



Actividad:

Taller Arquitectura de software

GA4-220501095-AA2-EV06

Aprendiz:

Wilmer Jair Espinosa Silva

CC: 1.095.910.391

Instructor:

ISRAEL ARBONA GUERRERO

Servicio Nacional de aprendizaje-SENA

Curso: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha: 2455285

Elaborar un taller sobre arquitectura de software teniendo en cuenta las siguientes preguntas:

Elementos a tener cuenta en la actividad

¿Qué entiende por arquitectura de software?

RTA: La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

El concepto de arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo. Al igual que en la ingeniería civil, las decisiones críticas relativas al diseño general de un sistema de software complejo deben de hacerse desde un principio. El no crear este diseño desde etapas tempranas del desarrollo puede limitar severamente el que el producto final satisfaga las necesidades de los clientes. Además, el costo de las correcciones relacionadas con problemas en la arquitectura es muy elevado. Es así que la arquitectura de software juega un papel fundamental dentro del desarrollo.

¿Cuál es su función?

RTA: Identificar soluciones de software que beneficien al usuario final. Encargarse de los proyectos de desarrollo de software y comunicar los avances a la dirección de la compañía.

Al hacer que este proceso sea más eficaz, puedes justificar todos tus requisitos de manera adecuada y brindar a las partes interesadas la oportunidad de presentar sus aportaciones

¿Cómo se elabora la arquitectura de software?

RTA:

1. **Comprende claramente cuáles son tus requisitos:** Todo diseño que comiences tendrá requisitos funcionales y no funcionales. Estos requisitos guían tu arquitectura de software y te permiten concluir el proyecto con la presentación de un producto final que deje satisfechas a las partes interesadas.

2. **Comienza a pensar en cada componente:** Es necesario que averigües qué requisitos plantearán desafíos importantes para tu diseño o plan de proyecto, sin adelantarte demasiado al pensar en la implementación. Algunos requisitos pueden ser imposibles de cumplir bajo ciertos supuestos u opciones.
3. **Divide tu arquitectura en “rebanadas”:** Para esto, es útil pensar en cómo se rebanan un pastel. La mayoría de la gente lo hace pasando el cuchillo por cada una de las capas. Digamos que una rebanada completa tiene glaseado, relleno y dos o tres capas de bizcocho en su interior. Esta rebanada vertical tiene un poco de todos los ingredientes. Un pastel también puede rebanarse horizontalmente, separando cada una de las capas.
 - De manera similar, si cada rebanada es un piso que se utiliza para planear la arquitectura de tu software, entonces hay capas. Si rebanas el pastel verticalmente, creas pisos enfocados en características individuales, mientras que, si lo rebanas horizontalmente, se muestran los componentes individuales. Necesitas tener una mentalidad tanto horizontal como vertical para tu proyecto.
4. **Hacer un prototipo:** Siempre crea un prototipo. Los prototipos te permiten descubrir fallas de forma rápida y temprana, por lo que obtendrás retroalimentación con rapidez y podrás descubrir tu prueba de concepto. Esta es una parte importante de la validación de tu trabajo y de la comprobación de tus supuestos para asegurarte de que sean válidos y minuciosos.
 - **Mantén un historial de cambios riguroso:** por supuesto, si no documentas lo que vas descubriendo en la creación de los prototipos, corres el riesgo de repetir tus errores. Anota todo: documenta minuciosamente tus decisiones de diseño y los cambios que realices sobre la marcha.
 - **Ten una sola fuente de información:** no será conveniente que tengas múltiples cambios y diferentes versiones que frenen tu progreso, así que establece un sólido control de versiones basadas en una única fuente de información para tu documentación.
 - **Haz diagramas de tus prototipos:** puedes usar diagramas para ayudarte a administrar los cambios de prototipos y visualizar las diferencias entre cada versión.
5. **Identifica y cuantifica los requisitos no funcionales:** Además de los requisitos funcionales, deberás tener en cuenta los requisitos no funcionales. Estos requisitos son tan importantes como tus requisitos funcionales para el diseño porque definen las características del sistema.

- Los requisitos no funcionales normalmente son los requisitos de calidad general para todo el proyecto, pero no siempre. Es posible que tu sistema tenga requisitos no funcionales específicos solo para una parte de tu arquitectura de software. Por lo tanto, hay que estar preparados para incluir a las partes interesadas en los requisitos locales no funcionales. Por ejemplo, una rebanada vertical particular del proyecto puede interesar al equipo de atención al cliente en especial, y ese equipo puede tener sus propias expectativas sobre esos requisitos no funcionales.

Desempeño: qué tan bien funciona todo tu sistema, así como las rebanadas o capas individuales.

Escalabilidad: el potencial actual y futuro para escalar tu sistema junto con tus necesidades.

Portabilidad: la portabilidad de tus datos, así como la posible portabilidad de los componentes de tu sistema si procede o es necesario.

Extensibilidad: explica el crecimiento futuro de tu sistema y de tu empresa, la capacidad de adaptación de tu sistema y el esfuerzo que conlleva la adaptación.

Cumplimiento normativo: es otro factor esencial y que tiene un impacto importante en el diseño general de tu proyecto.

6. **Visualiza tu diseño:** El uso de elementos visuales en todo tu trabajo conceptual de diseño, así como en tu implementación, permite a tu equipo ver la perspectiva general en la que se basa tu diseño. Los diagramas son una excelente forma de visualizar procesos y diferentes aspectos de tus opciones de diseño.
7. **No elijas patrones:** Mantén tu proceso de diseño enfocado en el panorama general sin recurrir a los patrones como punto de partida. En lugar de usar patrones, mantén un nivel muy general con una visión de los componentes principales para reducir el riesgo de tener una sobreingeniería de tus sistemas.

¿Cómo lograr una buena arquitectura?

RTA:

Establece sistemas robustos, pero libres de frameworks. Esto permite que la estructura sea estable e independiente.

Ajusta la construcción y el uso de la base de datos. La base de datos es la que debe alinearse a tus necesidades de negocio, no al revés.

Elige las herramientas necesarias para optimizar procesos. Considera la posibilidad de unificar servicios de automatización de sistemas e integración de aplicaciones, colocándolos de manera estratégica desde un inicio.

Auditable y testable. Es importante crear un ambiente donde el código pueda ser evaluado, medido y regulado

¿Cuáles son los elementos de diseño de una arquitectura de software?

RTA: Entre los componentes de la arquitectura de software existe un conjunto de interacciones entre las que sobresalen :

- llamadas a procedimientos.
- comportamiento de variables.
- protocolos cliente servidor.
- transmisión asíncrona de eventos.