



¿Por qué deberíamos usar esa arquitectura?

La arquitectura planteada en el diagrama con micro-servicios, se debería usar ya que plantea el uso de herramientas y algoritmos que priman la optimización en el rendimiento del micro-servicio ya que no requiere de gran robustez y cantidad de herramientas que pueden brindar otras tecnologías, siendo la demora en el procesamiento y respuesta del servicio la principal problemática y el objetivo principal del servicio.

¿Por qué deberíamos usar esas tecnologías?

Flask- Python es conocido por ser un framework sencillo que permite versatilidad y rapidez en desarrollo, mantenimiento de acuerdo al requerimiento del micro-servicio el performance que brinda Flask es ideal para la solución requerida.

¿Cuáles son las ventajas de su arquitectura propuesta sobre el resto de arquitecturas existentes? Compare por lo menos 3 arquitecturas.

La arquitectura basada en micro-servicios planteada en la solución de esta situación es ideal en comparación con otras arquitecturas como lo puede ser la arquitectura en capas y servicios monolitos ya que permite la rápida gestión, administración y mantenimiento del servicio, enfocando los recursos del entorno al rendimiento del servicio sin afectar los demás servicios del sistema como lo pueden llegar a hacer las demás posturas expuestas ya que al estar enfocada en la solución de un proceso directamente, su mejora, monitoreo independiente permite utilizar las herramientas ideales.

¿Cuáles son las ventajas de las tecnologías propuestas para desarrollar dicha arquitectura? compara contra otras tecnologías dentro del mercado.

En el caso del micro-Framework Flask la versatilidad, sencillez y el performance en los procesos le pone en ventaja a otras tecnologías como Django o FastApi. En el caso de Django a pesar de que es un framework muy potente y con la comunidad activa más grande de los 3, dada su robustez le hace más lento y complejo que Flask y no optimo para el micro-servicio requerido, por el lado de FastApi a pesar que es el FrameWork más rápido entre los 3 es relativamente nuevo y no se tienen muchos casos de referencia, siendo Flask la opción ideal brindando velocidad, escalabilidad y estabilidad