

ARQUITECTURA IDEAL SEGUN EL TIPO DE APP

K.J Guzman Ramirez

7690-21-2903 Universidad Mariano Galvez

Seminario de Tecnologias de Informacion

kguzmanr2@miumg.edu.gt

URL REPOSITORIO LA TEX:

<https://github.com/KennetJRamirez/Foro-Seleccion-Arquitectura>

Resumen

En el desarrollo , el elegir una arquitectura correcta es vital para definir el exito de un proyecto. Esta define la estructura, componentes y como sera la interaccion del sistema, afectando el rendimiento, mantenimiento y escalabilidad a largo plazo. Para esto es que se realizan preguntas clave sobre el software, usuarios finales, expectativas, recursos y como seria posible una integracion a sistemas ya existentes. En la actualidad existen variedad de opciones, monolitica, microservicios, cliente-servidor, capas, reactivas, nube, etc, cada una con sus ventajas como desventajas y retos. No hay una solucion definitiva, esta dependera del contexto, capacidades, recursos, presupuesto etc. Es importante siempre recordar que aunque la arquitectura pueda parecer una decision tecnica, sus efectos se mostraran en todo el ciclo de vida de un proyecto, por lo que importante que se defina desde las etapas iniciales para evitar tener problemas futuros, que impliquen en gastos innecesarios o retrasos que impliquen perdidas para la empresa. .

Palabras clave:

Desarrollo del tema

La actualidad en al cual nos encontramos, hace que cada avance que de la tecnologia nos dificulte aun mas la eleccion cuando se desea crear algun tipo de software. Y se debe de hacer enfasis por que es vital ya que de aca se determinara si tendra exito o no. Debido a que aca se define la estructura, arquitectura, organizacion, componentes, interacciones, etc, lo cual viene a tener un impacto en el rendimiento y posible escalabilidad que tendra la aplicacion a futuro. Por lo que antes de empezar a tomar esta decision importante debemos de delimitar bien , usando algunas preguntas como:

- ¿Cuales seran las funciones de mi software?
- ¿Quienes son mis usuarios y las expectativas a cumplir?
- ¿Requisitos de escalabilidad, rendimiento y seguridad?
- ¿Limitaciones de recursos?
- ¿Se integrara a un sistema existente?

Opciones en la actualidad

Hoy en dia por suerte los desarrolladores tienen una mejor variedad de opciones que se pueden ser aptas para la creacion de software en base a las necesidades que se desean solventar. Cada una ofrece ventajas como posibles desventajas, por lo que es importantes conocer que nos brinda cada una de ellas, ya que no hay ninguna opcion que sea la unica buena de todas, ya que dependera de la situacion, contexto, recursos, etc. Por lo que entre las mas usadas a dia de hoy nos encontramos con:

- **Arquitectura Monolitica:** Aca todo se encuentra en un mismo sitio. Facilidad y rapidez de desarrollo son las palabras que mejor la definen en proyectos pequeños. Por lo que no es recomendada si a futuro se desea escalar y mas aun si se necesita mas complejidad.
- **Arquitectura de Microservicios:** Aca todo lo que integra al sistema esta separado y de comunicacion independiente. Haciendo posible que si ocurre un error no implique que todo el sistema estara sin poder utilizar, ya que solo el area afectada sera la que no se encuentre disponible. Pero es dificil de controlar si se tiene un sistema demasiado complejo.
- **Arquitectura Cliente-Servidor:** Aca como su nombre lo indica , contamos con un cliente que hace peticiones a un recurso el cual el servidor debera responder para devolverle dicho recurso al cliente. Aca una de las ventajas es que el cliente no debe tener instalado nada y que todo ocurre del lado del servidor, como lo puede ser una actualizacion, la cual luego afectara a todos los clientes, puede ser algo bueno o malo, depende como se maneja.
- **Arquitectura de Capas:** Aca se separa en 4 capas la representacion del sistema, siendo: modelo, vista, controlador, logica de negocio. Es muy util debido a que se tiene todo correctamente gestionado y los cambios son mas faciles de implementar, pero se vuelve demasiado complejo cuando el sistema crece a tal punto que empieza a ser compleja la comunicacion entre el mismo.
- **Arquitectura Basada en Eventos:** Aca todo se maneja en base a los eventos que ocurran, cada componente va a reaccionar en base a lo que ocurra, sin tener que hacer llamadas directas, lo que permite un alto nivel de respuesta y capacidad de usuarios sin afectar el rendimiento, ya que asigna recursos en base a la cantidad de trafico que exista. Es usada mayormente en la actualidad, debido a la alta respuesta que ofrece a los usuarios que cada vez mas exigen todo en segundos. Pero es dificil de implementar, ademas de que se debe tener una buena cantidad de recursos debido a que es algo robusta y demandante.
- **Arquitectura sin Servidor:** Aca simplemente con tener ya el sistema hecho, se busca un proveedor que brinde acceso a una plataforma, en la cual ya no hay que preocuparse de instalar, configurar o mantener el servidor. Ya que el proveedor toma estas tareas, pero asi como suena muy innovador y que facilita la vida, tiene un enorme problema, siendo este su alto costo si se tiene un uso muy alto, que pasa casi siempre desapercibido hasta que llega el momento del cobro, ademas de estar siempre requiriendo de la participacion del proveedor, en casos de caida, actualizaciones, seguridad, etc.

Otras consideraciones

No basta solamente con conocer acerca de las opciones que hay disponibles, tambien es necesario en pensar en el comportamiento que tendra a largo plazo. Entre estos podemos hacer menciones a:

- Los equipos de desarrollo y su capacidad
- Facilidad de darle mantenimiento y escalabilidad
- Presupuesto y costos operativos

Importancia de una buena eleccion de arquitectura

Elegir un tipo o tipos de arquitectura es clave tal como se menciono al comienzo, ya que influye en el orden , exito, eficiencia y un desarrollo acorde a los objetivos de la empresa, siendo asi que si ya se elegio la arquitectura, los beneficios que tendremos despues de este paso , pueden ser:

- Guia que evita improvisar
- Adoptar tecnologías y herramientas estandarizadas:
- Obtener un sistema que cumpla con las necesidades
- Cambios y escalabilidad sin romper nada
- Mejor comunicacion entre clientes y equipo de desarrollo
- Alinear necesidad contra lo que la tecnologia ofrece

Observaciones y comentarios

- No existe una arquitectura que sea la mejor , cada una de ellas tiene sus ventajas, desventajas, retos de implementar, etc. Todo depende de que se quiere lograr
- Es normal que no se considere tan importante el elegir correctamente un tipo de arquitectura, pero a futuro obtiene o presentan muchos problemas muy complejos en cuanto vaya creciendo tanto la complejidad del sistema como el trafico del mismo.

Conclusiones

- Se llego a la conclusion de que es vital desde primeras etapas definir la cual se va a usar, ya que asi se evitan problemas a futuro como lo pueden ser el rendimiento, mantenimiento y escalabilidad.
- Se llego a la conclusion de que es importante conocer las opciones existentes para poder llegar a desarrollar, pero lo es mas poder adecuarlas en base a lo que en realidad necesitamos cubrir, ya que asi se garantiza que la posible solucion sea efectiva, util y apegada a los objetivos.

Referencias

- [1] Yokwejuste. (2023). *Choosing the best architecture for your software*. Dev.to. Recuperado de https://dev-to.translate.googleusercontent.com/translate/en/choosing-the-best-architecture-for-your-software-23h9?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- [2] Sandaruwandha, R. (2023). *Choosing the right software architecture: A guide to selecting the best fit for your project*. Medium. Recuperado de https://ravindusandaruwandh-medium-com.translate.googleusercontent.com/translate/en/choosing-the-right-software-architecture-a-guide-to-selecting-the-best-fit-for-your-project-eedda7812fb1?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- [3] Cellenza. (2023). *How to choose the best software architecture with architectural drivers*. Blog Cellenza. Recuperado de https://blog-cellenza-com.translate.googleusercontent.com/en/methodology/software-craft/how-to-choose-the-best-software-architecture-with-architectural-drivers/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- [4] AppMaster. (2023). *Cómo elegir la arquitectura de software*. Recuperado de <https://appmaster.io/es/blog/como-elegir-la-arquitectura-de-software>