

1. Lectura previa

1.1 Leer los capítulos introductorios de Clean Architecture (Robert C. Martin).

1.2 Subrayar y anotar ideas clave relacionadas

¿Qué es arquitectura de software?

Es la estructura fundamental de un sistema, la forma en que están organizados sus componentes y cómo se comunican entre sí

No solo se refiere a diagramas o modelos sino a las decisiones estratégicas sobre diseño que impactan en la vida útil del software

Su propósito principal es maximizar la vida útil del sistema y facilitar su evolución

¿Por qué define el éxito o fracaso de un producto?

Una buena arquitectura permite que el software sea fácil de cambiar, mantener y escalar con el tiempo

Una mala arquitectura genera un sistema rígido que se vuelve costoso y riesgoso de modificar lo cual puede llevar al fracaso del producto

El código fuente cambia constantemente pero la arquitectura es lo que garantiza que esos cambios no rompan todo el sistema

O sea que:

Éxito = Arquitectura adaptable, que permite responder rápido al mercado y a nuevas necesidades

Fracaso = Arquitectura rígida que vuelve lento y caro el desarrollo

Principales atributos de calidad: escalabilidad, seguridad, disponibilidad, mantenibilidad, flexibilidad

Escalabilidad

El sistema debe poder crecer en usuarios datos o funcionalidades sin perder rendimiento

Seguridad

Proteger los datos y las interacciones del sistema contra accesos no autorizados o ataques

La arquitectura define cómo se autentican los usuarios cómo se encriptan datos y cómo se segmenta la información

Disponibilidad

El sistema debe estar accesible la mayor parte del tiempo.

La arquitectura influye en la redundancia tolerancia a fallos y planes de recuperación

Mantenibilidad

Capacidad de modificar y corregir el sistema de manera rápida y segura

Un diseño limpio y desacoplado permite arreglar errores y añadir funciones sin romper lo existente

Flexibilidad

Facilidad para adaptarse a cambios en los requisitos del negocio o en la tecnología

Una arquitectura flexible evita el “lock-in” con un proveedor o framework

2. Preparación del debate (individual):

2.1 Redactar un resumen de máximo 1 página:

La arquitectura de software puede entenderse como el conjunto de decisiones que definen la estructura de un sistema y la forma en que sus diferentes elementos se relacionan entre sí. Desde una perspectiva personal no se trata únicamente de diseñar cómo se verán las partes de un sistema o de elegir qué tecnologías se van a emplear sino de crear una base sólida que garantice que el producto pueda crecer, mantenerse y adaptarse a las necesidades del entorno con el paso del tiempo. Una buena arquitectura es como los cimientos de una construcción: pues aunque no se ven, son los que determinan la resistencia, la estabilidad y la vida útil de todo lo que se construya encima.

Cuando la arquitectura de un sistema está bien diseñada, el software se vuelve más sencillo de entender, de modificar y de extender; lo que a la larga se traduce en un producto más competitivo y con mayores posibilidades de éxito. En cambio, cuando las decisiones arquitectónicas se toman sin una visión a largo plazo, se generan sistemas frágiles, difíciles de mantener y costosos de escalar, lo que puede llevar incluso al fracaso de un proyecto o de una empresa que dependa de él. Por eso se puede decir que la arquitectura de software es uno de los factores que más influyen en el éxito o fracaso de un producto digital.

Un atributo esencial de la arquitectura de software es la escalabilidad. La escalabilidad permite que un sistema que empieza siendo pequeño pueda crecer junto con la cantidad de usuarios, de datos y de procesos sin perder velocidad ni calidad de servicio. Este atributo es muy importante para proyectos que esperan un crecimiento acelerado porque asegura que no sea necesario rehacer todo el sistema desde cero cuando se multiplique la demanda.

Otro atributo fundamental es la seguridad. La seguridad no se limita a poner contraseñas fuertes sino a diseñar el sistema de manera que proteja los datos sensibles, evite accesos indebidos y reduzca los riesgos frente a ataques externos. Cuando un sistema es inseguro, los usuarios pierden la confianza y la empresa corre grandes riesgos legales, económicos y de reputación. Por eso la seguridad se considera uno de los pilares de una arquitectura sólida.

Un tercer atributo clave es la mantenibilidad La mantenibilidad significa que el sistema se puede modificar corregir y mejorar sin que eso represente un riesgo de romper otras partes ni un gasto excesivo en tiempo o dinero Este atributo permite que los equipos de desarrollo respondan rápido a cambios en el mercado a nuevas necesidades de los usuarios o a la corrección de errores inevitables Con una buena mantenibilidad el software puede evolucionar y seguir siendo útil durante años mientras que sin ella se vuelve rígido y termina quedando obsoleto muy pronto

2.2 Pregunta para el debate:

Qué debería priorizar una empresa nueva: escalabilidad rápida o facilidad de mantenimiento?