

**Arquitecturas de Software**

# **Arquitectura dirigida a eventos (EDA).**

Basilio Franco Miguel Angel  
Mendoza Ramírez Omar  
Sánchez Barrera María del Rocío

Enfoque en el que los componentes de un sistema interactúan mediante eventos. Los eventos son cambios significativos en el estado del sistema que desencadenan acciones o respuestas.

# Estructura

## **Emisores de Eventos (Producers)**

Componentes que generan eventos cuando ocurre una acción o cambio de estado.

## **Canales de Eventos**

Medio de comunicación que transporta los eventos desde los emisores hasta los consumidores.

## **Consumidores de Eventos (Consumers)**

Componentes que escuchan o reciben eventos y toman acciones en consecuencia.

# Estructura

## Procesadores de Eventos

Elementos que procesan los eventos entrantes, realizando tareas específicas, y pueden generar nuevos eventos como resultado.

## Middleware de Eventos

Sistemas intermedios que gestionan, enrutan y almacenan eventos, como sistemas de mensajería (ej. RabbitMQ, Kafka).

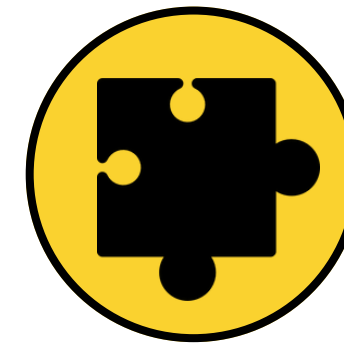
## Colas de Mensajes

Almacenes temporales que mantienen los eventos hasta que los consumidores los procesan.

# Características

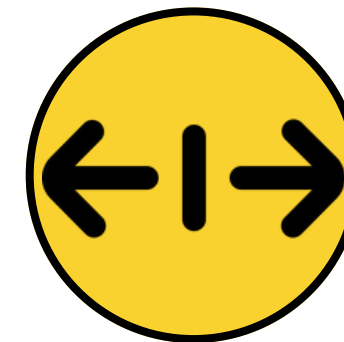
## Desacoplamiento

Los componentes de la arquitectura están desacoplados, ya que los emisores y consumidores no necesitan conocerse entre sí.



## Escalabilidad

Se adapta fácilmente a un gran número de emisores y consumidores, lo que facilita la escalabilidad horizontal.



## Reactiva

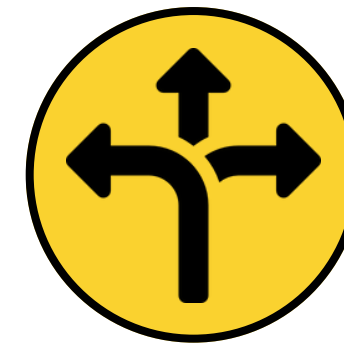
Los consumidores reaccionan a los eventos en tiempo real, lo que permite un sistema más dinámico y adaptable.



# Características

## Flexibilidad

Permite la integración con diferentes sistemas y tecnologías a través de eventos comunes.



## Asíncrona

Las operaciones no necesitan ocurrir en tiempo real, ya que los eventos pueden ser almacenados y procesados cuando sea necesario.





# Ventajas

1. Escalabilidad
2. Flexibilidad
3. Eficiencia
4. Desacoplamiento
5. Mejora de la respuesta
6. Mayor tolerancia a fallos

# Desventajas

1. Complejidad
2. Depuración complicada
3. Dificultad para garantizar la consistencia
4. Requisitos de infraestructura



**EVENTO**



Cola de eventos

**MEDIADOR DE EVENTOS**

**Canal del evento**

**Canal del evento**

**Canal del evento**

**Procesamiento del  
evento**

**Modulo**

**Modulo**

**Modulo**

**Modulo**

**Procesamiento del  
evento**

**Modulo**

**Modulo**

**Modulo**

**Modulo**

**Procesamiento del  
evento**

**Modulo**

**Modulo**

**Modulo**

**Modulo**

**Procesamiento del  
evento**

**Modulo**

**Modulo**

**Modulo**

**Modulo**

**Procesamiento del  
evento**

**Modulo**

**Modulo**

**Modulo**

**Modulo**

# Ejemplos

## PayPal



Se generan múltiples eventos que activan procesos como la verificación de pagos, notificación al vendedor, y actualizaciones de saldo.

## Uber



Se basa en eventos para coordinar todas las interacciones en su plataforma, asegurando que las respuestas del sistema sean rápidas y escalables

## Amazon Web Services (AWS)



Los desarrolladores pueden definir funciones que se activan por diferentes eventos, como cambios en bases de datos, subidas de archivos o solicitudes HTTP a través de API Gateway.



# Resumen

Permite que los componentes funcionen de manera independiente, facilitando la escalabilidad, el mantenimiento y la flexibilidad del sistema. La integración de nuevos servicios o funcionalidades se logra con facilidad, ya que los componentes solo necesitan suscribirse a los eventos pertinentes. Además, al ser asíncrona, mejora la capacidad de respuesta y resiliencia del sistema frente a fallos o picos de carga, adaptándose de manera eficiente a diferentes condiciones.

**Muchas  
gracias**