UNIVERSIDAD TECNOLOGICA NACIONAL

Trabajo integrador

Programación II

Alumnos

Gabriel De Los Rios

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Programación II

Docente Titular

Cinthia Rigoni

05 de diciembre de 2025

Objetivo

La finalidad de este trabajo práctico es poner en práctica todo lo aprendido respecto al lenguaje Java en el curso de Programación II incluyendo la persistencia de los datos utilizando una base de datos relacional, JDBC y el patrón DAO.

Arquitectura

El proyecto se divide en cuatro capas:

- UI: es la primera capa de interacción con el cliente, mostrando información sobre las opciones de navegación de la aplicacion y recibiendo y manejando entradas del usuario.

- Model: contiene los modelos que representan las distintas entidades que hacen parte del dominio de la aplicación (clientes, propiedades, etc).

- DAO: es la capa de acceso a datos de la aplicación, se encarga de las operaciones de persistencia interactuando con la base de datos.

- Service: desarolla la reglas del negocio, valida los datos de entrada y maneja los datos de salida, se encuentra en un nível intermedio entre las capas de UI y DAO.

Fundamento de las decisiones de diseño

UI

La capa de UI si implementa apoyandonos en la herencia debido a que todos los elementos de esta capa comparten su comportamiento, por una parte muestran las opciones y luego manejan la entrada del usuario bien sea delegando o señalando feedback al usuario (en caso de una entrada inválida). El menú funciona como una pila donde un menú puede motrsrar otro menú, volver al menú anterior, o delegar a la capa de servicio.

Model

Como se menciona anteriormente contiene una representación de clase de las entidades que forman parte del negocio de la aplicaión, las mismas no contienen validaciones, esta función se mantiene en la capa de servicio. Al igual que en la capa de UI, se utiliza una clase base con las propiedades comunes a todas las entidades.

DAO

Esta capa igualmente se organiza en una clase base GenericDAO que contiene algunas implementaciones para las operaciones comunes de persistencia (findById, findAll, delete) y la declaracion de las operaciones restantes para ser implementadas por las clases que la extiendan. Inpirados en el patron de ActiveRecord, se utiliza el nombre de los modelos como nombre de las tablas esto permite la implementación de ciertas operaciones de forma genérica.

Igualmente se implemento una clase Activable que contiene la logica referida a activar y desactivar (baja lógica) de las entidades, pero manteniendola como una cualidad opcional para las entidades que forma parte del programa.

Al no utilizar conceptos avanzados como Reflection (que hubiese permitido mapear los nombres de las propiedades a los nombres de las columnas) se implementa un metodo mapper para esta tarea, cada clase implementa su propio mapper como una interfaz entre el modelo y su tabla correspondiente en la base de datos.

Service

La capa de servicio esta organizada por entidad, cada servicio se ocupa de getionar una clase, interactua con el usuario poniendo a sus disposición todos los metodos necesarios para las operaciones CRUD, válida sus entradas y delega a la capa DAO la ejecución de las instrucciones en la base de datos, luego procesa la respuesta y genera la retroalimentación para el usuario con el estado final de la operación (éxito o error).

Todas las clases de esta capa implementan la interfaz GenericService que obliga a implementar un método para cada operación CRUD, adicionalmente se genera una interfaz Activable que obliga a implementar los métodos para activar/desactivar una entidad (baja lógica).

Otros

Se desarrolla un paquete **config** de configugración que contiene aspectos relacionados a la iniciación del proyecto como la conexión a la base de datos y la posterior creación de las tablas si fuese necesario, la creación del pool de conexiones con hikari, etc.

El paquete **utils** contiene clases de utileria como los validadores para las entradas del usuario, los textos comunes para facilitar su gestión, etc.

Conclusión

Este proyecto emplea todos los conceptos que se estudiaron durante este curso incluyendo herencia, polimorfismo, sobrecarga, metodos toString y equals, etc. Se realiza una aplicación en capas que garantiza la escalabilidad y mantenibilidad del proyecto. Adicionalmente se utiliza la persistencia usando el patón DAO y JDBC, empleando librerias externas y permitiendo asi integrar la aplicación a una base de datos.

Bibliografía consultada para este proyecto:

Java Documentation
https://docs.oracle.com/en/java/

StackOverflow
https://stackoverflow.com/questions

PROGRAMACIÓN desde -10 en Java
https://www.youtube.com/watch?v=saSW7TLoOqk&list=PLOw7b-NX043b7ZhI-WkG-R9J0RptkCwKU

Cómo escriturar una casa y cuanto cuesta en 2025
https://www.zonaprop.com.ar/blog/escriturar-casa/