Actividad 7: Análisis Comparativo con Arquitecturas Industriales

Unax Aller

15/03/2025

La arquitectura que hemos diseñado para nuestra Plataforma Inteligente de Gestión de Proyectos está hecha con un enfoque modular que sigue el patrón MVC y además tiene patrones de diseño como Factory Method Facade y Observer y principios de diseño como S.O.L.I.D. DRY KISS y YAGNI. Este enfoque lo hemos elegido porque así conseguimos un código que es más limpio fácil de mantener y que se puede escalar bien además de que sigue las prácticas teóricas y ejemplos reales que aparecen en los documentos de referencia.

Comparación con Arquitecturas Industriales

En la industria se usan diferentes tipos de arquitecturas como las monolíticas en capas microservicios Clean Architecture o incluso arquitecturas que están basadas en DDD y Hexagonal Architecture.

Arquitecturas en Capas y MVC

Las soluciones industriales que usan MVC o arquitecturas en capas tienen el mismo objetivo de separar la lógica de negocio la presentación y el control lo que hace que sean más modulares y fáciles de mantener. Nuestra propuesta es parecida a esta idea porque aprovecha esa separación de responsabilidades y además permite hacer pruebas unitarias y extender el sistema más fácilmente.

Microservicios y Clean Architecture

Arquitecturas como los microservicios o Clean Architecture están pensadas para que los sistemas puedan crecer en horizontal y para que los módulos sean independientes. Aunque nuestro diseño modular es bastante bueno y escalable todavía no tiene una división en servicios independientes ni una separación total de los dominios que es algo más común en sistemas que tienen una carga de trabajo muy alta.

Fortalezas del Enfoque Propuesto

Modularidad y Separación de Responsabilidades

Como usamos S.O.L.I.D. y el patrón MVC cada parte del sistema tiene su función bien definida lo que hace que sea más fácil mantenerlo y mejorarlo con el tiempo.

Facilidad de Integración y Pruebas

Gracias a patrones como Observer y Facade el sistema responde mejor y es más fácil añadir módulos nuevos además de que se pueden hacer pruebas unitarias y comprobar que todo funciona bien de forma continua.

Flexibilidad para Extender Funcionalidades

El patrón Factory Method hace que se puedan añadir nuevos tipos de proyectos sin tocar la lógica que ya existe lo que hace que el sistema pueda crecer mejor.

Áreas de Mejora

Escalabilidad Horizontal

Aunque el diseño modular está bien en sistemas con mucha carga de trabajo podría ser mejor usar una arquitectura basada en microservicios o Clean Architecture para repartir mejor la carga y que funcione más rápido.

Aislamiento del Dominio

Si aplicamos principios de Domain-Driven Design DDD y usamos arquitecturas hexagonales podríamos hacer que la lógica de negocio esté más aislada lo que ayudaría a integrar mejor el sistema con otros y a reducir el acoplamiento.

Automatización y DevOps

En la industria es muy importante usar pipelines de CI/CD y herramientas de análisis estático como SonarQube. Si profundizamos en esto podríamos hacer que el sistema sea aún más robusto y mejorar su desarrollo de forma continua.

La arquitectura que hemos diseñado cumple con los estándares de modularidad mantenimiento y calidad que se requieren en la industria ya que se basa en principios y patrones bien fundamentados. Sin embargo si se quiere usar en sistemas con muchísima demanda y que necesitan una escalabilidad extrema lo mejor sería considerar microservicios o Clean Architecture además de trabajar más con DDD. Esta comparación muestra tanto los puntos fuertes de nuestro diseño como los aspectos en los que podríamos mejorarlo para que se parezca aún más a lo que se usa en la industria.