UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”

“PROGRAMACIÓN ORIENTADA A OBJETOS”

Actividad Autónoma N° 01

Aplicación de Gestión de Tareas con Interfaz Gráfica y Base de Datos NoSQL

Tema de la Actividad:

Creación de una aplicación que permita gestionar tareas con una interfaz gráfica y base de datos NoSQL, implementando principios SOLID.

Integrantes Grupo # 06

Mónica Carolina Angamarca Cela.

Vanessa Fernanda Chiriguaya Ruales.

Luis Alejandro Sánchez Durán.

NRC: 1322

Tutor asignado:

Mgtr. Luis Enrique Jaramillo

Quito, miércoles 26 de febrero del 2025

1.- Introducción

En la actualidad, la información ha adquirido un enorme poder, gracias en gran medida a su sencillo y fácil acceso, siendo reconocida su importancia en la sociedad hasta por encima del dinero, razón por la cual expertos hablan acerca de la "Era de la Información".

Desde los años 70s se han desarrollado las bases de datos como herramientas para el almacenamiento de información utilizando un modelo relacional, sin embargó debido al incremento de información y los requerimientos actuales como la velocidad, la variedad o el volumen en los datos, ha sido necesario una evolución en tales bases. Hacia finales del 2009 Eric Evans2 menciona el término "NoSQL" haciendo referencia a una "fuente abierta, para el almacenamiento de datos no relacionales" afirmando que el término más correcto sería NoREL (Not Only Relational). Este tipo de almacenamiento no pretende desplazar a las bases de datos relacionales, si no que sirve como una herramienta útil en ciertos entornos donde se busque velocidad y rendimiento. Hoy en día, ya es utilizada por muchas empresas reconocidas, entre estas redes sociales de gran importancia como Facebook y Twitter. En el escenario de la información, muchos de los sectores económicos requieren el apoyo para la gestión de los grandes volúmenes de datos, uno de estos, es el transporte, una actividad fundamental dentro de la sociedad, encaminada a la prestación de servicios, donde el nuevo enfoque propuesto por las bases de datos NoSQL generan una forma más eficiente para el manejo en este sector.

2.- Objetivos:

2.1.- Objetivo general

- Desarrollar una aplicación que permita a los usuarios gestionar sus tareas de manera eficiente, utilizando una interfaz gráfica amigable y una base de datos NoSQL para el

almacenamiento y recuperación de datos, todo ello bajo la guía de los principios SOLID para garantizar un código robusto y mantenible.

2.2.- Objetivos específicos

- Diseñar una interfaz gráfica intuitiva y fácil de usar que permita a los usuarios crear, editar, eliminar y organizar sus tareas de manera eficiente.
- Implementar una base de datos NoSQL para el almacenamiento y recuperación de datos, aprovechando las ventajas de este tipo de bases de datos en cuanto a escalabilidad y flexibilidad.
- Aplicar los principios SOLID en el diseño y desarrollo de la aplicación para asegurar un código modular, extensible y fácil de mantener.

3.- Marco Teórico

3.1. Gestión de Tareas

La gestión de tareas es el proceso de planificar, organizar, seguir y completar tareas. Una aplicación de gestión de tareas eficiente debe permitir a los usuarios crear tareas, asignarles prioridades, establecer fechas límite, realizar un seguimiento de su progreso y categorizarlas.

3.2. Bases de Datos NoSQL

Las bases de datos NoSQL son un tipo de base de datos que no se adhiere al modelo relacional tradicional. En su lugar, utilizan una variedad de modelos de datos, como documentos, grafos o clave-valor. Las bases de datos NoSQL son ideales para aplicaciones que requieren alta escalabilidad y flexibilidad, como la gestión de tareas.

3.3. Principios SOLID

SOLID es un acrónimo que representa cinco principios de diseño en programación orientada a objetos y son fundamentales en el diseño de software.

1. Single Responsibility Principle (SRP):

- Cada clase tiene una única responsabilidad. Task maneja los datos de la tarea, Database maneja las operaciones de la base de datos, y TaskManagerApp maneja la interfaz gráfica.

2. Open/Closed Principle (OCP):

- Las clases están abiertas para extensión pero cerradas para modificación. Por ejemplo, puedes extender Task para agregar más atributos sin modificar la clase base.

3. Liskov Substitution Principle (LSP):

- Las subclases pueden sustituir a las clases base sin alterar el comportamiento del programa. Task puede ser sustituida por cualquier subclase que mantenga su comportamiento.

4. Interface Segregation Principle (ISP):

- Las clases no dependen de métodos que no usan. La interfaz gráfica no depende de métodos de Database que no usa.

5. Dependency Inversion Principle (DIP):

- Las clases dependen de abstracciones, no de concreciones. TaskManagerApp depende de lEl sistema de bases de datos MongoDB está especialmente diseñado para el almacenamiento de datos de gran volumen, por lo que es adecuado para trabajar con grandes conjuntos de datos distribuidos.

3.4. Programa Mongol

MongoDB es un sistema de gestión de bases de datos para documentos multiplataforma y «source-available» que se enmarca dentro de la categoría de bases de datos no relacionales o

NoSQL. Los registros de MongoDB, desarrollada por MongoDB Inc., funcionan como objetos JSON con esquemas opcionales.

En lugar de utilizar tablas y filas como las bases de datos relacionales clásicas, MongoDB utiliza colecciones y documentos consistentes en pares clave-valor, que son las unidades de datos básicas en MongoDB; las colecciones, por su parte, contienen conjuntos de documentos.

El sistema de bases de datos MongoDB está especialmente diseñado para el almacenamiento de datos de gran volumen, por lo que es adecuado para trabajar con grandes conjuntos de datos distribuidos.

MongoDB es compatible con numerosos lenguajes de programación, incluyendo C, C++, C#, Go, Java, Python, Ruby y Swift. Sus funcionalidades incluyen consultas «ad hoc», indexación, balanceo de carga, agregación, ejecución de JavaScript en el servidor, alta disponibilidad mediante funciones de replicación y failover integradas, y escalabilidad horizontal con «sharding» nativo.

4.- Metodología:

El desarrollo de la aplicación se llevó a cabo siguiendo una metodología ágil, con iteraciones cortas y frecuentes. Se utilizó un enfoque de diseño centrado en el usuario para garantizar que la interfaz gráfica fuera intuitiva y fácil de usar.

La base de datos NoSQL elegida fue MongoDB, debido a su flexibilidad y escalabilidad. Se aplicaron los principios SOLID en el diseño y desarrollo de la aplicación para asegurar un código modular, extensible y fácil de mantener.

Las tecnologías que hemos utilizado para el desarrollo de nuestro trabajo son:

- Lenguaje de Programación: Java
- Base de Datos NoSQL: MongoDB

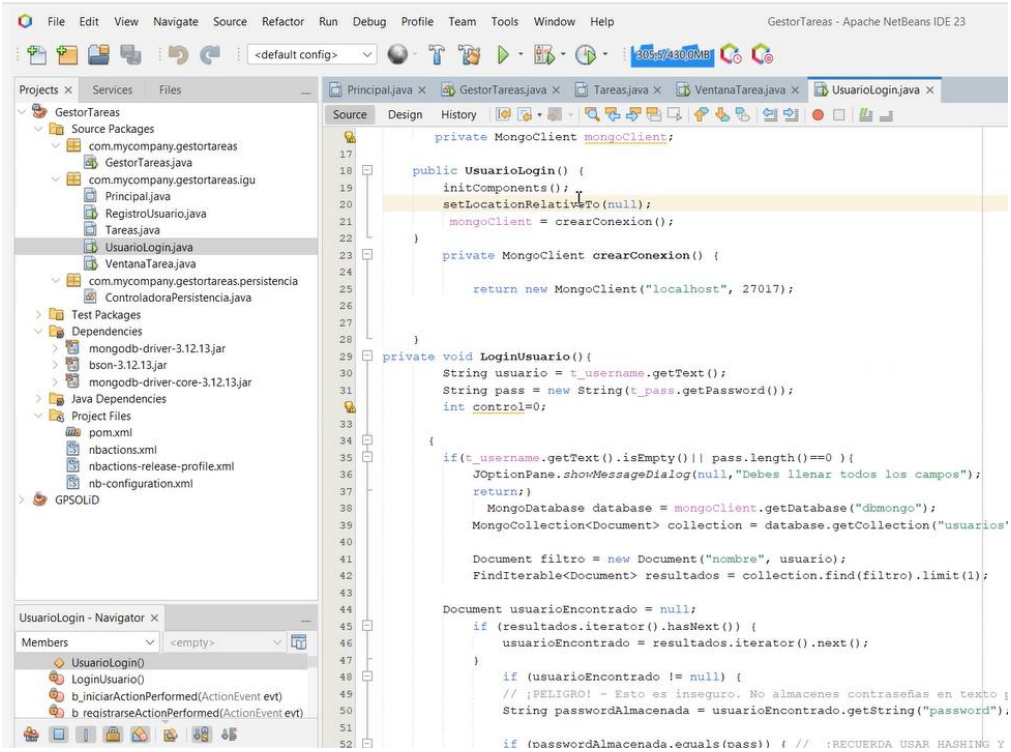
Diagrama UML

Este diagrama UML proporciona una visión general de la estructura de la aplicación y cómo las clases interactúan entre sí.

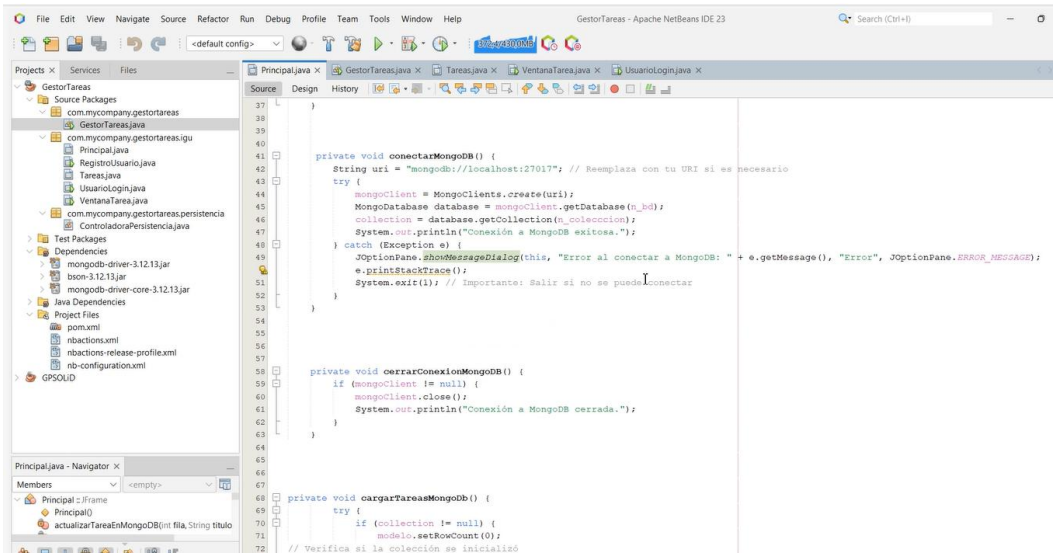
Estructura del Proyecto

Nuestro proyecto está organizado en módulos para seguir los principios SOLID:

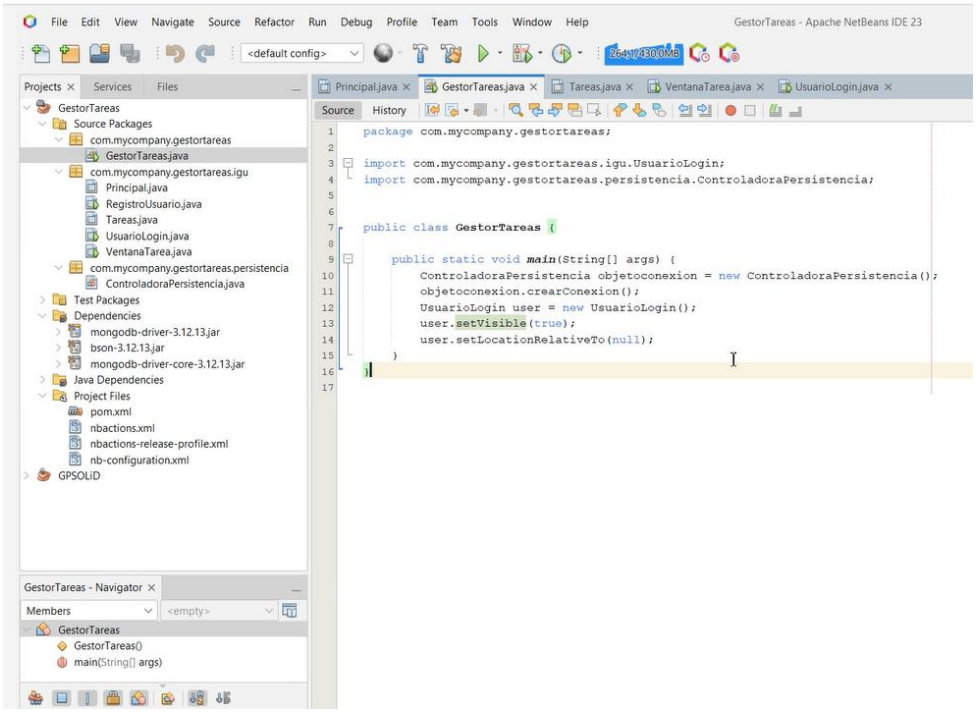
- **Usuario login.java**



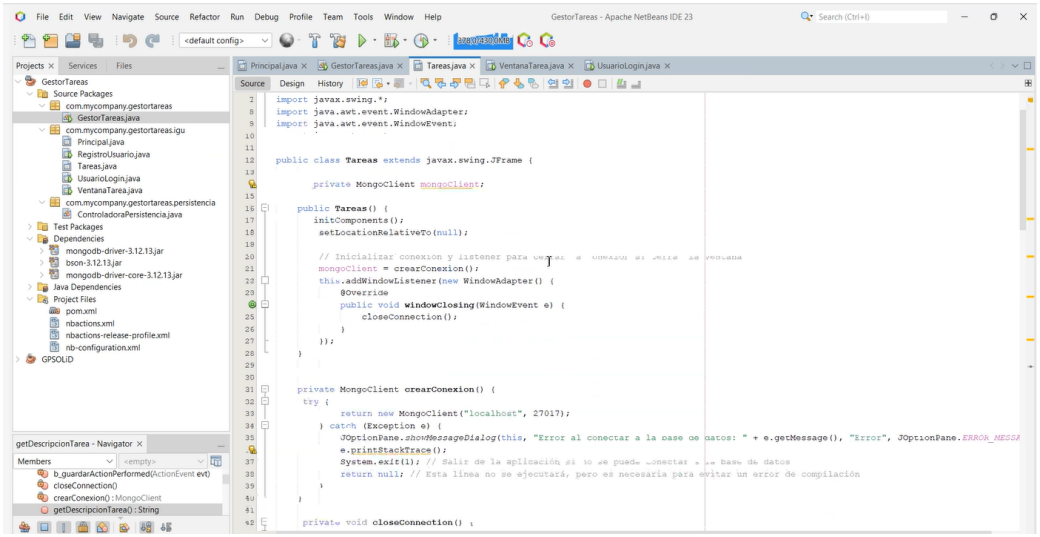
- **Principal.java**



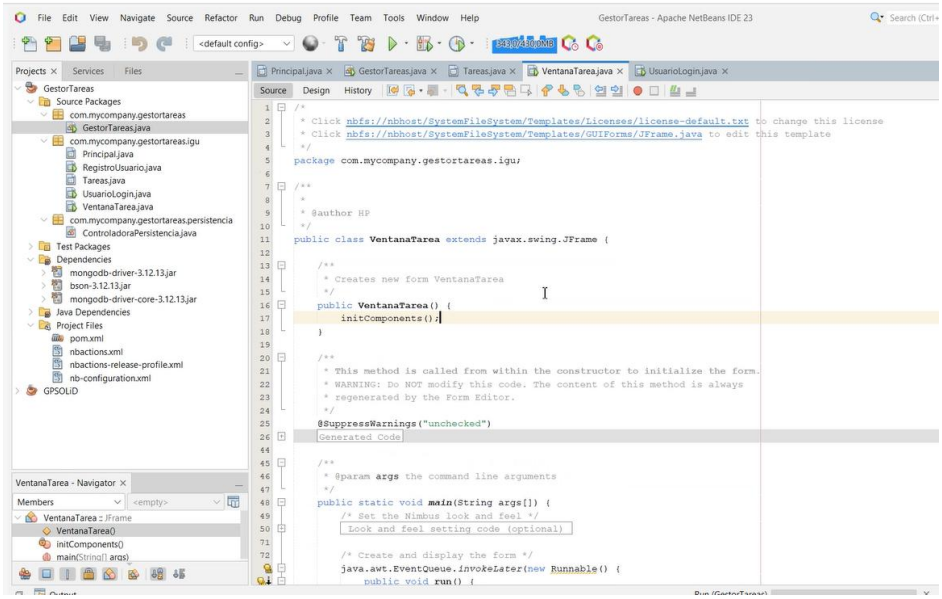
- **Gestor de tareas,java**



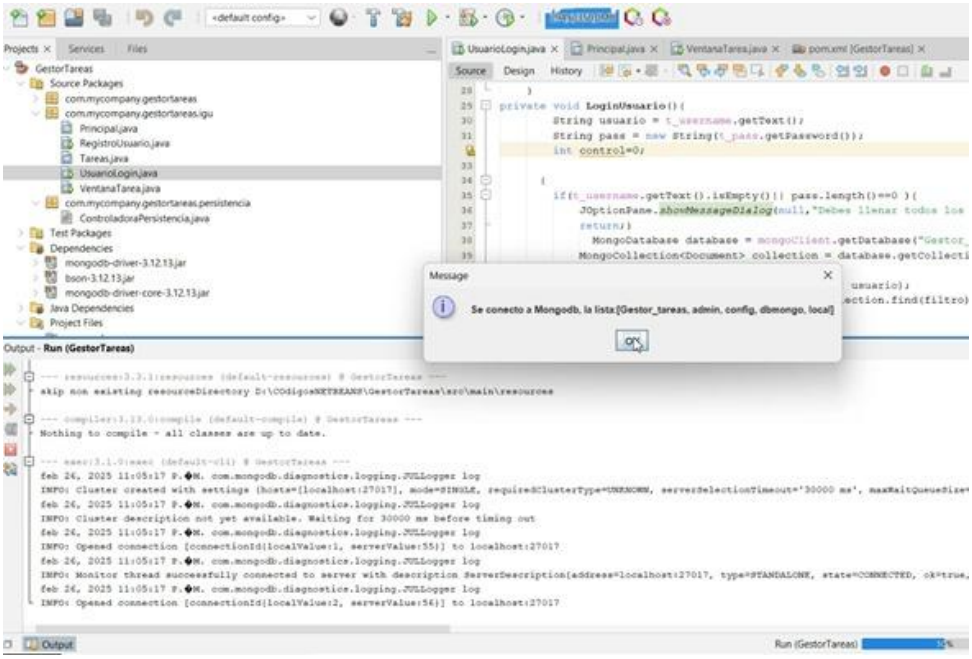
- **Tareas.java**



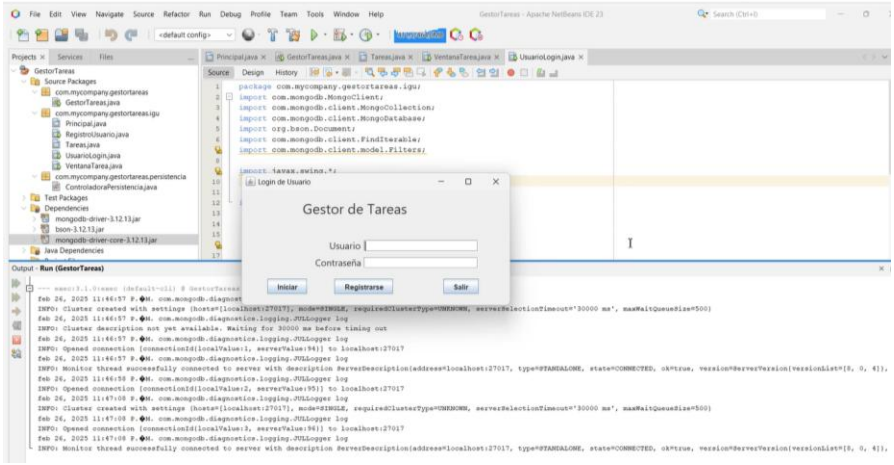
- **VentanaTarea.java**



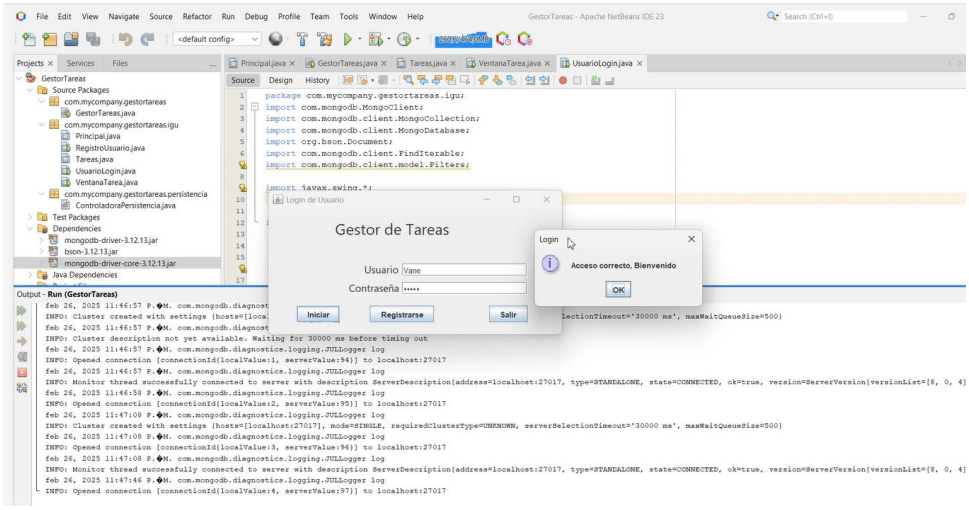
- **<u>Conexión Mongo:</u>**



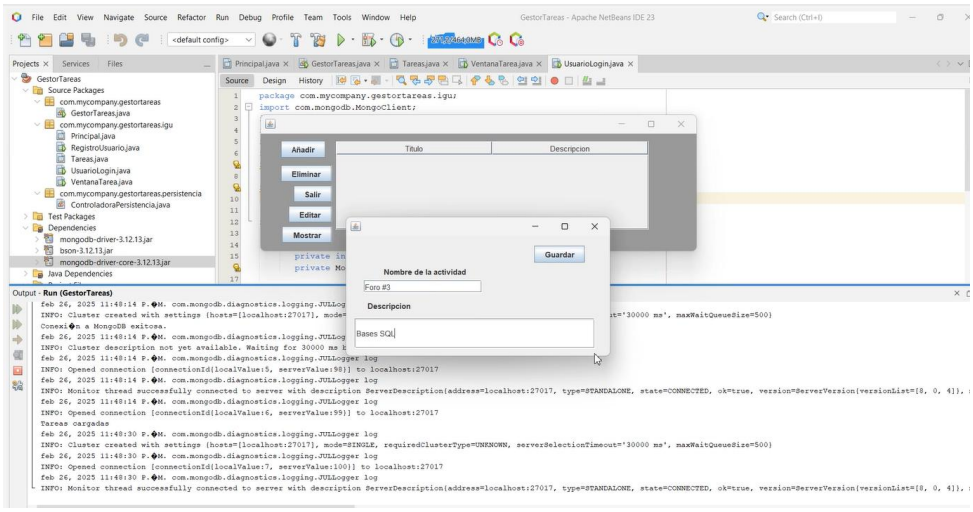
- **Login**



- **Ingreso usuario:**



- **Creación de tareas:**



5.- Recomendaciones:

Lo que como grupo de trabajo podríamos recomendar es que al momento de realizar este proceso analicemos mejorar la Interfaz Gráfica, considerando usar un framework más avanzado como PyQt o Kivy para una interfaz más moderna y atractiva.

También creemos importante tomar en cuenta incrementar la seguridad y autenticación, pues esto implementa un sistema de autenticación para que los usuarios puedan iniciar sesión y gestionar sus propias tareas.

Realizar pruebas que sean necesarias, así como mantenimientos para asegurar que cada componente de la aplicación creada funcione correctamente y con ello podamos documentar el código para facilitar el mantenimiento y la expansión futura de así requerirse.

6.- Conclusiones:

La aplicación de gestión de tareas desarrollada nos muestra que el trabajo sigue los principios SOLID, lo que asegura un diseño de software robusto, mantenible y escalable. La combinación de una interfaz gráfica amigable y una base de datos NoSQL nos proporciona una solución eficiente para la gestión de tareas que deseábamos crear.

La adopción de los principios SOLID representa una inversión significativa en la calidad y sostenibilidad del desarrollo de software. A través de la exploración y aplicación práctica de cada principio, los desarrolladores pueden alcanzar un mayor entendimiento y apreciación por un diseño de software que no solo cumple con los requisitos actuales de manera eficaz, sino que también facilita la adaptación y crecimiento futuro.

7.- Referencias Bibliográficas:

Martin, R. C. (2002). Agile software development, principles, patterns, and practices. Prentice Hall.

Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley Professional.

Chodorow, K., & Dirolf, M. (2010). MongoDB: The definitive guide. O'Reilly Media, Inc.

MongoDB, Inc. (2023). MongoDB documentation. Retrieved from https://docs.mongodb.com/