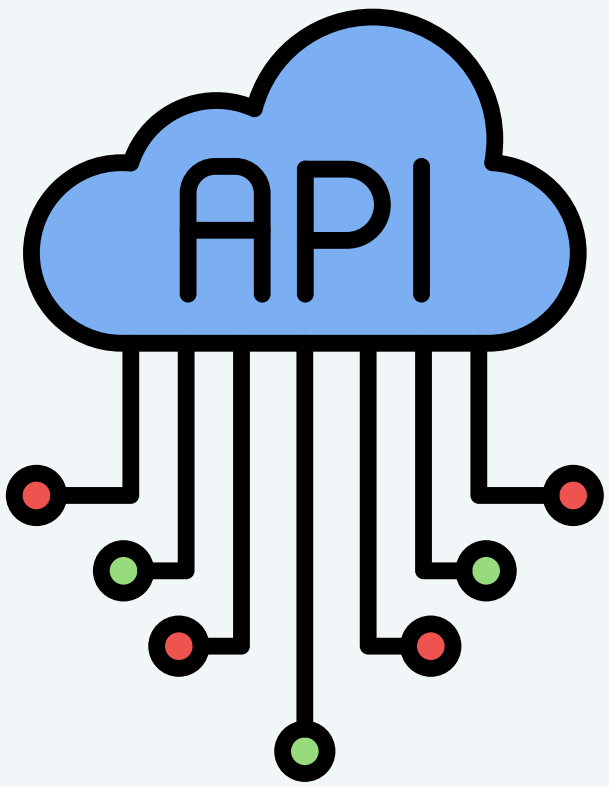
Rate Limiting para APIs

Cómo proteger tu API sin afectar a los usuarios legítimos



PABLO DEL ÁLAMO

Introducción

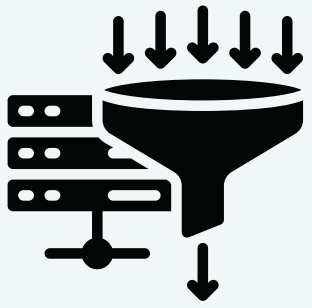
El rate limiting es una técnica que limita la cantidad de solicitudes que los usuarios pueden hacer a una API en un periodo de tiempo.

Su objetivo es proteger tu sistema de abusos, ataques o sobrecargas, garantizando al mismo tiempo una buena experiencia para los usuarios legítimos.

PABLO DEL ÁLAMO

¿Por qué necesitas rate limiting?

- Prevención de abusos: Evita ataques DDoS o scraping excesivo.
- Protección de recursos: Asegura que tu infraestructura no se sature.
- Equidad: Garantiza que todos los usuarios tengan acceso justo a los recursos.



PABLO DEL ÁLAMO

Funcionamiento básico del rate limiting

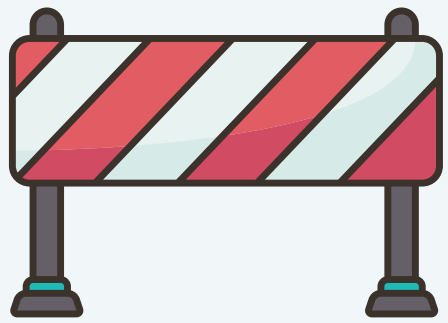
El Rate limiting funciona asignando un límite de solicitudes por usuario, clave API o IP en un intervalo de tiempo específico.

Las solicitudes que exceden el límite son rechazadas con un código de estado HTTP como 429 Too Many Requests.

PABLO DEL ÁLAMO

Métodos comunes de rate limiting

- Fixed Window: Límite fijo por intervalo de tiempo (ej., 100 solicitudes por minuto).
- Sliding Window: Ajusta límites dinámicamente según la actividad reciente.
- Token Bucket: Los usuarios consumen "tokens" por cada solicitud.

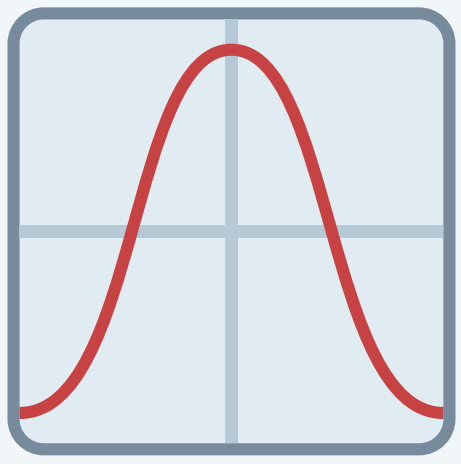


PABLO DEL ÁLAMO

Fixed Window

- Ventaja: Fácil de implementar.
- Desventaja: Picos al inicio de cada ventana.

Ejemplo: Un usuario puede hacer 100 solicitudes al final de un minuto y otras 100 al inicio del siguiente.

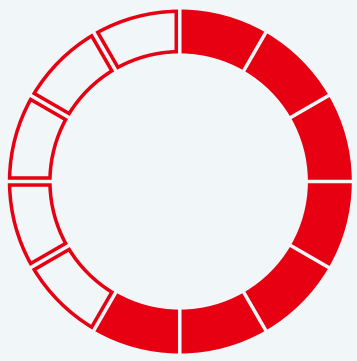


PABLO DEL ÁLAMO

Sliding Window

- Ventaja: Distribución más uniforme de solicitudes.
- Desventaja: Más compleja de implementar.

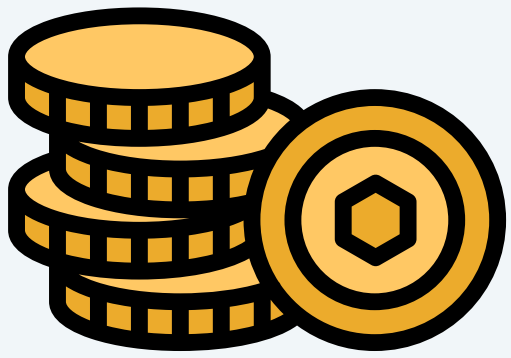
Ejemplo: Una API permite 100 solicitudes en los últimos 60 segundos en cualquier momento.



PABLO DEL ÁLAMO

Token Bucket

- Ventaja: Permite ráfagas controladas.
- Desventaja: Requiere mayor control sobre el almacenamiento de tokens.
- Ejemplo: Cada usuario tiene 100 tokens por minuto; cada solicitud consume uno.



PABLO DEL ÁLAMO

Implementación de rate limiting con herramientas populares

- NGINX: Configuración directa en el servidor.
- AWS API Gateway: Límite por usuario o clave API.
- Express.js (Node.js): Middleware como express-rate-limit.

PABLO DEL ÁLAMO

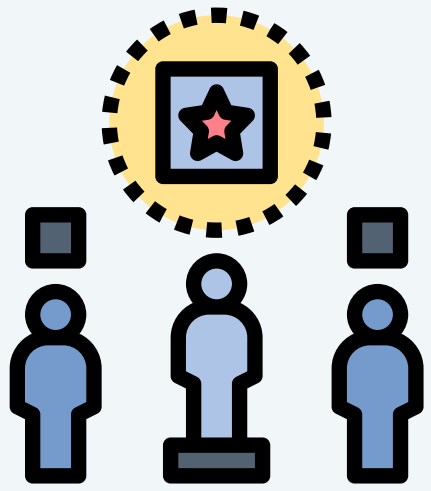
Evitando impacto en usuarios legítimos

- Incrementa límites para usuarios autenticados.
- Permite ráfagas controladas con sistemas como Token Bucket.
- Prioriza solicitudes críticas en tu sistema.

PABLO DEL ÁLAMO

Diferenciación de usuarios

- Usuarios autenticados: Límites más altos.
- Usuarios anónimos: Límites más bajos para evitar abuso.
- Claves API personalizadas: Límites ajustados según el cliente.



PABLO DEL ÁLAMO

Estrategias avanzadas para mejorar la experiencia

- Expón los límites en las respuestas: Incluye cabeceras como X-RateLimit-Limit y X-RateLimit-Remaining.

- Ofrece mecanismos de auto-retry: Los usuarios legítimos pueden manejar los límites fácilmente.

PABLO DEL ÁLAMO

X-RateLimit-Limit: 100
X-RateLimit-Remaining: 10
X-RateLimit-Reset: 30

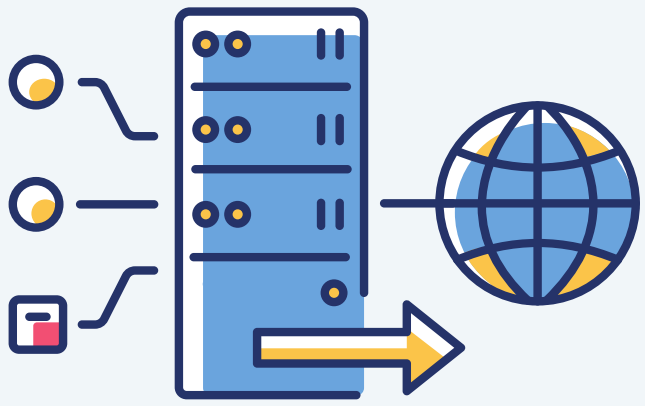
Esto indica que el límite es 100, que quedan 10 solicitudes y que el límite se restablecerá en 30 segundos.

PABLO DEL ÁLAMO

Rate limiting basado en IP

Útil para APIs públicas.

Reto: Usuarios detrás de proxies o NAT pueden compartir la misma IP, causando bloqueos no deseados.

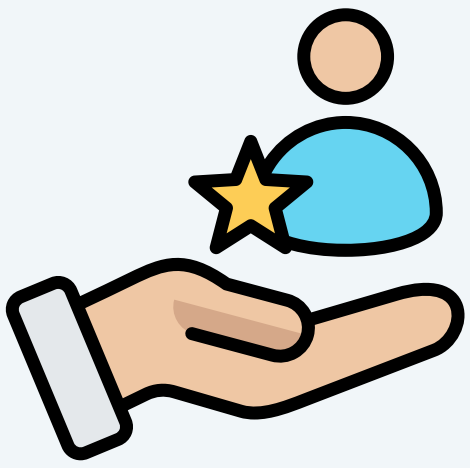


PABLO DEL ÁLAMO

Rate limiting basado en usuario o clave API

Ventaja: Personalización según cliente.

Ejemplo: Un cliente premium tiene un límite de 1000 solicitudes por minuto, mientras que uno estándar solo 100.



PABLO DEL ÁLAMO

Gestión de excepciones

- Lista blanca: Permite a IPs o clientes de confianza evitar límites.
- Límites diferenciados: Servicios internos pueden tener reglas menos estrictas.

PABLO DEL ÁLAMO

Monitoreo y alertas

Usa herramientas como:

- Prometheus para métricas.
- Grafana para alertas de tráfico inusual.
- Elastic Stack para analizar logs.

PABLO DEL ÁLAMO

Prevención de abuso y ataques

- Rate limiting combinado con captchas: Para bloquear bots sin afectar a humanos.

- Bloqueo dinámico: Incrementa límites si detectas patrones sospechosos.

PABLO DEL ÁLAMO

Conclusión

El rate limiting es una herramienta esencial para proteger tus APIs de abusos y garantizar un uso equitativo de los recursos.

Al implementar estrategias como Fixed Window, Sliding Window o Token Bucket, puedes equilibrar seguridad y experiencia de usuario.

Adapta los límites según el tipo de cliente, monitorea constantemente y ajusta según las necesidades.

Proteger tu API no significa complicar la vida de los usuarios legítimos, ¡asegúrate de que ambos objetivos se cumplan!

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