

Desarrollo de Software

Artículo

Cinco pasos clave para diseñar una arquitectura de software

by Carlos Antonio Suarez Quintero y Pedro Guerrero Guerrero

4 min read

28 de septiembre de 2023



Comparte



La arquitectura de software desempeña un papel fundamental en el éxito de cualquier proyecto de desarrollo. Para garantizar que su software sea robusto, escalable y cumpla con los más altos estándares de calidad, es esencial seguir un enfoque meticuloso y estructurado. En este artículo, explicaremos en detalle los pasos clave para construir una arquitectura de software sólida, siguiendo las mejores prácticas y estándares definidos en la norma ISO 25010. Estos pasos cruciales incluyen:

Paso 1: Definición de Objetivos

Cuando creamos una arquitectura de software, cada decisión que tomamos debe estar basada en objetivos claros, de otra manera es muy difícil entender hacia donde apuntan nuestros esfuerzos y cuál es la forma más eficiente de alcanzar la meta que buscamos.

Por otro lado, identificar los requerimientos que afectan la estructura de una aplicación nos puede ayudar a reducir riesgos de negocio y establecer una relación clara entre los requerimientos técnicos y de negocio.

Estos son algunos de los objetivos más comunes :

- **Crear una arquitectura escalable:** construir sistemas capaces de crecer horizontalmente a través de servicios pequeños e independientes es clave para muchos tipos



de proyectos. La forma en la que definamos la arquitectura del sistema va a permitir que nuestras soluciones puedan crecer rápidamente, pero estas características no se pueden generalizar a todos y cada uno de los proyectos en los que trabajamos, por lo que identificar, desde el comienzo, si la escalabilidad va a ser uno de nuestros objetivos a la hora de plantearnos la arquitectura de un proyecto

- **Diseñar una arquitectura enfocada en la observabilidad:** este tipo de arquitecturas es útil cuando creamos soluciones que necesitan ser medidas constantemente y en las que es fundamental reaccionar rápidamente si se detectan anomalías. Definir con precisión todo lo relacionado con las métricas que se van a emplear en este tipo de proyectos es una parte fundamental para su éxito.
- **Construir una arquitectura centrada en la mejora continua:** si el negocio para el cual estamos construyendo una solución requiere que nuestro sistema evolucione y se adapte permanentemente, es importante enfocar nuestros esfuerzos en permitirle al equipo de desarrollo la posibilidad de incorporar cambios de manera ágil y sin mayores obstáculos.
- **Arquitectura estable :** existen negocios para los cuales contar con un sistema resiliente y capaz de resistir a cualquier tipo de factor externo es fundamental. Por eso, las herramientas que nos ayudan a simular fallos tienen



un papel importante para este tipo de arquitecturas. Uno de los ejemplos más conocidos es Chaos Monkey una herramienta open source creada por el equipo de desarrollo de Netflix y que está fundamentada en la premisa según la cual para evitar grandes fallos, es necesario fallar constantemente. Lo que hace Chaos Monkey programar una simulación de fallos que permiten poner a prueba la resiliencia de sus servicios de AWS.

También es importante tener en cuenta que nuestra arquitectura debe ser fácil de entender para que, en el largo plazo, otros equipos no tengan dificultades para trabajar con ella y construir sobre lo construido. Del mismo modo, y sobre todo en los proyectos donde la estabilidad es una prioridad, construir la arquitectura pensando en la posibilidad de hacer pruebas individuales, de aceptación y/o funcionales va a tener una particular importancia.

Paso 2: Definir los Drivers de Arquitectura

Una vez que los objetivos están claramente definidos, es posible determinar los requerimientos que van a guiar el diseño de la arquitectura del sistema, también conocidos como drivers.

Los drivers nos permiten cumplir con los estándares y en la mayoría de los proyectos actuales estos drivers están determinados por el modelo de calidad ISO 25010.



Un modelo de calidad determina las características que se van a tener en cuenta cuando evaluamos las propiedades de un producto. En el caso del modelo ISO 25010, encontramos que se hace una valoración del software a partir de atributos de calidad como la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, la mantenibilidad.

Para conocer más sobre los atributos de calidad del modelo ISO 25010 puedes [seguir este link](#).

Paso 3: Definición de contextos y toma de decisiones de diseño

Una arquitectura de software exitosa no puede desarrollarse en un vacío; debe adaptarse a su entorno y contexto. Definir los contextos implica comprender las limitaciones tecnológicas, las restricciones presupuestarias y las necesidades específicas de los usuarios.

Una vez contamos con un conocimiento sólido del contexto, es hora de tomar decisiones. ¿Cuál es el patrón de diseño, las tecnologías y los componentes más eficientes para alcanzar los objetivos que nos hemos planteado?

En este punto, es crucial encontrar un equilibrio entre las necesidades funcionales y las no funcionales, así como considerar aspectos como la modularidad, la seguridad y la escalabilidad. Cada decisión de arquitectura debe estar



alineada con los objetivos y drivers definidos previamente para asegurar que la arquitectura sea coherente y efectiva

Paso 4: Definir las tácticas de Arquitectura

Las tácticas son decisiones de diseño que afectan el control de atributos de calidad específicos. Las tácticas se enfocan en atributos de calidad específicos en lugar de ofrecer soluciones detalladas, como ocurre con los patrones de diseño.

Las tácticas se presentan de manera jerárquica, con categorías de atributos de calidad en la raíz, preocupaciones específicas debajo de cada categoría y tácticas específicas debajo de las preocupaciones. Por ejemplo, para mejorar el desempeño, se pueden aplicar tácticas relacionadas con la administración de recursos, como la introducción de concurrencia o el mantenimiento de copias múltiples.

Paso 5: Validaciones ATAM (TradeOff)

Las validaciones ATAM, o Método de Análisis de Intercambio de Arquitectura de Software por sus siglas en inglés, son una poderosa herramienta para los arquitectos de software. Este método se enfoca en analizar cómo una arquitectura satisface objetivos de calidad, pero logra ir mucho más allá.

Por un lado, su capacidad para analizar sistemas heredados. Esto es especialmente valioso cuando un sistema heredado debe adaptarse a modificaciones importantes, integrarse



con otros sistemas o someterse a actualizaciones significativas. Por otro lado, gracias a las validaciones ATAM, podemos entender mejor cómo interactúan entre sí los atributos de calidad de un sistema y cómo se compensan mutuamente.

El ATAM nos permite evaluar las consecuencias de las decisiones arquitectónicas en función de los requisitos de atributos de calidad. Su objetivo es identificar riesgos potenciales en la arquitectura de un sistema de software complejo y hacerlo de manera temprana, económica y detallada.

En resumen, la construcción de una arquitectura de software exitosa es un proceso meticuloso y estratégico que requiere una planificación cuidadosa desde la definición de objetivos hasta las validaciones ATAM. Siguiendo estos pasos de manera minuciosa, estará estableciendo las bases para el éxito a largo plazo de su software en un entorno digital en constante evolución. La inversión en la construcción de una arquitectura sólida es esencial para asegurar la calidad, la confiabilidad y el rendimiento de su software en el mercado actual.

