Informe Final Integrado

1. Resumen Ejecutivo

Este informe detalla el proceso de diseño, implementación y validación de una plataforma de gestión de proyectos potenciada por inteligencia artificial. A lo largo del desarrollo, se aplicaron principios y patrones de diseño para garantizar una arquitectura modular, mantenible y escalable. Además, se realizaron pruebas unitarias para verificar la calidad del software.

2. Justificación de Decisiones Técnicas y Arquitectónicas

Para el desarrollo del proyecto, se optó por **Java con Maven**, debido a su estabilidad y compatibilidad con múltiples frameworks. Se utilizaron los principios **SOLID, DRY, KISS y YAGNI** para asegurar un código limpio y eficiente. Además, se implementaron los patrones de diseño **Factory Method, Facade y Observer** para estructurar mejor los componentes del sistema y facilitar su mantenimiento.

En cuanto a la arquitectura, se adoptó un enfoque modular que permite una posible transición a microservicios en el futuro, manteniendo por ahora una estructura monolítica bien organizada.

3. Documentación de Actividades

Actividad 1: Análisis y Selección de Principios de Diseño

Se analizaron y aplicaron los principios SOLID, DRY, KISS y YAGNI en la arquitectura del sistema. Se priorizó la reutilización de código y la claridad en la estructura del proyecto.

Actividad 2: Identificación y Justificación de Patrones de Diseño

Se seleccionaron tres patrones de diseño adecuados para la plataforma:

- **Factory Method:** Para la creación flexible de diferentes tipos de tareas.
- **Facade:** Para simplificar el acceso a los servicios principales.
- **Observer:** Para la gestión de notificaciones automáticas a los usuarios.

Actividad 3: Esquema de la Arquitectura del Software

Se diseñó un **diagrama de clases UML**, reflejando los principales componentes y sus relaciones. La estructura favorece la separación de responsabilidades y la extensibilidad.

Actividad 4: Implementación Práctica de Patrones de Diseño

Se implementaron los patrones seleccionados en Java con Maven, organizando el código en paquetes específicos para cada patrón.

Actividad 5: Estrategia de Control de Calidad y Pruebas Unitarias

Se crearon pruebas unitarias con **JUnit 5**, verificando el correcto funcionamiento de la fábrica de tareas, la gestión de proyectos y el sistema de notificaciones.

Actividad 6: Documentación de Pruebas y Corrección de Errores

Se corrigieron problemas en la configuración de Maven para ejecutar correctamente las pruebas. También se mejoraron los mensajes de salida para hacerlos más visibles.

Actividad 7: Análisis Comparativo con Arquitecturas Industriales

Se comparó la arquitectura desarrollada con soluciones reales como **Microservicios y MVC**. Se concluyó que la solución es adecuada para un sistema en crecimiento, pero podría beneficiarse de mayor separación de servicios en el futuro.

Repositorio

- **Repositorio GitHub:** [https://github.com/dxn1l/GestorProyectosIA]

4. Conclusiones y Aprendizajes

- **Modularidad y escalabilidad:** Se logró un diseño estructurado y fácil de extender en el futuro.
- **Importancia de las pruebas unitarias:** Detectamos errores tempranos y mejoramos la confiabilidad del software.
- **Comparación con la industria:** Nuestra solución es adecuada, pero puede evolucionar a una arquitectura más distribuida si el proyecto crece.

En general, este proceso permitió consolidar buenas prácticas en el desarrollo de software y fortalecer el uso de patrones de diseño para la creación de sistemas eficientes y mantenibles.