Imagen que contiene dibujo, alimentos, señal

Descripción generada automáticamente

PROGRA II

Alumno: Josué Gabriel Quesada Angulo.

Profesor: Juan Gamboa Abarca

Año:2024

Documentación del CRUD

INDICE

[Introducción. 3](#_Toc169087305)

[Objetivo 4](#_Toc169087306)

[Arquitectura del CRUD 5](#_Toc169087307)

# Introducción.

Este proyecto tiene como propósito proporcionar una librería que facilite el mapeo automático de las clases de la base de datos, específicamente para Oracle XE 21c, MongoDB y MySQL.

* Oracle XE 21c es una versión gratuita de la base de datos Oracle, que ofrece una amplia gama de características avanzadas para la gestión de datos, tales como soporte para transacciones, integridad referencial y capacidades de consulta avanzada. La integración de esta librería con Oracle XE permite a los desarrolladores automatizar la creación de tablas y la inserción de datos, simplificando significativamente el trabajo con esta poderosa base de datos.
* MongoDB es una base de datos NoSQL ampliamente utilizada para manejar datos no estructurados o semi-estructurados. Su modelo de datos basado en documentos y su capacidad para escalar horizontalmente lo convierten en una opción popular para aplicaciones modernas que requieren flexibilidad y rendimiento. Esta librería proporciona métodos para mapear automáticamente clases Java a documentos de MongoDB, facilitando la inserción y gestión de datos en este entorno.
* MySQL es uno de los sistemas de gestión de bases de datos relacionales más populares del mundo, conocido por su rendimiento, fiabilidad y facilidad de uso. Aunque actualmente no está incluido en la librería, se planea integrar soporte para MySQL en futuras versiones del proyecto. Esto permitirá a los usuarios aprovechar las capacidades de MySQL junto con las ya soportadas Oracle XE y MongoDB.

# Objetivo

El objetivo de esta librería es simplificar al usuario del proceso de mapeo de clases a diferentes bases de datos, permitiendo al usuario centrarse en desarrollar las clases o objetos para transferirlos a la base de datos, centrándose en la lógica sin preocuparse por los detalles específicos de cada DBMS, esta librería proporciona una API unificada y muy fácil de utilizar, para realizar operación de CRUD (Crear, Leer, Actualizar, Eliminar) en las bases de datos soportadas.

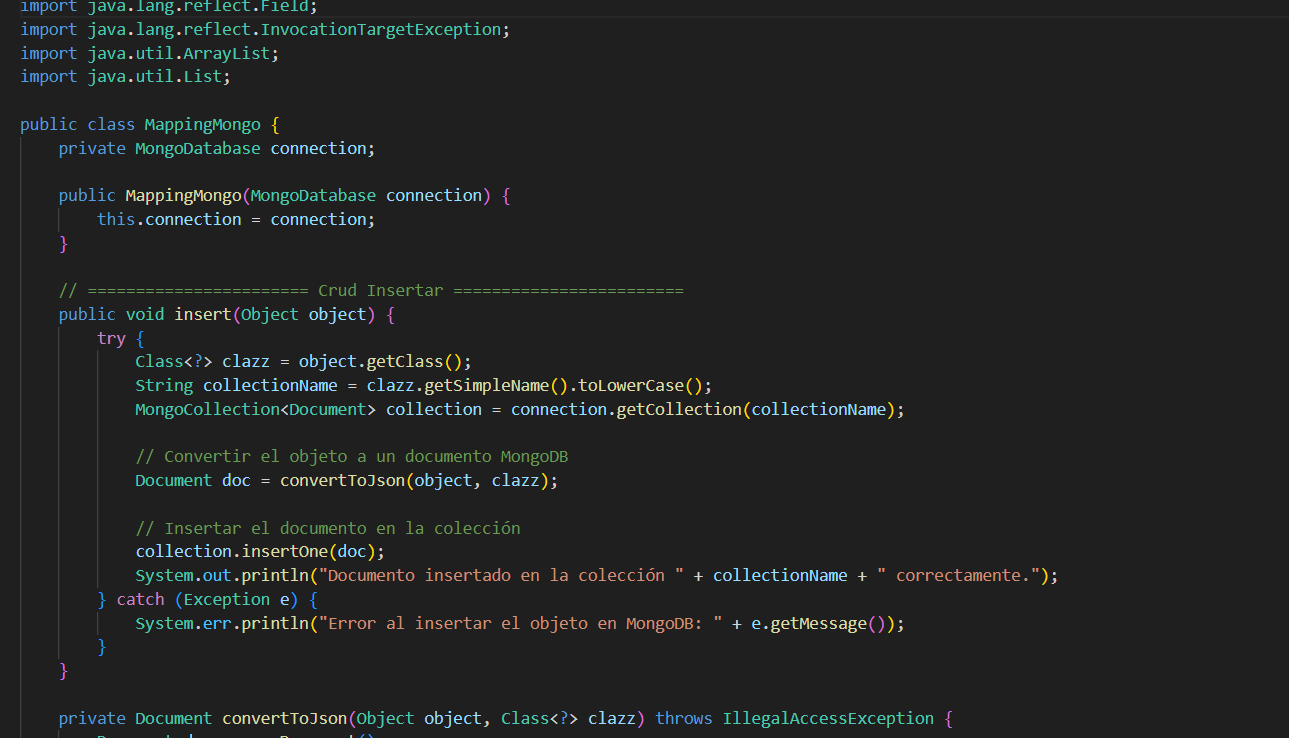
Este documento presenta una guía detallada sobre cómo se utiliza esta librería, incluyendo la configuración de las dependencias, la conexión de cada base de datos y la realización de la operaciones del CRUD, se proporciona ejemplos prácticos para ilustrar cada funcionalidad, permitiendo que el usuario pueda implementar rápidamente esta librería a su proyectos.

# Arquitectura del CRUD

La clase llamada (DatabaseConnection) se encargar de hacer las conexiones en las 3 bases de datos Oracle, Mongo y MySQL.



En las siguientes clases está separado por cada base de datos, ya que su estructura es similar más algunos comandos son diferentes, para que este ordenado separe la estructura por cada base de datos, tomar en cuenta que para cada acción del CRUD se tiene que hacer una conexión antes.



Esta clase **MappingMongo** primero se encarga de hacer una conexión a la base de datos, después se muestran las acciones del CRUD, la cual se le pasara por parámetros lo que se quiera hacer.

En el Main se utilizara la acción de la base de datos que quiera utilizar ya sea eliminar datos por id, actualizar datos por id, como crear objetos.

Texto

Descripción generada automáticamente

En esta clase llamada **MappingMySQL** se encarga primero de hacer una conexión a la base correspondida, después de hacer la conexión puede ejecutar cualquier función del CRUD ya sea eliminar, actualizar, insertar, o mostrar la cual se llamaran desde el Main.

Texto

Descripción generada automáticamente

En esta clase llamada **MappingOracle** se encargar primero de hacer una conexión a la base de datos correspondiente, para ejecutar las funciones del CRUD , es importante recalcar que estas funciones se ejecutaran en el Main, ya que se pasaran por parámetro lo que se quiera eliminar o insertar entre otros.

Estas clases próximas también fueron separadas, para tener un código limpio y que sean pruebas para cada base de datos por separado, teniendo una visualización de lo que hace el código y como se podría modificar a futuro.

* **TestMongo**

Texto

Descripción generada automáticamente

Esta clase se encarga de hacer el Test a la clase MappingMongo, las funciones de esta clase nada mas es para pasare por parámetro los objetos que se quieran insertar, actualizar, eliminar entre otras funciones del CRUD es importante recalcar que, a la hora de crear un objeto, se mostrara el ID de ese objeto para que a futuro si quiera actualizar, eliminar ese objeto sea por el ID de ese mismo.

* **TestMySQL**

Texto

Descripción generada automáticamente

Esta clase igualmente que la anterior se encargar de hacer un Test al CRUD, pero está en específico lo hace con la base de datos de MySQL, esta clase llamada a MappingMySQL para que funcione las funciones del Test, si se ve bien pueden notar que reciben los objetos o el ID por parámetro.

* **TestOracle**

Texto

Descripción generada automáticamente

Son muy similares a las clases anteriores más la estructura y los objetos que pasan por parámetro pueden cambiarse, ya sea que quiero ingresar otro objeto a la tabla o buscar algo en específico.

* Objetos

Mascota

Texto

Descripción generada automáticamente

Trabajador

Texto

Descripción generada automáticamente

Estos objetos como pueden ver, deben tener su estructura como objeto y un ID para cuando sea creado el objeto, sea identificado con identificador único.

Para que más adelante si se quiere eliminar o actualizar ese objeto Utilice una función del CRUD de la base de datos que se está utilizando y la verifique en el Test de esa base de datos y se utiliza, es muy importante recalcar que para utilizar este CRUD deben verificar primero que el usuario de esa base de datos pueda crear tablas, para que no haya errores.

En resumen, este CRUD tiene lo básico para utilizar una base de datos, se utiliza simple, se crea un objeto y en el Test de esa base de datos, pasa el objeto y en el main ejecuta esa función que quiera hacer, ya sea crear ese objeto nuevo o eliminar un objeto ya existente como también actualizarlo o mostrar los objetos de esa tabla.

**pom.xml**

este archivo es importante ya que, si esto no pueden hacer ninguna conexión en ninguna base de datos, recordar que cada vez que quieran agregar una base de datos al CRUD primero hay que ingresar a este archivo y poner su dependencia de esa base de datos en específico, ejemplo:

