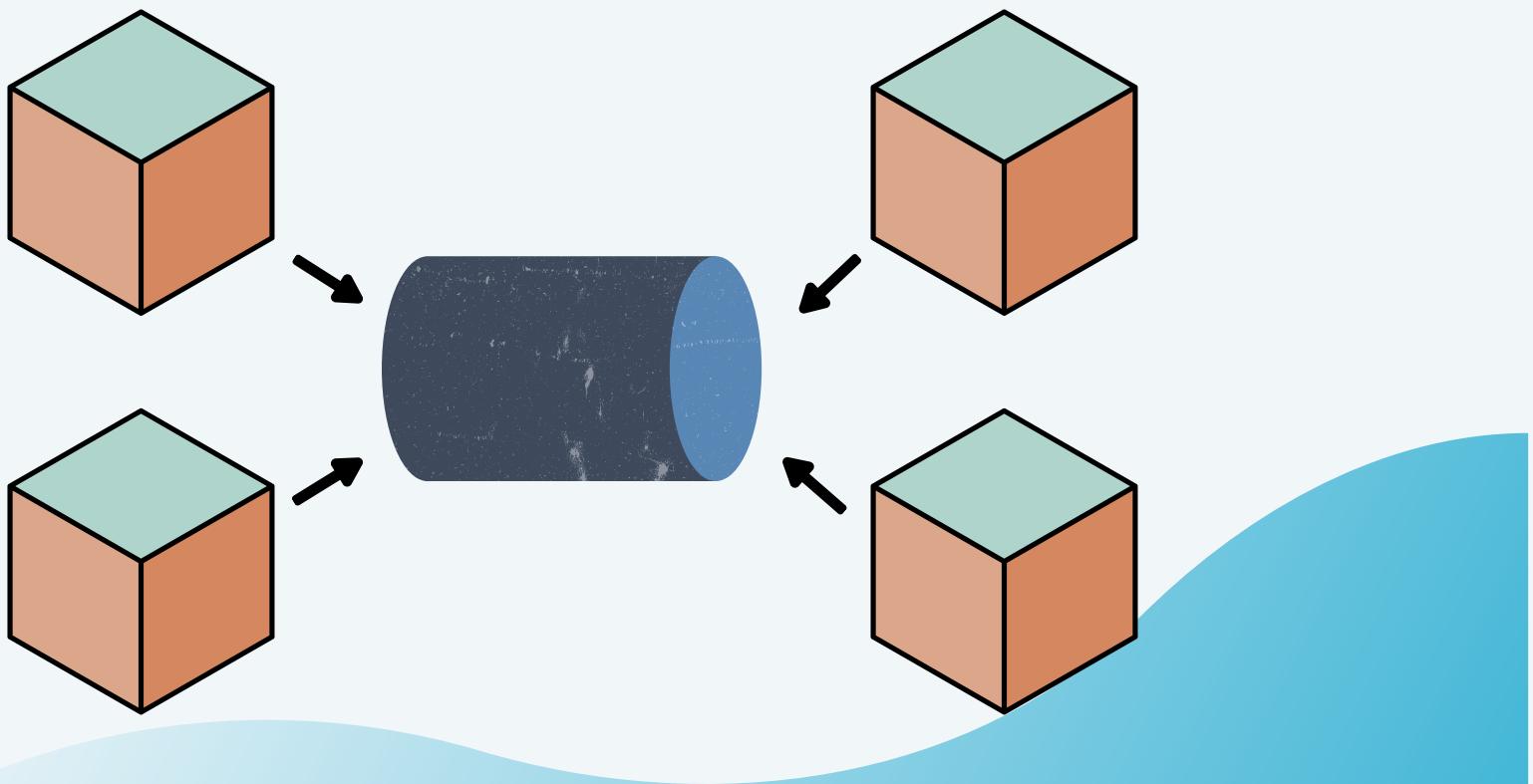




# Arquitectura Orientada a Eventos

**Una Guía para entender todas sus claves y su potencial en sistemas distribuidos**



PABLO DEL ÁLAMO



# Qué es la Arquitectura Orientada a Eventos

Una arquitectura orientada a eventos se centra en el flujo y procesamiento de eventos transmitidos entre servicios.

Es como un sistema de mensajería, en el que en vez de haber una comunicación directa entre los servicios, hay un ente intermedio que los comunica (sería como un mensajero comunicando por carta a 2 personas, en vez de que estas se comuniquen directamente).



PABLO DEL ÁLAMO



PABLO DEL ÁLAMO



# ¿Por qué eventos?

Con los eventos, puedes desencadenar acciones y comunicar servicios de manera asíncrona.

Así, los servicios no están esperando constantemente una respuesta, si no que envían el mensaje y listo.

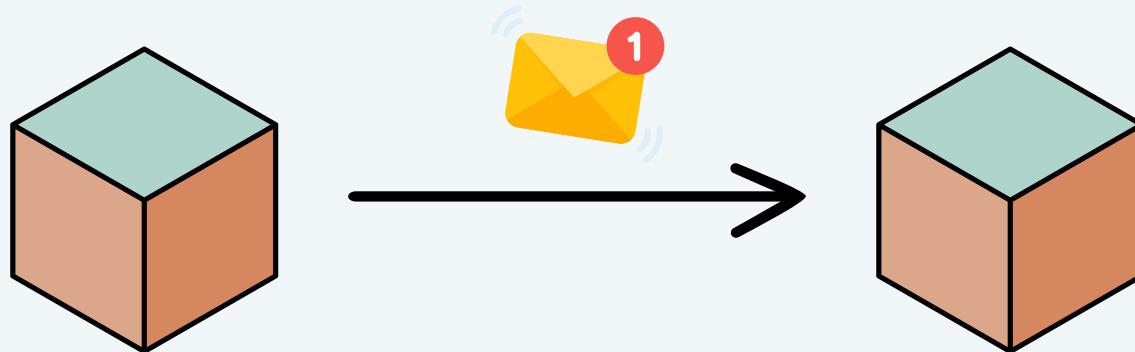
Además, permite que un mismo servicio comunique el mismo mensaje a múltiples servicios con una sola transacción.



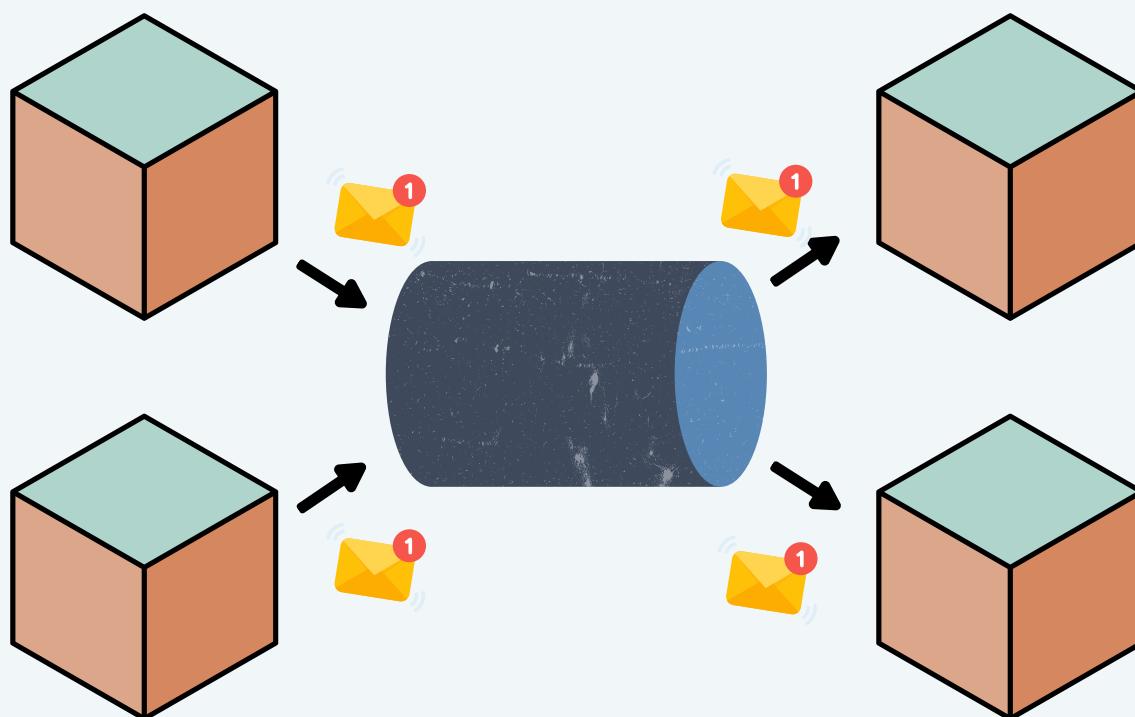
PABLO DEL ÁLAMO



## Comunicación directa



## Arquitectura orientada a eventos



PABLO DEL ÁLAMO



# Beneficios Clave

1. Escalabilidad
2. Reactividad
3. Flexibilidad
4. Desacoplamiento entre componentes



PABLO DEL ÁLAMO

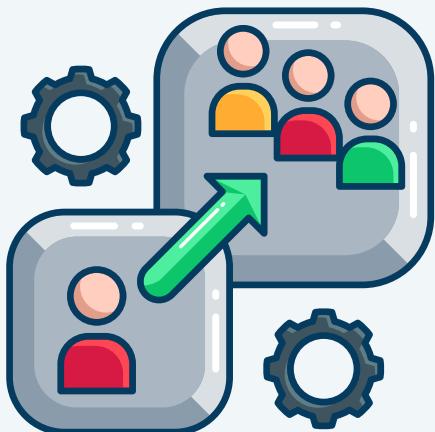


# Escalabilidad

Tu app puede crecer sin sufrir, porque cada servicio trabaja por su cuenta, y solo actúa cuando recibe un evento.

Así, si aumentan los usuarios, puedes agregar más instancias de cada servicio sin que el sistema se quede lento.

Perfecto para apps que necesitan estar listas para grandes volúmenes de tráfico.



PABLO DEL ÁLAMO



# Reactividad

Esta arquitectura hace que tu app responda en tiempo real: apenas ocurre un evento, el sistema reacciona al instante.

Ideal para apps donde los usuarios quieren ver cambios al segundo, como en chats, redes sociales o sistemas de trading.



PABLO DEL ÁLAMO

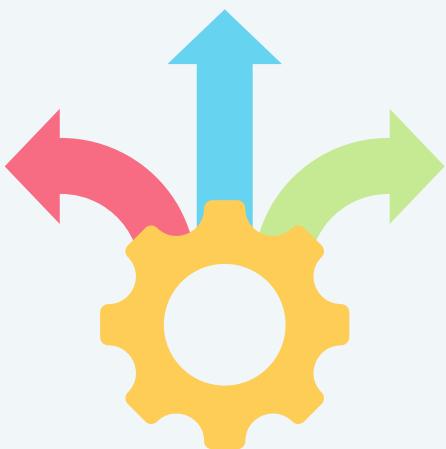


# Flexibilidad

Si tienes que añadir nuevas funciones o ajustar alguna parte, es súper fácil.

Solo tienes que agregar un nuevo módulo que escuche eventos ya existentes o envíe nuevos, sin tocar todo lo demás.

Así, el sistema se adapta rápido a cualquier cambio que necesites.



PABLO DEL ÁLAMO



# Desacoplamiento entre Componentes

Aquí, cada pieza del sistema va por libre.

Los componentes no dependen unos de otros, solo tienen que estar pendientes de los eventos relevantes.

Esto permite que los desarrolladores trabajen en distintas partes sin pisarse, y hace que todo sea más fácil de mantener y actualizar.



PABLO DEL ÁLAMO



# Componentes Principales

Tres componentes esenciales: emisor, consumidor y el evento en sí.

Piénsalo, ¡es como un tweet siendo retuiteado!



PABLO DEL ÁLAMO



# Ejemplo

**Los emisores lanzan eventos y los consumidores los escuchan y actúan.**

**Un emisor podría ser un microservicio que completa el proceso de compra, y manda un mensaje con los detalles de la compra, que reciben los consumidores, que podrían ser un ms de gestión de stock (tras recibir el mensaje reduce el stock), y uno de preparar medido (al recibir el mensaje notifica al almacén para que comiencen a preparar el pedido)**



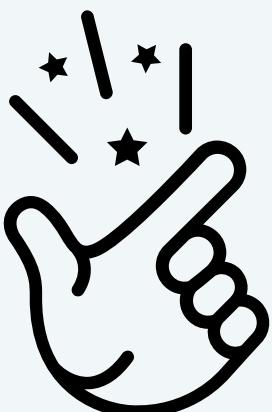
PABLO DEL ÁLAMO



# Procesamiento de Eventos

Hasta ahora hemos hablado de los eventos (mensajes), pero no hemos hablado de cómo se gestionan.

En las diferentes diapositivas, habrás visto que siempre que ilusto la arquitectura orientada a eventos, pongo un cilindro en medio de los servicios. Bien, esto es una cola de mensajería de un event broker, que gestiona los eventos, veamos cómo.



PABLO DEL ÁLAMO



# Qué es un Event Broker

Un event broker (como Apache Kafka, del que tengo otro post hablando en detalle para que aprendas a usarlo 😊) recibe los eventos, y se asegura de que lleguen a los componentes interesados (es el mensajero).

Así, los componentes no se comunican directamente entre sí, sino a través de este intermediario, lo que permite que estén desacoplados y funcionen independientemente.



PABLO DEL ÁLAMO



**El event broker (como Kafka, RabbitMQ o AWS SNS/SQS) actúa como el intermediario central que recibe, organiza y distribuye los eventos a los servicios que necesitan procesarlos.**

Para hacer esto de manera eficiente, el broker usa colas de mensajería para almacenar los eventos temporalmente hasta que los receptores (o consumidores) los procesen.



PABLO DEL ÁLAMO



# ¿Qué son las colas de mensajería?

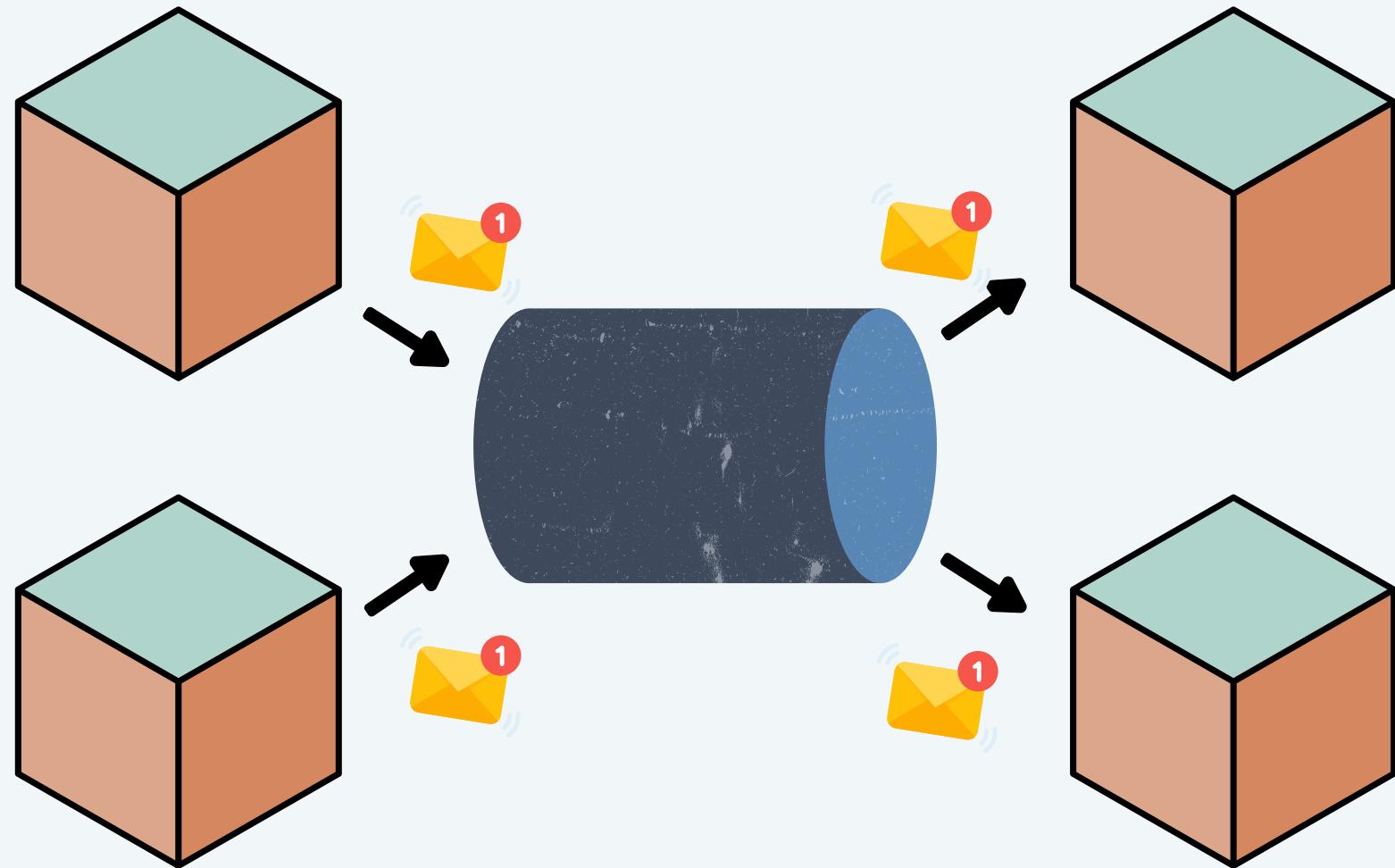
Las colas de mensajería son herramientas que ayudan a gestionar los eventos que se envían en la arquitectura.

Piensa en ellas como una "bandeja de entrada" (el cilindro que he puesto ya en varias diapositivas) donde se ponen los eventos en espera hasta que el componente receptor esté listo para procesarlos.

Esto es súper útil para manejar muchos eventos sin sobrecargar el sistema, ya que cada componente puede procesarlos a su ritmo.



PABLO DEL ÁLAMO



PABLO DEL ÁLAMO



# Guía Completa y Práctica de Arquitectura Orientada a Eventos + Kafka

Si quieres entrar en los detalles técnicos, y aprender a implementar una arquitectura de este estilo, tienes otro post en mi perfil donde hablo sobre cómo hacerlo con Apache Kafka al detalle.



PABLO DEL ÁLAMO



Pablo del Álamo Rodríguez • Tú

Software Engineer | AWS Certified Developer | Backend Develop...

1 semana •

...

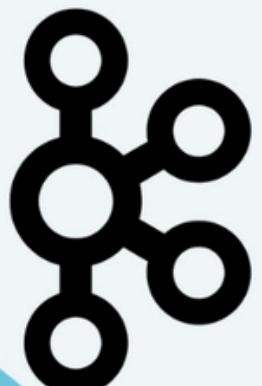
¿Te gustaría trabajar con datos en tiempo real? Kafka es la respuesta.

¡Vamos a explorar cómo! ... más



# KAFKA + ZOOKEEPER

De 0 a expert@



PABLO DEL ÁLAMO



PABLO DEL ÁLAMO



# AWS y Arquitectura de Eventos

En AWS, puedes usar servicios como SNS, SQS y Lambda para crear flujos de eventos robustos.

Haré una guía completa en detalle para mostrar cómo, pero si ya tienes la base de esta guía + la guía de Kafka, te resultará súper sencillo de entender



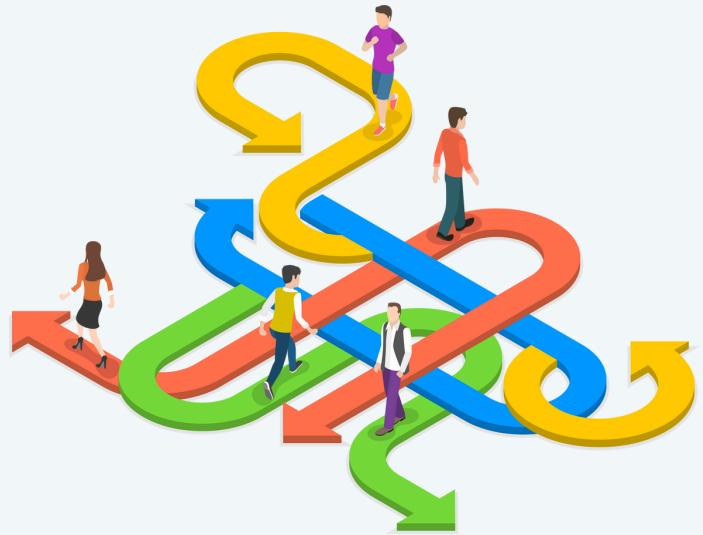
PABLO DEL ÁLAMO



# Desventajas a Considerar

Como todo en esta vida, ¡No todo es color de rosa!

La complejidad aumenta y necesitas gestionarla bien; si no, será un caos.

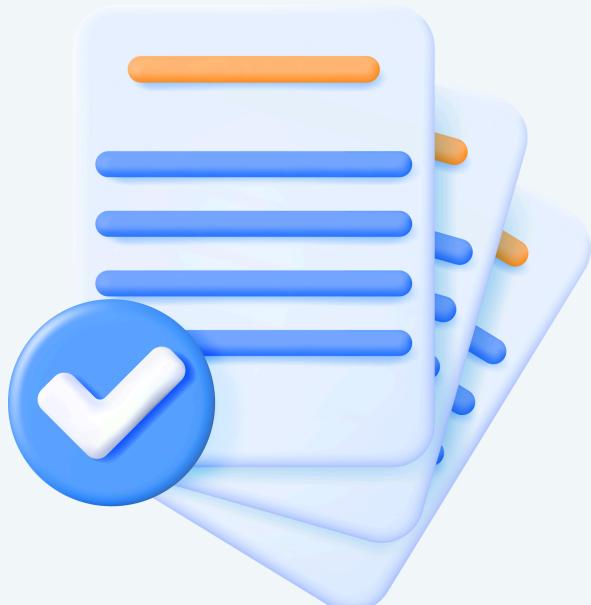


PABLO DEL ÁLAMO



# Casos de Uso

A continuación te dejo diferentes ejemplos de casos de uso típicos, en los que usar este tipo de arquitecturas tiene mucho sentido.



PABLO DEL ÁLAMO



# Sistemas de Recomendación

Utiliza eventos de usuario (como clics, compras, etc.) para generar recomendaciones en tiempo real.



PABLO DEL ÁLAMO



# Procesamiento de Transacciones Financieras

Captura eventos de transacciones y actualiza el estado de la cuenta o envía notificaciones a otros sistemas de forma asincrónica.



PABLO DEL ÁLAMO



# Monitoreo de IoT

Recibe eventos de dispositivos IoT, permite el análisis en tiempo real y activa acciones basadas en esos eventos (como alertas).



PABLO DEL ÁLAMO



# Gestión de Pedidos

En una tienda en línea, cada evento de pedido puede desencadenar procesos como la gestión de inventario, el envío de confirmaciones y la notificación al usuario.



PABLO DEL ÁLAMO



# Sistemas de Chat y Mensajería

Cada mensaje enviado o recibido se trata como un evento, permitiendo un manejo asincrónico y mejorando la escalabilidad.

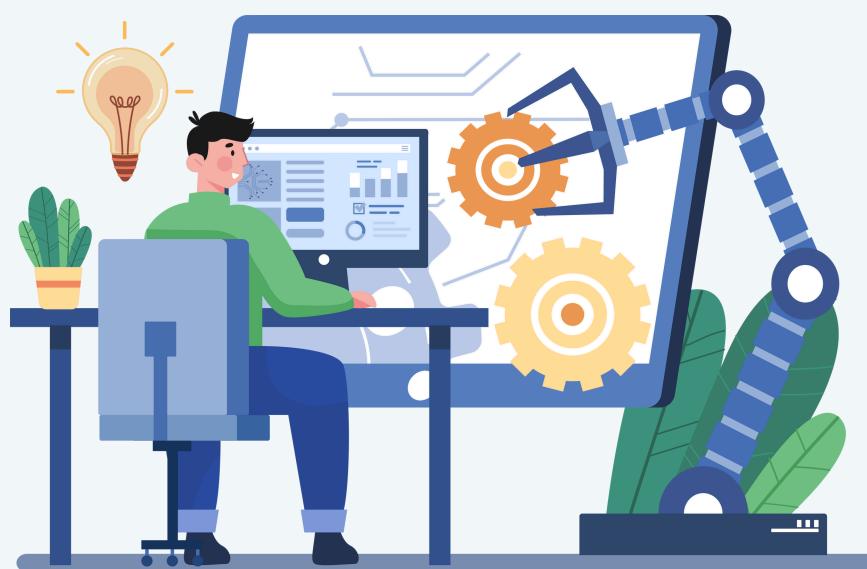


PABLO DEL ÁLAMO



# Automatización de Flujos de Trabajo

Usa eventos para desencadenar diferentes etapas en un flujo de trabajo, como la aprobación de documentos o la actualización de estados de proyectos.



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!

