

TP2 - Ejercicio Integrador

Materia: Diseño de Sistemas de Software.

Alumnos:

- Barbagallon Cruz, Ignacio William Ariel.
- Luna, Demetrio.
- Luna, Roman.
- Piu, Florencia.

9. Considerar el diseño de un registro de estudiantes, con la siguiente información: nombres, apellido, edad, género, número de documento, ciudad de residencia, número de libreta universitaria, carrera(s) en la que está inscripto, antigüedad en cada una de esas carreras, y si se graduó o no.

- a. Diseñar el diagrama de objetos y el diagrama DER correspondiente.
- b. Implementar consultas para:
 - dar de alta un estudiante
 - matricular un estudiante en una carrera
 - recuperar todos los estudiantes, y especificar algún criterio de ordenamiento simple.
 - recuperar un estudiante, en base a su número de libreta universitaria.
 - recuperar todos los estudiantes, en base a su género.
 - recuperar las carreras con estudiantes inscriptos, y ordenar por cantidad de inscriptos.
 - recuperar los estudiantes de una determinada carrera, filtrado por ciudad de residencia.
- c. Generar un reporte de las carreras, que para cada carrera incluya información de los inscriptos y egresados por año. Se deben ordenar las carreras alfabéticamente, y presentar los años de manera cronológica.

Nota: las consultas deben ser resueltas mayormente en JPQL, y no en código Java.

El trabajo se realizó a partir de una implementación basada en patrones de diseño para la gestión de una base de datos en MySQL (v8.0.33), utilizando el lenguaje de programación Java en conjunto con Maven como herramienta de gestión de dependencias y construcción de proyectos. Se han empleado los patrones DAO (Data Access Object), Factory Method y Singleton con el objetivo de promover un diseño modular, flexible y reutilizable en la capa de acceso a datos.

El patrón DAO se ha utilizado para encapsular el acceso a la base de datos, permitiendo que la lógica de negocio esté separada de las operaciones de persistencia. Esto facilita la gestión de datos y mejora la mantenibilidad del sistema. El Factory Method se ha implementado para crear instancias de los objetos DAO de forma dinámica, proporcionando una forma flexible y escalable para manejar diferentes entidades en el sistema. Por último, el patrón Singleton se ha utilizado para garantizar que la conexión a la base de datos sea única en todo el sistema, mejorando el rendimiento al evitar la creación de múltiples conexiones innecesarias.

El proyecto también hace uso del framework Hibernate, para el mapeo objeto-relacional (ORM), lo que simplifica las interacciones con la base de datos MySQL. Adicionalmente, se han incluido dependencias como javax.persistence-api para garantizar la compatibilidad con la API de persistencia de Java, y Derby para realizar pruebas en un entorno de base de datos embebida.

El proyecto se encuentra en un repositorio de github en el siguiente [LINK](#).

Diagrama Objeto

El diagrama se encuentra en el siguiente [LINK](#).

Diagrama [DER](#)

