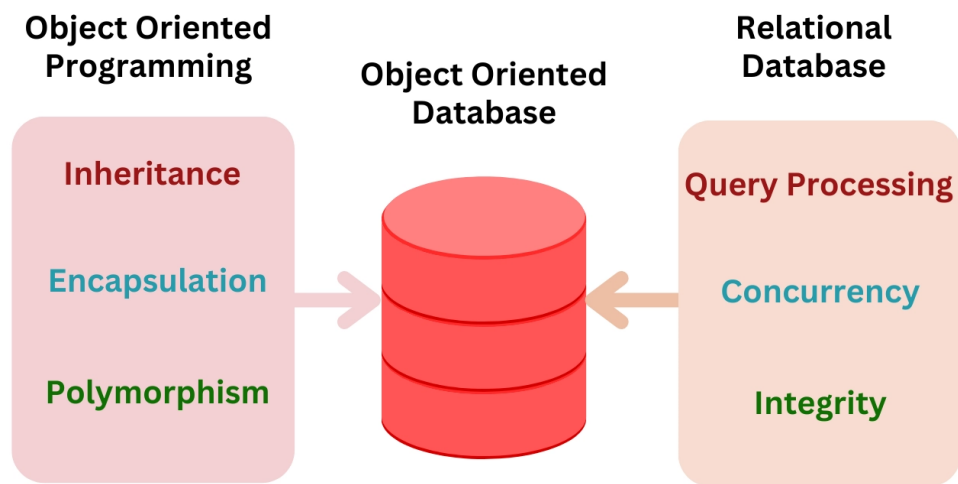


Taller UD11_2: CRUD con ObjectDB



1. **Introducción**

1. 1. [Pasos a seguir](#)

2. **Ejemplo con dos tablas**

3. **Actividades**

4. **Fuentes de información**

1. Introducción

Siguiendo con la misma base de datos que usaste el último taller de la unidad 10, vamos a realizar un mini-proyecto en el que puedas comprobar todo lo aprendido en la unidad 11. La idea básica es adaptar el proyecto del taller 3 de la unidad 10 a ObjectDB, pero solo para dos tablas (las que tienen alguna relación).

1.1. Pasos a seguir

Los pasos a seguir para realizar el proyecto serán los siguientes:

1. Genera al menos dos clases (una para cada tabla de tu futura DBO) con todas las etiquetas ObjectDB necesarias, añadiendo comentarios al lado de cada una de las etiquetas, justificando porque las estas usando.
2. Genera una o varias clases más para poder realizar un menú similar al de la unidad anterior, que te permita elegir de un menú principal cual de las dos tablas quieres gestionar.
3. Para cada una de las tablas deberás generar el CRUD completo (similar al tema anterior).
4. Solo para una de ellas, además deberás añadir la posibilidad de mantener la relación que existe entre ellas.

Consideraciones adicionales:

1. La BD se generará en una carpeta llamada `BD` dentro de la estructura de tu proyecto y puede ser un único fichero `odb`.
2. Para simplificar el proyecto no será necesario seguir los patrones vistos en la unidad anterior.
3. Recuerda que en el proyecto no solo es necesario programar, sino también comentar lo que vas haciendo.

2. Ejemplo con dos tablas

Para clarificar las tareas a realizar, seguiremos con la BD de starwars del tema anterior.

1. Generaremos dos tablas (characters/personajes y planets/planetas) con sus etiquetas para cada uno de sus campos y la relación que existe entre ellos, cada personaje tiene un planeta (y un planeta puede tener muchos personajes).
2. En mi caso generaremos una clase principal, con el menú principal que usará las clases del punto siguiente. Esta clase también se encargará de generar contenido en la DBO (en caso de que esté vacía en la primera ejecución)
3. Desarrollaremos una clase para gestionar personajes y otra para gestionar planetas.
4. Desde la clase para gestionar personajes, se permitirá elegir un planeta asociado al personaje (así implementamos la relación entre las dos tablas)

Más consideraciones:

1. El fichero de la BD se llamará `starwars.oddb` y se guardará en la carpeta BD de mi proyecto.
2. Aunque no es necesario, aplicar el patrón Singleton para acceder a la BD es relativamente sencillo e igual al de las BDR, así que unificaremos la conexión en una única clase.

3. Actividades

Genera un proyecto en `IntelliJ` con las características descritas, exporta el proyecto como `zip` y sube el proyecto a la tarea de AULES.

4. Fuentes de información

- [Wikipedia](#)
- [Programación \(Grado Superior\) - Juan Carlos Moreno Pérez \(Ed. Ra-ma\)](#)
- Apuntes IES Henri Matisse (Javi García Jimenez?)
- Apuntes AulaCampus
- [Apuntes José Luis Comesaña](#)
- [Apuntes IOC Programació bàsica \(Joan Arnedo Moreno\)](#)
- [Apuntes IOC Programació Orientada a Objectes \(Joan Arnedo Moreno\)](#)
- [FXDocs](#)
- <https://openjfx.io/openjfx-docs/>
- <https://arturoblasco.github.io/pr>