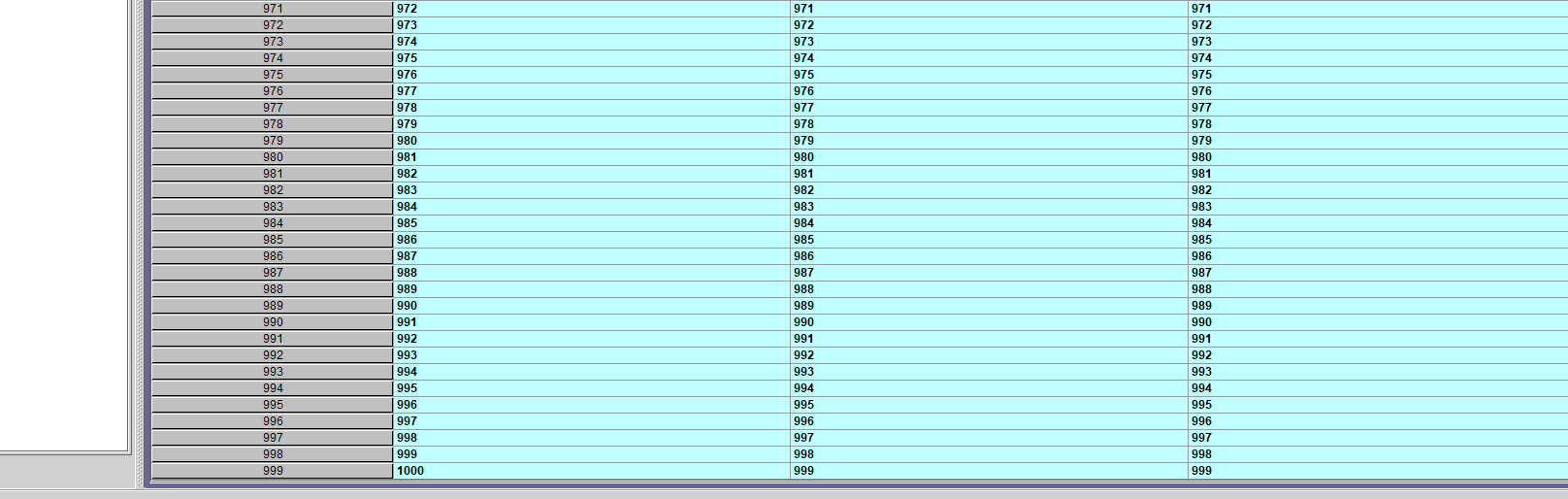
* **MODIFICAR**





* **BORRAR**

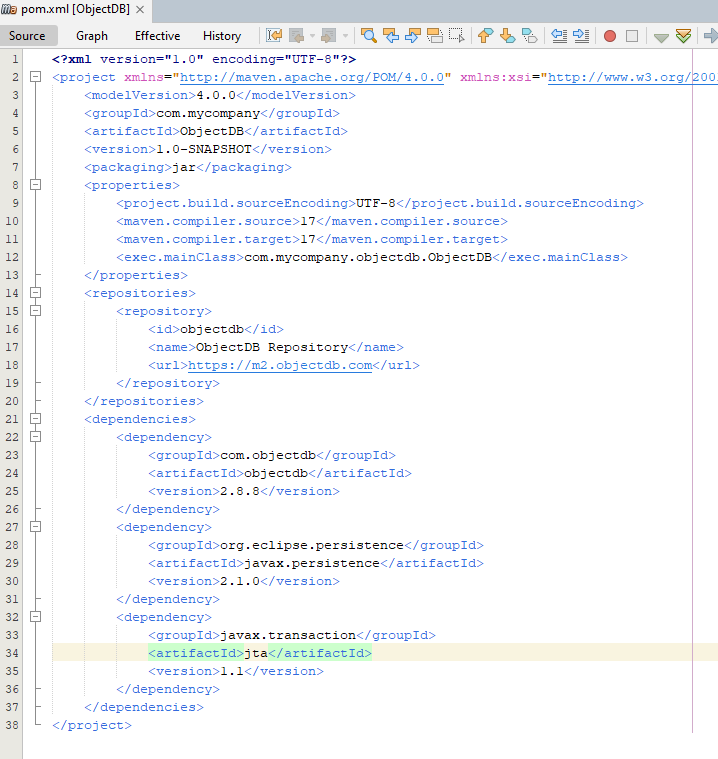


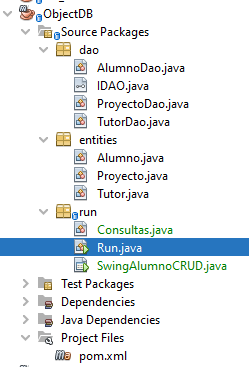


* **Configuración de un proyecto Java con ObjectDB**

Como ya vimos en el punto anterior tenemos dos opciones de integrar ObjectDB a nuestro proyecto java, ya sea utilizando el jar que previamente instalaremos para poder optar a la librería de ObjectDB en un proyecto con Ant o añadir al pom.xml el repositorio de ObjectDB con sus dependencias en un proyecto con Maven

En nuestro caso vamos a utilizar el Método Maven, para ello añadiremos las dependencias necesarias a nuestro Pom.xml para el correcto funcionamiento de ObjectDB

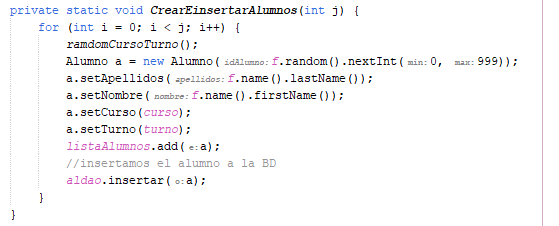


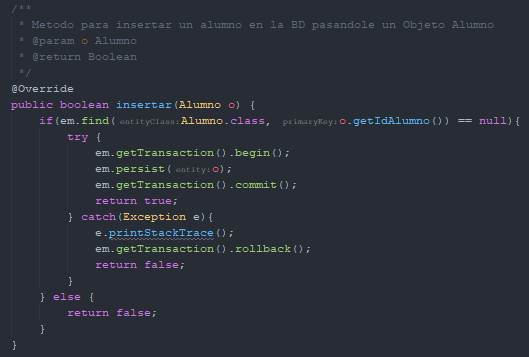


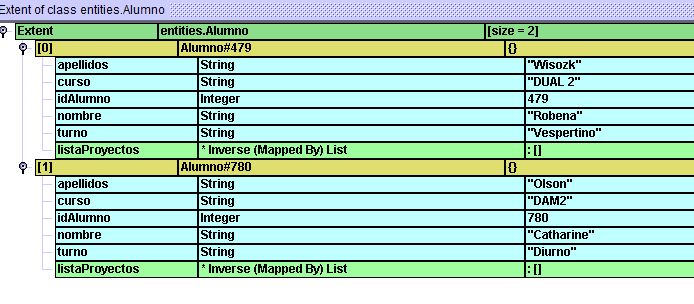
* **Explicación y código de las operaciones CRUD**

DAO ALUMNO

* **CREAR ALUMNO**

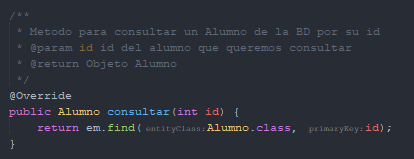


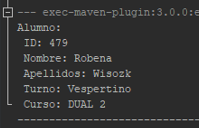




* **LEER ALUMNO**

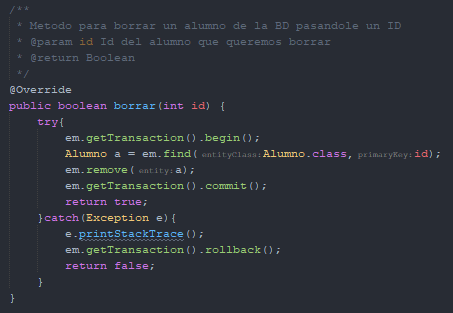


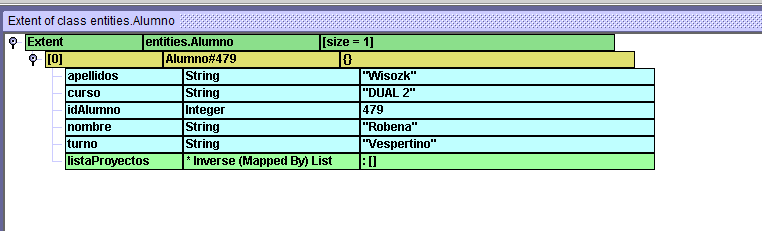




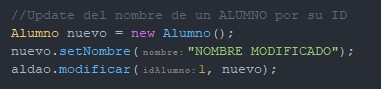
* **BORRAR ALUMNO**



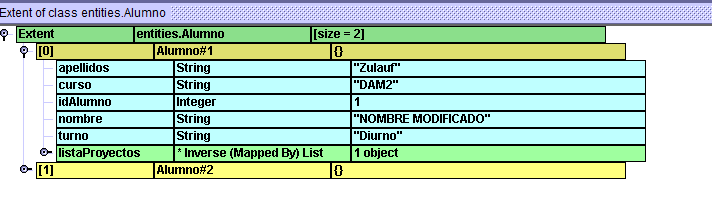




* **ACTUALIZAR ALUMNO**



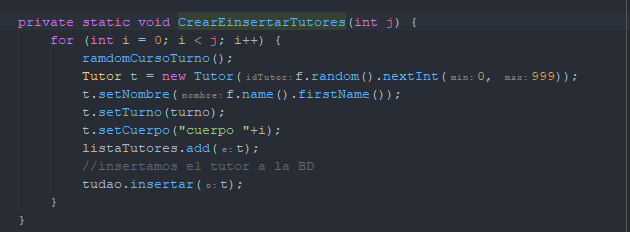


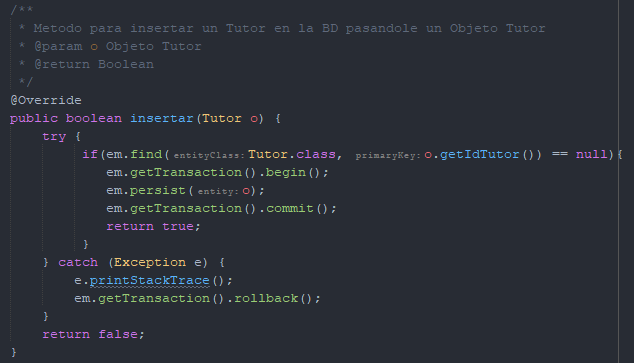


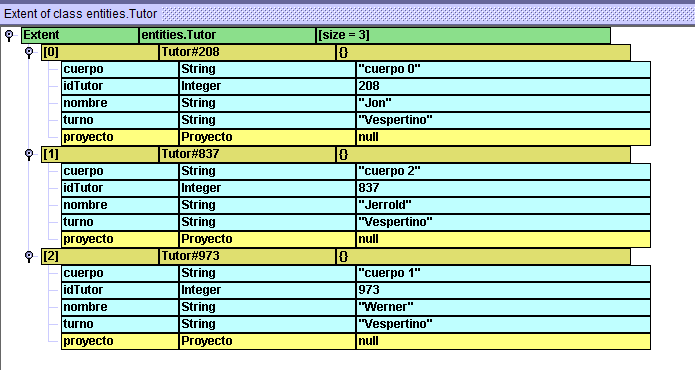
DAO TUTOR

* **CREAR TUTOR**



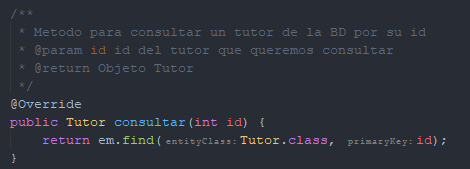


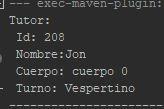




* **LEER TUTOR**

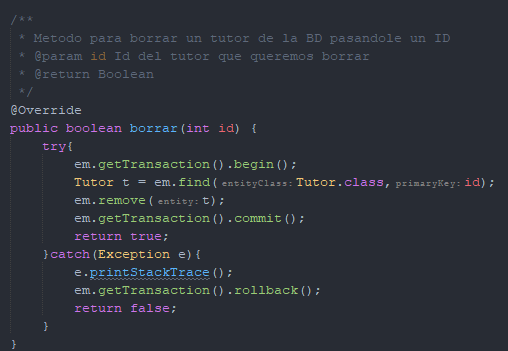


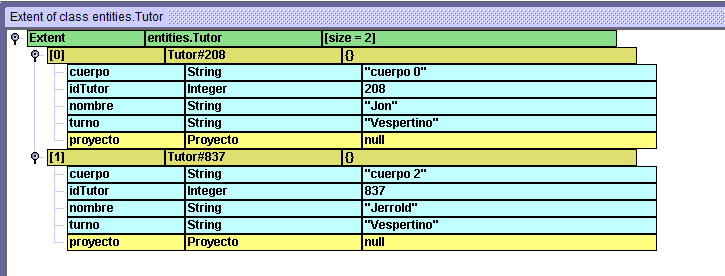




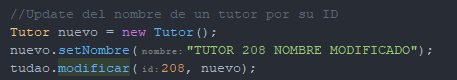
* **BORRAR TUTOR**



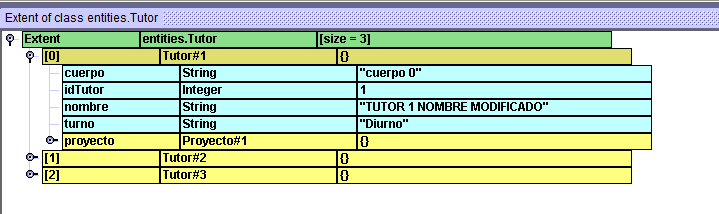




* **ACTUALIZAR TUTOR**



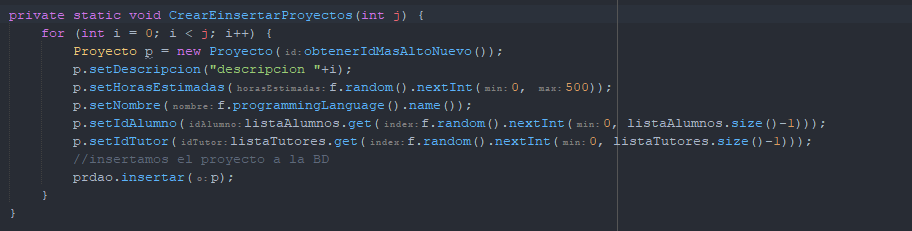




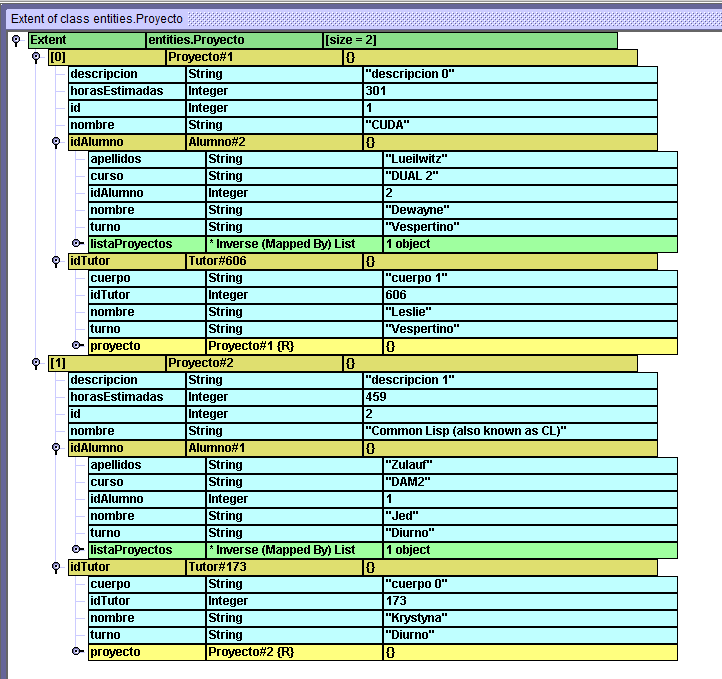
DAO PROYECTO

* + **CREAR PROYECTO**

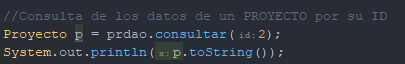


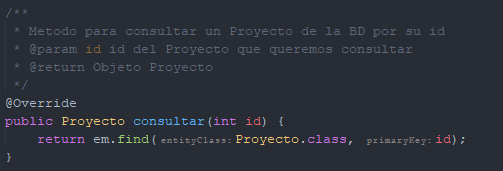


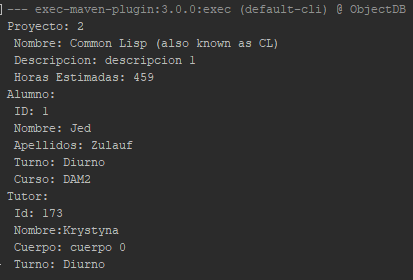




* + **LEER PROYECTO**

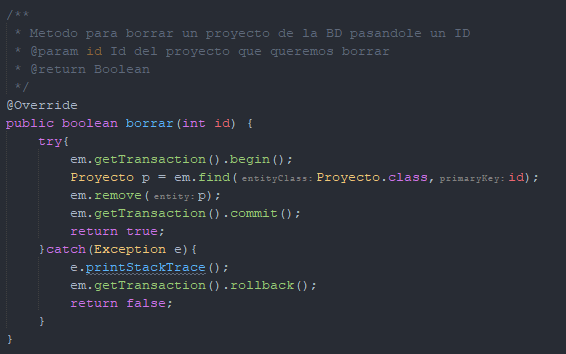


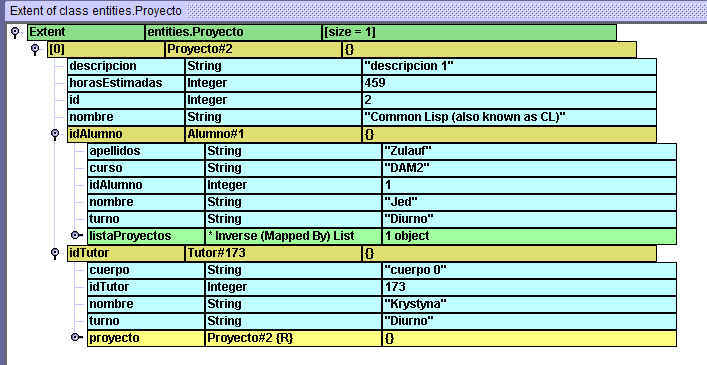




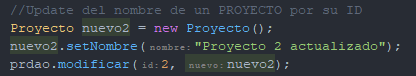
* + **BORRAR PROYECTO**



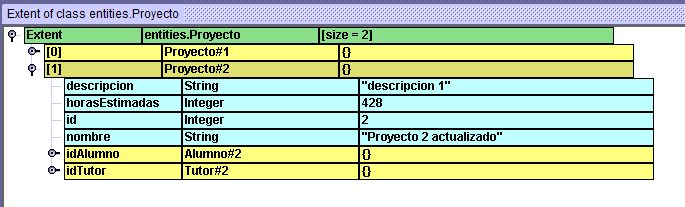




* + **ACTUALIZAR PROYECTO**

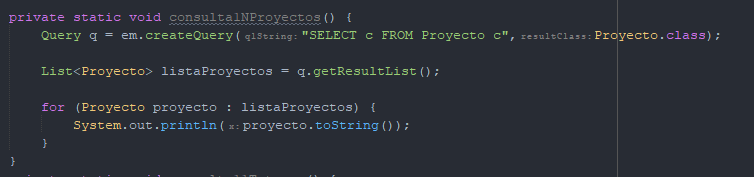


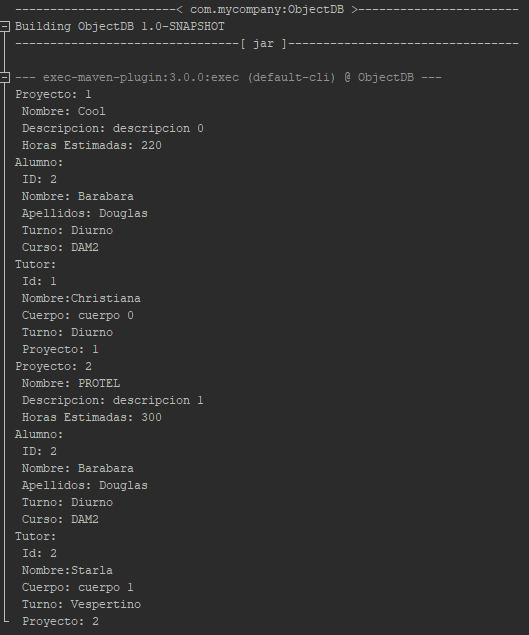




* **Tipos de consultas e implementación**
  + JPQL
    - Consulta 1:N

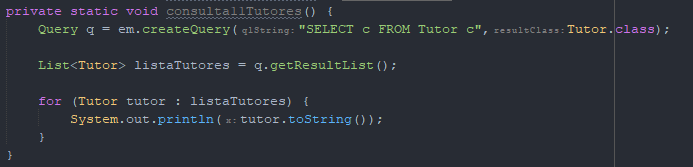
****

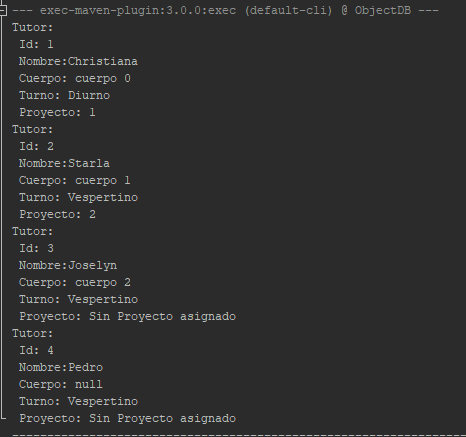
****

****

* + - Consulta 1:1: Mostrar los tutores que hay y el proyecto asignado de cada uno

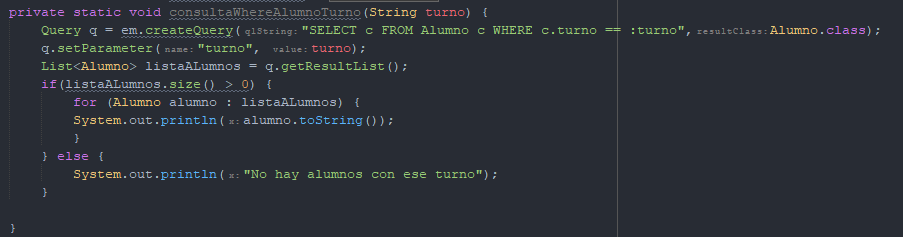
****

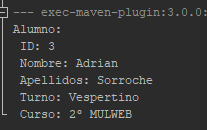
****

****

* + - Consulta Where: Alumno coincida con el turno pasado

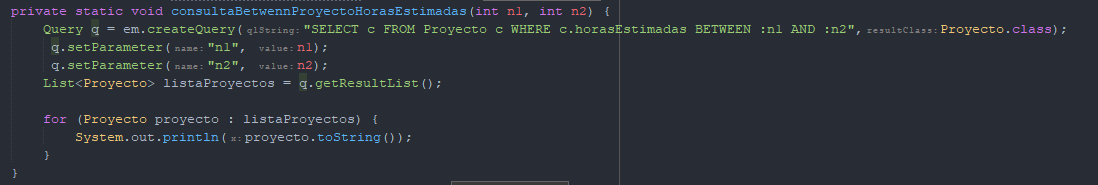
****

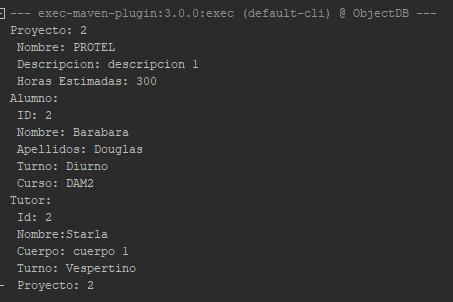
****

****

* + - Consulta Between: Mostrar proyectos que este en el rango de horas especificado

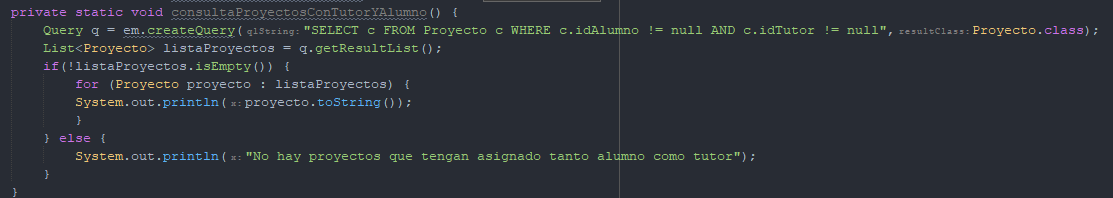
****

****

****

* + - Consulta AND: Mostrar proyectos con tutor y alumno asignado

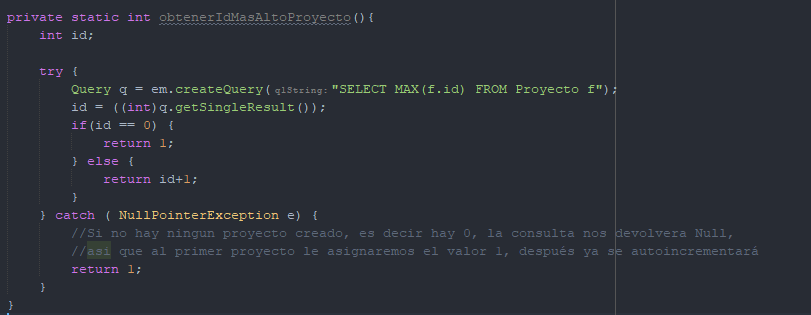
****

****

****

* + - Obtener el id mas alto del proyecto y sumarle 1

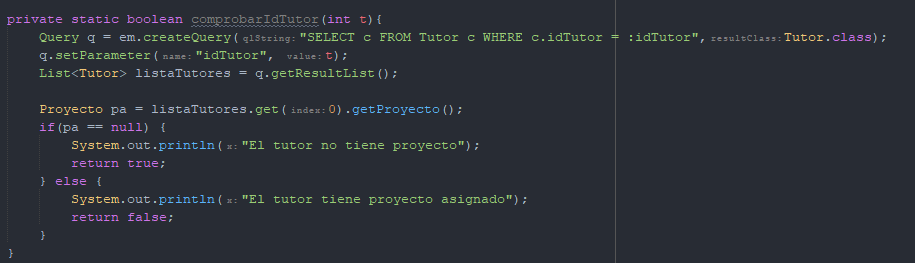
****

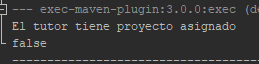
****

****

* + - Comprobar si un tutor tiene proyecto asignado

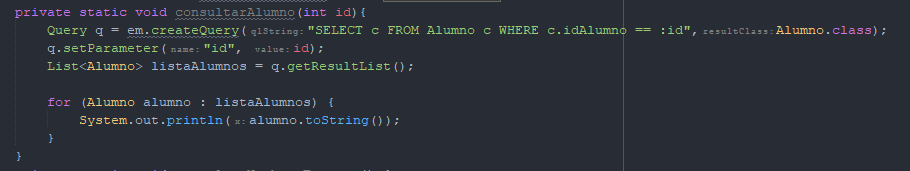
****

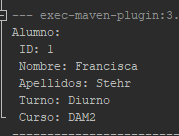
****

****

* + - Consultar un alumno por su id

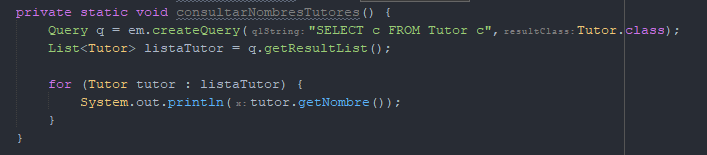
****

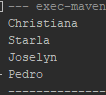
****

****

* + - Consultar los nombres de los tutores

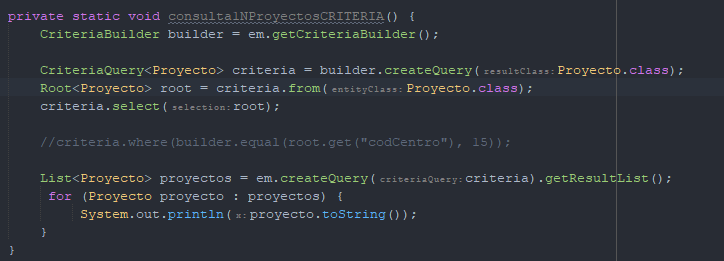
****

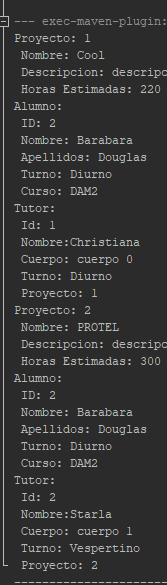
****

****

* + CRITERIA
    - Consulta 1:N

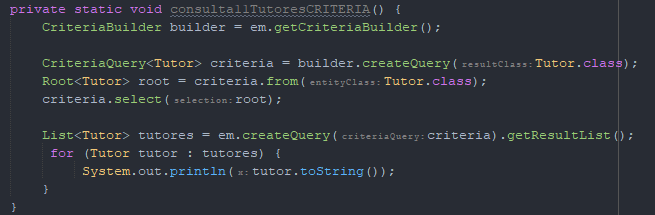
****

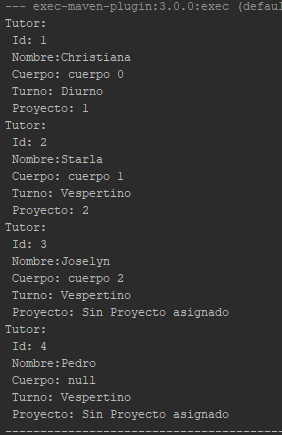
****

****

* + - Consulta 1:1

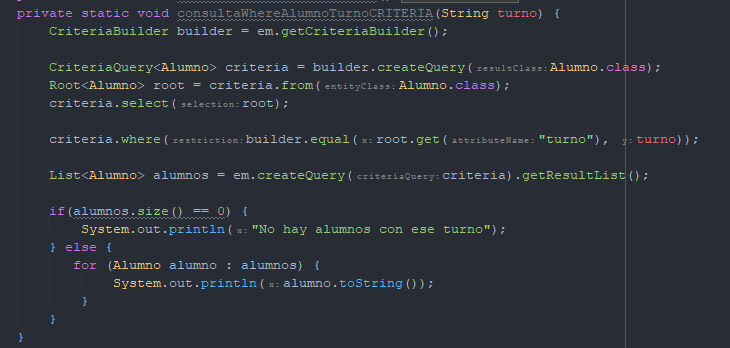
****

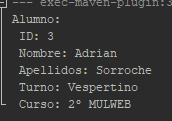
****

****

* + - Consulta Where

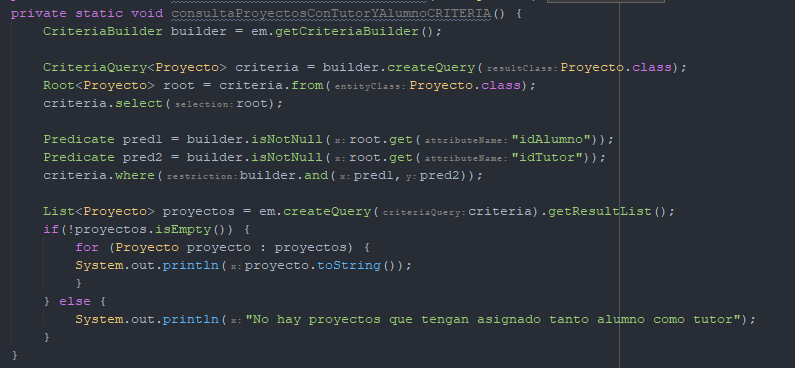
****

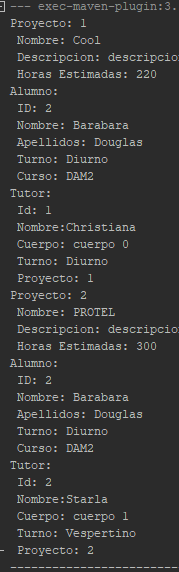
****

****

* + - Consulta AND

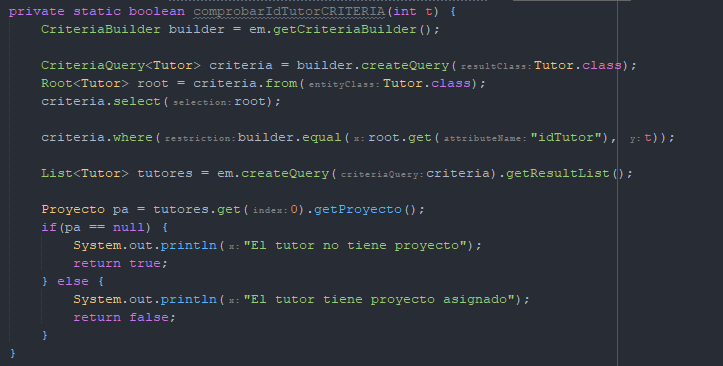
****

****

****

* + - Comprobar si un tutor tiene proyecto asignado

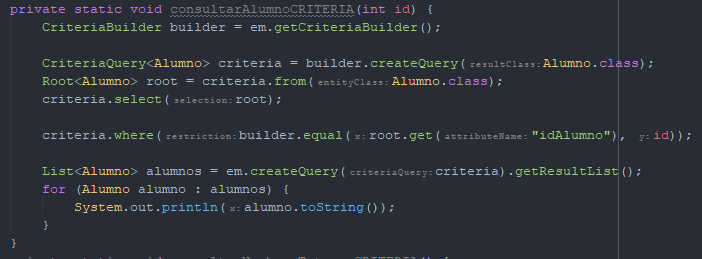
****

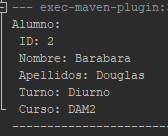
****

****

* + - Consulta alumno por id

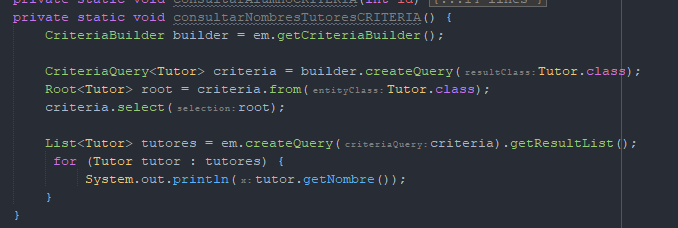
****

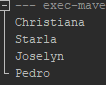
****

****

* + - Consultar los nombres de los tutores

****

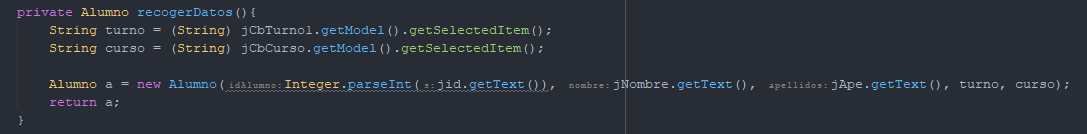
****

****

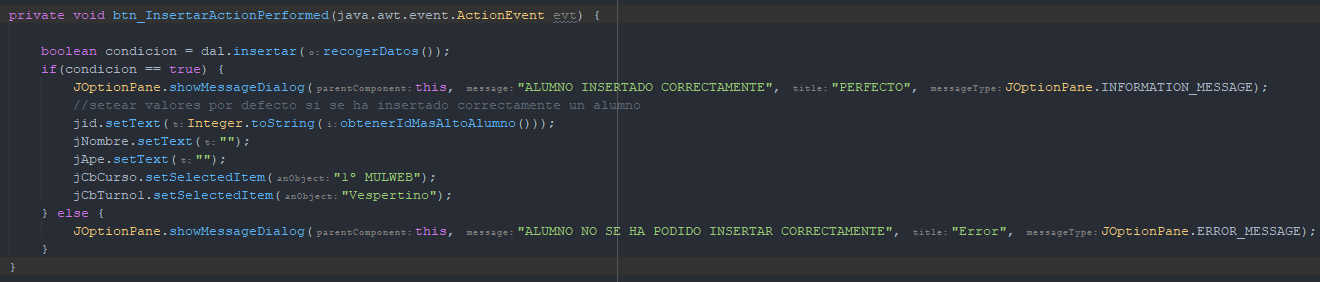
* **Swing**

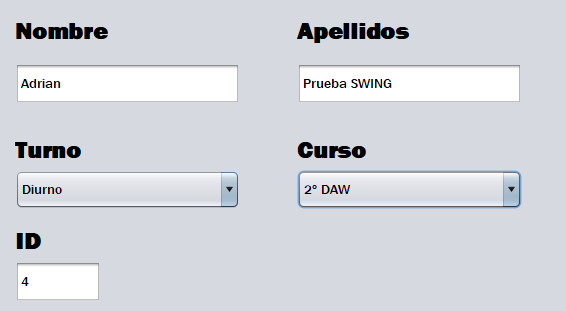
****

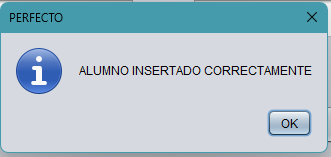
Recogemos los datos del swing y creamos un objeto alumno con dichos datos

****

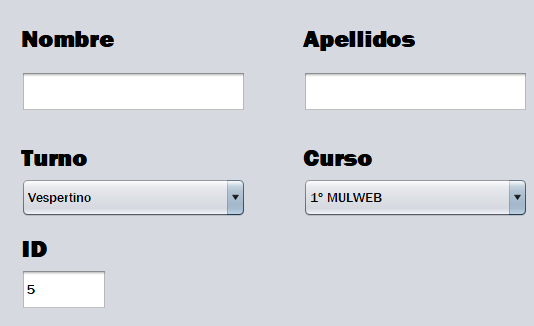
* + **INSERTAR**

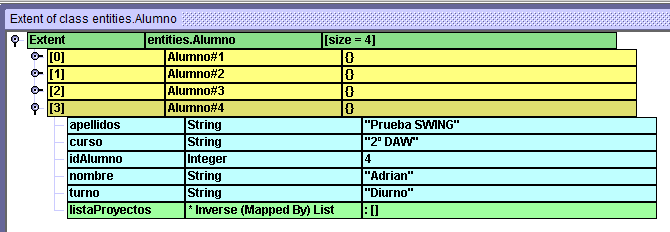
****

****

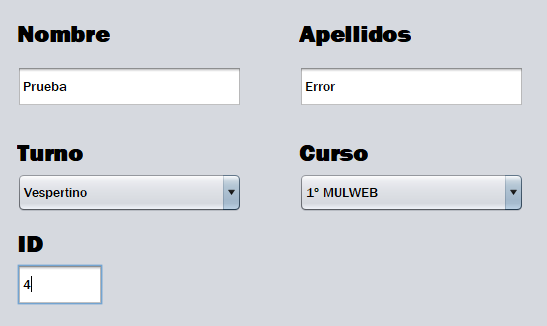
****

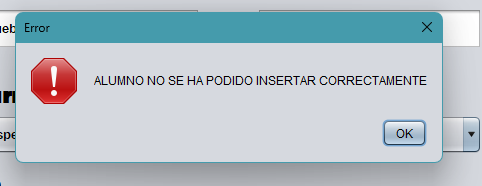
Se restablecen los valores por defecto y el Id aumenta ya que ahora hay un nuevo alumno, el id 5 será el del nuevo alumno en caso de que añadamos otro

****

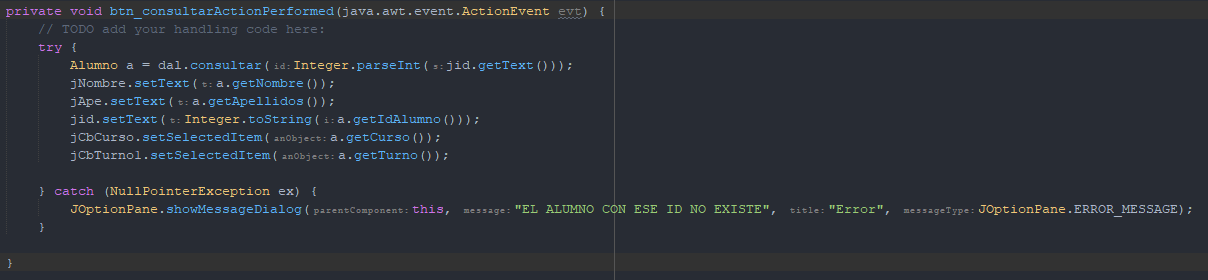
****

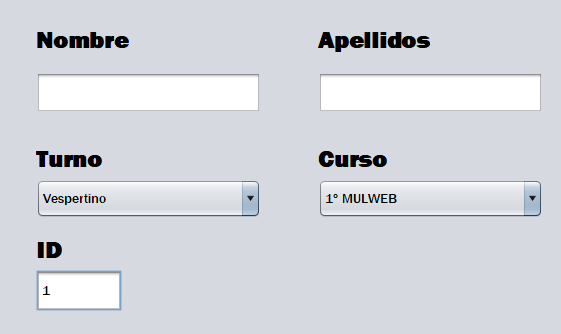
Error al insertar





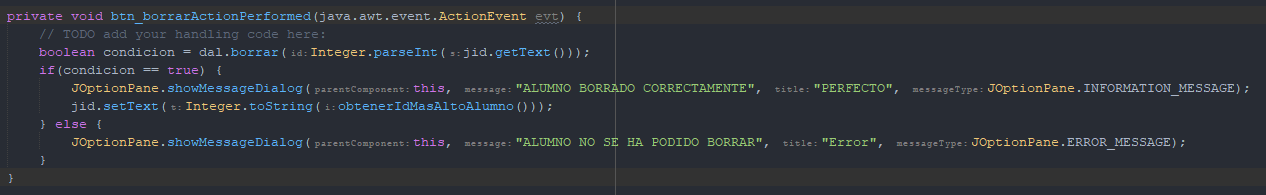
* + **CONSULTAR**

****

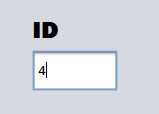
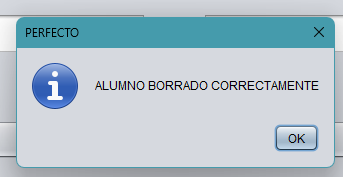
****

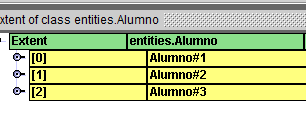
****

* + **BORRAR**

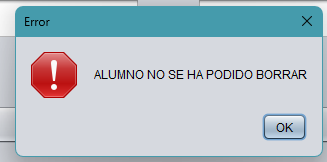
****

Vamos a borrar al alumno con id 4

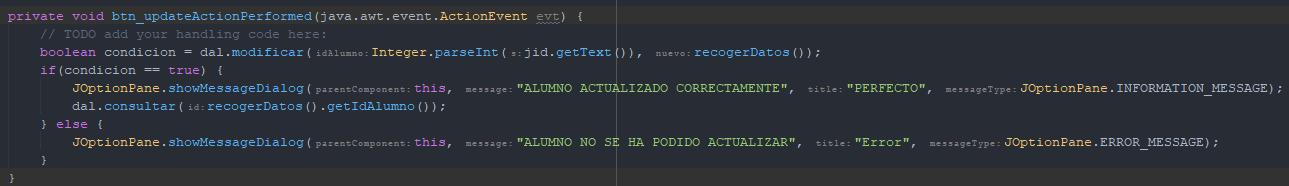




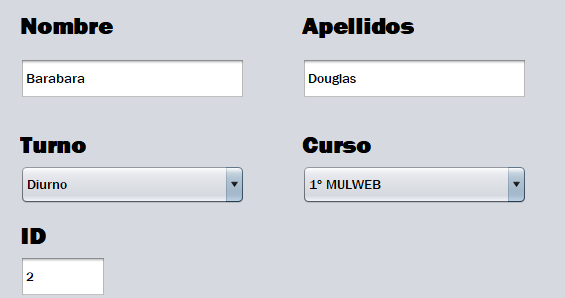
Error en el borrado al intentar borrar el id 4 de nuevo



* + **MODIFICAR**

****

Queremos modificar el alumno con id 2, primero consultamos los datos que tiene y luego modificamos el campo que nos interese



Le cambiamos el nombre

