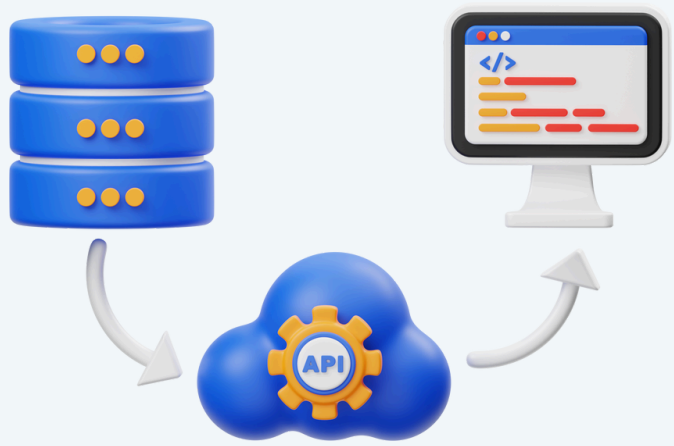
Cómo Diseñar APIs de Alto Rendimiento y Baja Latencia

Aprende las claves para que tus APIs sean rápidas, escalables y confiables.



PABLO DEL ÁLAMO

Introducción

Imagínate una autopista sin tráfico, así deberían ser las APIs de alto rendimiento.

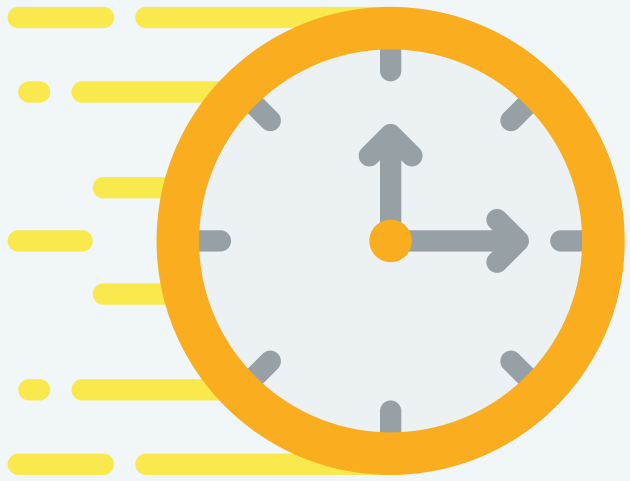
Hablemos de cómo lograr esa velocidad mientras mantenemos una baja latencia. 🚀

PABLO DEL ÁLAMO

¿Qué es el Rendimiento en una API?

El rendimiento aquí es cómo de rápido responde tu API a las peticiones.

Básicamente, si fuera una carrera de coches, querrías siempre llegar primero.



PABLO DEL ÁLAMO

¿Por qué es importante?

Cuando una API va lenta, la experiencia del usuario se ve afectada y los clientes se frustran.

Además, un rendimiento pobre puede costar dinero y prestigio de tu producto o servicio.

PABLO DEL ÁLAMO

Factores que Afectan el Rendimiento

Son muchos: diseño de la base de datos, tamaño de las respuestas, servidores saturados... y hasta el clima, si quieres ponerte metafórico jeje.

PABLO DEL ÁLAMO

Uso de Protocolos Eficientes

HTTP/2 y gRPC son como los superdeportivos del mundo de los protocolos.

Reducen el overhead y son más rápidos que su hermana pequeña, HTTP/1.

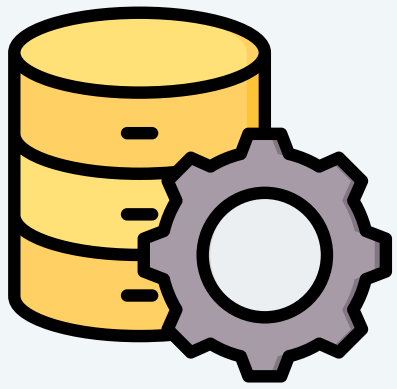
PABLO DEL ÁLAMO

Cachear, Cachear, Cachear

Aprovecha al máximo tu caché.

Mejora la velocidad y reduce la carga en tus servidores. ¡Redis o Memcached serán tus mejores aliados!

Tengo un post hablando en detalle hablando sobre el cacheo de datos, échale un ojo 😉



PABLO DEL ÁLAMO

Paginación en las Respuestas

No devuelvas una enciclopedia si te piden la hora.

Pagina las respuestas para manejar grandes volúmenes de datos.

PABLO DEL ÁLAMO

Optimización de Bases de Datos

Asegúrate de que tus consultas a la base de datos sean rápidas y eficientes.

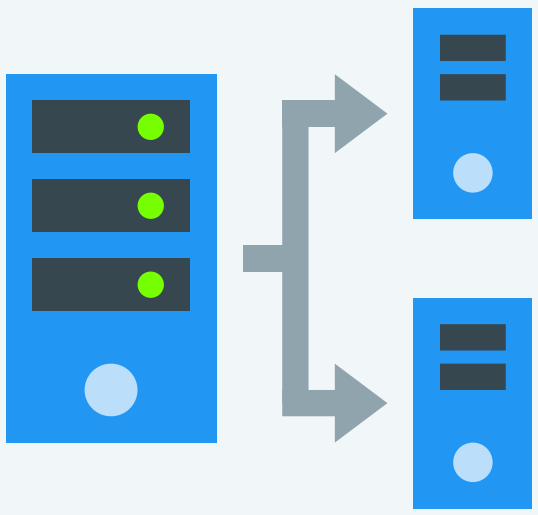
Usa índices siempre que sea necesario.

PABLO DEL ÁLAMO

Balanceadores de Carga

Distribuyen el tráfico entre múltiples servidores, ajustando así la carga según lo requieren las necesidades del momento.

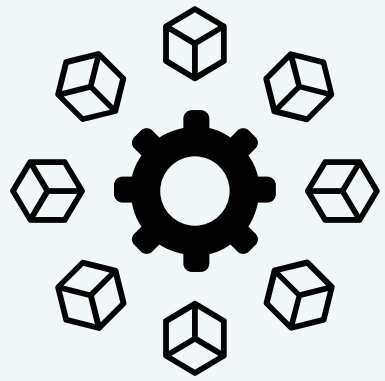
Si usas AWS por ejemplo, son muy fáciles de implementar.



PABLO DEL ÁLAMO

Desacoplamiento de Servicios

Depende del caso de uso, ya que siempre defenderé que no son la solución para todo, pero en caso de que tenga sentido para tu caso de uso, implementar microservicios permite que los componentes de la aplicación funcionen de manera independiente, incrementando la escalabilidad y el rendimiento.



PABLO DEL ÁLAMO

Priorizar el Uso de CDN

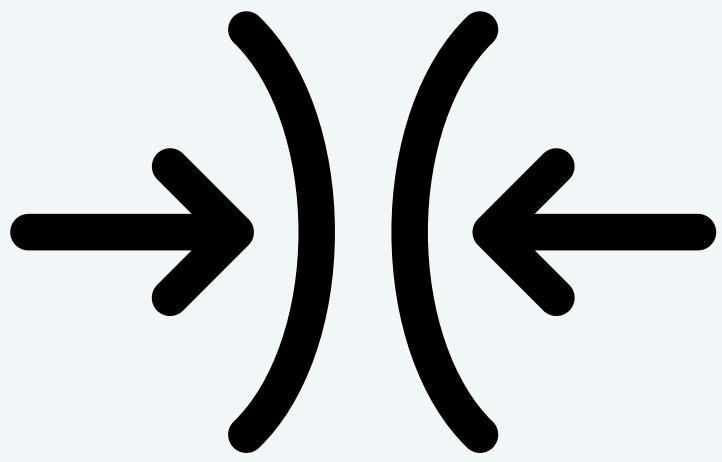
Un Content Delivery Network (CDN) mejora la velocidad al distribuir el contenido en varios servidores ubicados cerca del usuario final.

Esto reduce significativamente el tiempo de carga.

PABLO DEL ÁLAMO

Compresión de Datos

Utilizar técnicas de compresión como gzip o Brotli disminuye el tamaño de las respuestas, ofreciendo tiempos de transferencia más rápidos.



PABLO DEL ÁLAMO

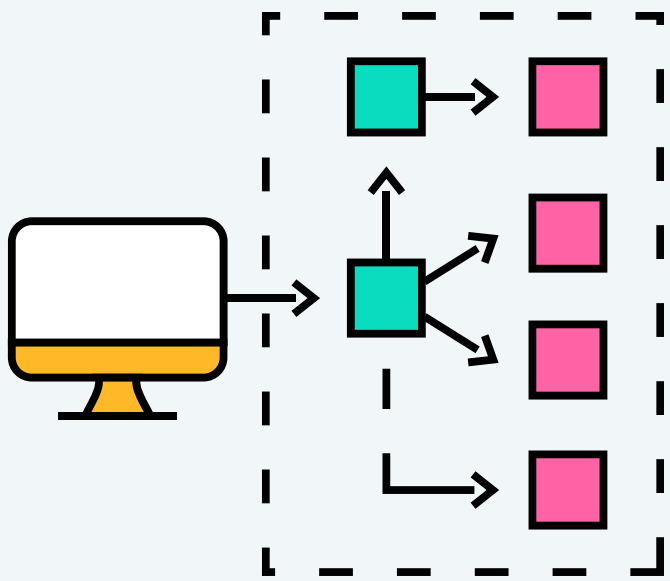
TLS Termination

Delegar las tareas de encriptación al CDN o a un balanceador de carga puede liberar los servidores de esta carga extra, mejorando así la velocidad de procesamiento

PABLO DEL ÁLAMO

Diagramas de Arquitectura

Utilizar diagramas para visualizar los componentes y el flujo de datos ayuda a entender mejor cómo interactúan y dónde se pueden aplicar mejoras.



PABLO DEL ÁLAMO

Pruebas de Estrés

Herramientas como Apache JMeter pueden simular situaciones de alta carga para evaluar el rendimiento de la API bajo condiciones extremas, identificando posibles cuellos de botella.

PABLO DEL ÁLAMO

Monitorización y Registro

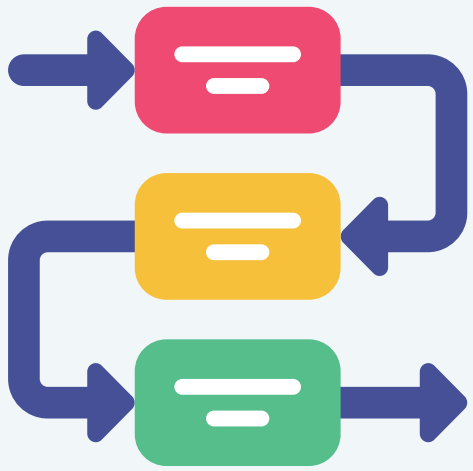
El uso de herramientas de monitorización como AWS CloudWatch permite detectar problemas proactivamente y optimizar el rendimiento basándose en datos concretos.

PABLO DEL ÁLAMO

Actualización de Dependencias

Mantener las librerías y dependencias actualizadas es esencial para beneficiarse de mejoras en rendimiento y seguridad.

Evitemos que se conviertan en obsoletas.



PABLO DEL ÁLAMO

Conclusión

Diseñar APIs de alto rendimiento y baja latencia es esencial para ofrecer una experiencia de usuario óptima.

Considera factores como cachés eficientes, paginación adecuada, protocolos económicos en recursos, balanceadores de carga robustos, y el uso de técnicas como compresión de datos.

La vigilancia constante a través de herramientas de monitorización asegura que tu API se mantenga ágil y segura. La eficiencia y la escalabilidad son clave; implementa estas prácticas y lleva tu API al siguiente nivel.

PABLO DEL ÁLAMO

¿Te ha resultado útil?

- Comparte esta guía con tu equipo o amigos desarrolladores.
- Guárdala para tenerla siempre a mano.
- ¡Dale un like o comenta si tienes preguntas!

PABLO DEL ÁLAMO