

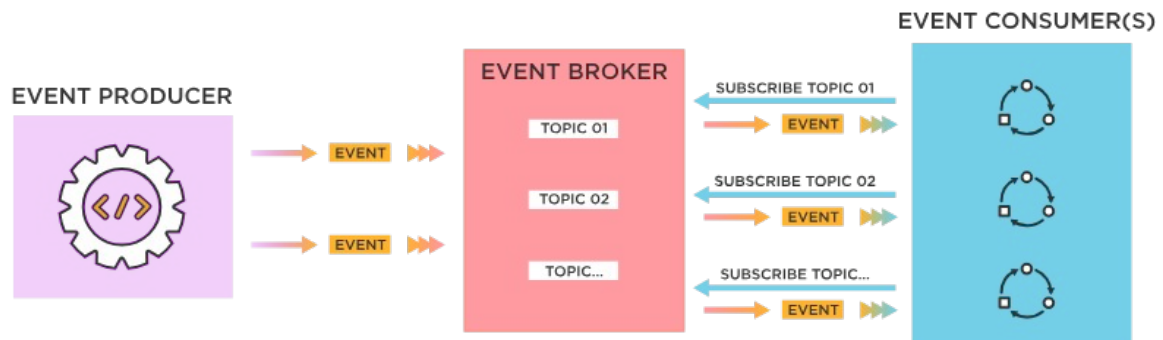
# **Arquitectura de Software**

## ***Glosario de conceptos***

**JONATHAN DARIO  
SIERRA GALINDO  
KAROLINA SEGURA  
PERDOMO**

## 1. Arquitectura de Eventos (Event-Driven Architecture)

La arquitectura basada en eventos es un modelo que permite que los componentes de software se comuniquen de manera asíncrona mediante la generación, captura y procesamiento de eventos. Los eventos representan cambios en el estado de un sistema o acción de un usuario, y estos se propagan sin esperar una respuesta inmediata, lo que aumenta la escalabilidad y flexibilidad.



Ejemplo:

En un sistema de procesamiento de pagos, cuando un usuario realiza una compra, se genera un evento de "compra realizada". Este evento es procesado por servicios independientes que verifican el estado de la tarjeta, autorizan el pago y notifican al usuario sin que estos servicios dependan unos de otros.

## Referencias

Chen, P. (2020). Event-Driven Architecture: A Comprehensive Guide. Tech Publishing.

## 2. Patrón Modelo-Vista-Controlador (MVC)

El patrón MVC separa una aplicación en tres componentes principales: el Modelo, la Vista y el Controlador. Esta arquitectura es ampliamente utilizada en el desarrollo de aplicaciones web para mejorar la organización y reutilización del código.



- Modelo: Gestiona los datos y la lógica de negocio.
- Vista: Presenta la información al usuario.
- Controlador: Enlaza las interacciones del usuario con el Modelo.

Ejemplo:

Una tienda en línea utiliza MVC para manejar el flujo de datos entre la base de datos (Modelo), la interfaz de usuario (Vista), y las acciones como "añadir al carrito" (Controlador).

Referencias

Fowler, M. (2019). Patterns of Enterprise Application Architecture. Addison-Wesley.

### 3. Arquitectura Orientada a Servicios (SOA)

SOA es un enfoque de diseño de software en el que las funcionalidades del sistema se encapsulan en servicios re utilizables que interactúan mediante interfaces bien definidas. Los servicios pueden ser utilizados por diferentes aplicaciones dentro de una organización, facilitando la interoperabilidad y el acoplamiento suelto.



Caso de Uso:

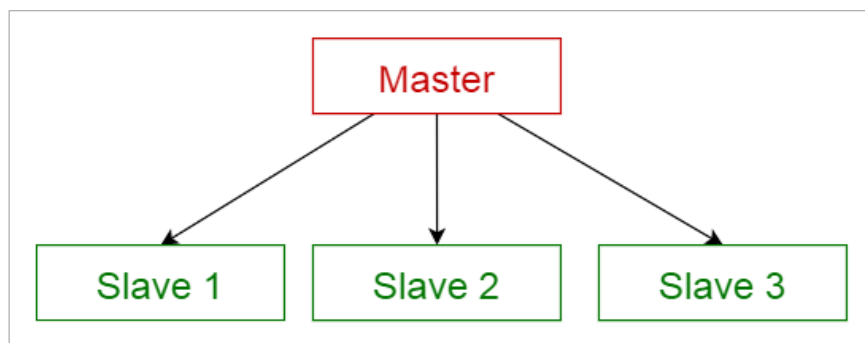
En una empresa financiera, los servicios de "verificación de crédito" y "procesamiento de pagos" se reutilizan en varias aplicaciones, como la plataforma de préstamos y el sistema de tarjetas de crédito.

Referencias

Papazoglou, M. P., & Heuvel, W. V. D. (2007). Service-Oriented Architectures: Approaches, Technologies and Research Issues. The Computer Journal, 48(3), 377–390.

#### 4. Patrón Maestro-Esclavo (Máster-Slave)

Este patrón de arquitectura organiza un sistema en el que una entidad principal, denominada "maestro", delega tareas a una o más entidades "esclavas", que ejecutan las tareas y reportan los resultados al maestro.



Ejemplo:

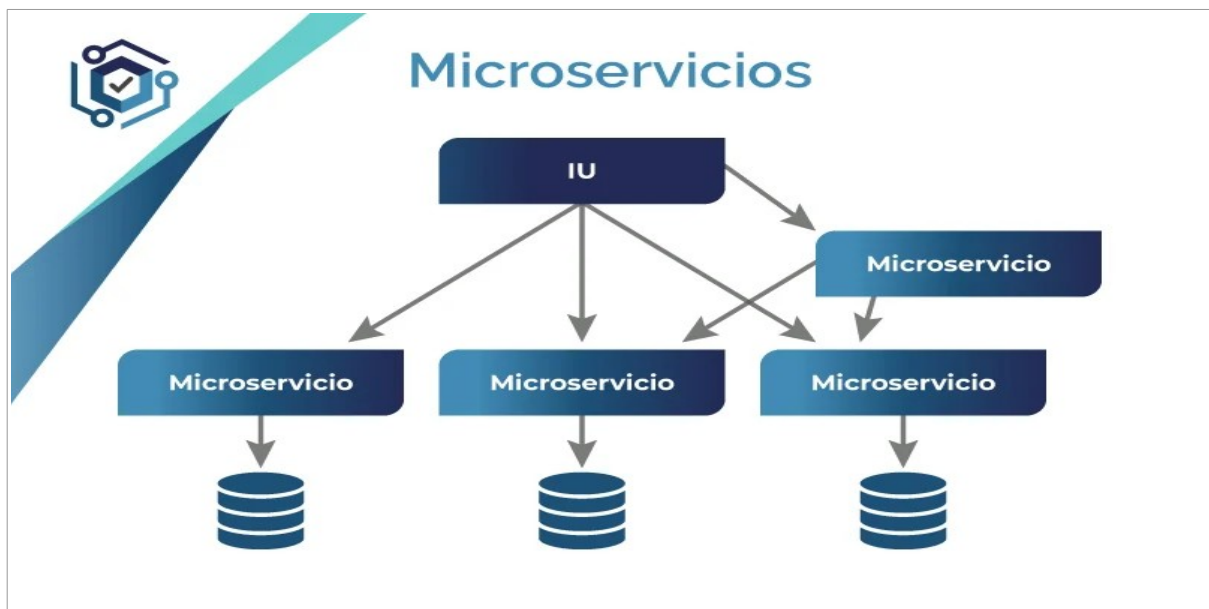
En un sistema de bases de datos distribuido, el nodo maestro gestiona las solicitudes de escritura, mientras que los nodos esclavos replican y manejan las consultas de lectura para mejorar el rendimiento.

## Referencias

Kumar, V. (2018). Distributed Systems: Concepts and Design. Pearson Education.

## 5. Arquitectura de Microservicios

Este enfoque de arquitectura divide una aplicación en pequeños servicios independientes que pueden ser desplegados y gestionados de forma autónoma. Cada microservicio tiene su propia lógica de negocio y base de datos, lo que facilita la escalabilidad y el despliegue continuo.



### Caso de Uso:

En una plataforma de comercio electrónico, diferentes microservicios se encargan de gestionar el catálogo de productos, el procesamiento de pagos, y el servicio de recomendaciones, lo que permite escalar cada componente de forma independiente.

## Referencias

Newman, S. (2021). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.