# Logotipo Descripción generada automáticamente

# 

# **DESARROLLO FULL STACK**

Actividad 1

“Investigación de arquitecturas”

**INTEGRANTES:**

Bastián Reyes

Cristian Lizama

Joshua mardones

**PROFESOR:**

MARIANGELES ROBINSON ALHUCEMA

**SECCIÓN:**

# 002D

**FECHA:**

16 de marzo de 2025

**Introducción**

En el presente informe responderemos y explicaremos los tipos de arquitectura que nos encomendaron. En el desarrollo de aplicaciones y sistemas informáticos, es fundamental comprender las diferentes arquitecturas de software existentes. Cada una ofrece un enfoque distinto para organizar, estructurar y gestionar tanto el hardware como el software de una aplicación, permitiendo resolver problemas específicos y adaptarse a las necesidades de los usuarios y empresas.

**1. Arquitectura de Capas (N-tier)**

1.1 definición:

Una arquitectura de N niveles divide una aplicación en capas lógicas y niveles físicos. Las capas son una manera de separar las responsabilidades y administrar las dependencias. Cada capa tiene una responsabilidad específica.

1.2 explica con tus palabras:

La arquitectura de N niveles organiza una aplicación en diferentes partes o secciones, desde el punto de vista lógico como físico las capas lógicas sirven para dividir el trabajo o las funciones . esto ayuda a organizar mejor el código y permite controlar el cómo se relacionan las diferentes partes entre sí.

1.3 beneficios

* **Mejor organización y mantenimiento** Al dividir la aplicación en capas, es más fácil entender, modificar y mantener el código. Si algo falla en una capa, no afecta directamente a las demás.
* **Escalabilidad** Permite escalar partes específicas del sistema (como la base de datos o la lógica de negocio) sin tener que modificar todo el sistema.

1.4 desventajas

* **Mayor complejidad** A medida que se añaden más capas, el sistema se vuelve más complejo de gestionar y entender, sobre todo para desarrollos pequeños.
* **Rendimiento** La comunicación entre capas puede generar más tiempo de respuesta, ya que hay más pasos por donde pasan los datos.

1.5 empresas que la utilizan

* **Amazon** Su arquitectura maneja múltiples capas para gestionar los servicios de la tienda en línea, bases de datos, sistemas de pagos, etc.
* **Microsoft** Muchas de sus aplicaciones empresariales y servicios en la nube (Azure, por ejemplo) utilizan arquitecturas de varias capas o niveles.

**2. Arquitectura Orientada a Servicios (SOA)**

2.1 definición:

La arquitectura orientada a servicios (SOA) es un paradigma de diseño que crea o complementa aplicaciones con bloques de construcción autónomos y reutilizables llamados servicios. Estos servicios proporcionan funciones específicas y se pueden invocar a través de una red. SOA permite la escalabilidad, flexibilidad, integración de aplicaciones y reutilización de componentes, reduciendo el costo y el tiempo de desarrollo

2.2 explica con tus palabras:

Es una forma de construir sistemas dividiéndolos en servicios independientes que se comunican entre sí. Cada servicio hace una tarea específica y puede ser usado por diferentes aplicaciones.

2.3 beneficios

**Interoperabilidad entre sistemas diferentes**

* SOA permite que distintos sistemas, aunque estén hechos en diferentes lenguajes o tecnologías, puedan comunicarse y trabajar juntos a través de servicios.

**Reutilización de servicios**

* Los servicios desarrollados pueden ser reutilizados por distintas aplicaciones o áreas de una empresa, ahorrando tiempo y recursos en el desarrollo.

2.4 desventajas

**Complejidad en la gestión y seguridad**

* Al tener muchos servicios distribuidos, se vuelve más complejo administrar la seguridad, el control de acceso y el mantenimiento de todos los servicios.

**Rendimiento más lento**

* Las comunicaciones entre servicios, especialmente si usan protocolos como SOAP o REST a través de la red, pueden hacer que el sistema sea más lento que una arquitectura monolítica.

2.5 empresas que la utilizan

* IBM  
   Ha sido una de las pioneras en promover SOA y ofrece soluciones empresariales basadas en esta arquitectura, como IBM WebSphere.
* Oracle  
   Ofrece una suite completa para implementar SOA, incluyendo Oracle SOA Suite, que muchas empresas usan para integrar sus sistemas.

### **3. Arquitectura de Microservicios**

3.1 definición:

La Arquitectura de Microservicios es un enfoque más moderno donde en vez de tener un sistema grande con varios servicios dentro (como en SOA), se divide todo en servicios pequeños e independientes. Cada uno se encarga de una función específica y se comunica con los demás a través de APIs. La idea es que cada microservicio pueda desarrollarse, actualizarse y escalarse por separado, sin afectar al resto de la aplicación. Es como si en vez de tener un solo equipo manejando todo el sistema, tuvieras varios equipos más pequeños, cada uno enfocado en una tarea específica y trabajando de forma independiente.

3.2 explica con tus palabras:

divide un sistema grande en pequeños servicios independientes, cada uno encargado de una función específica. Estos servicios se comunican entre sí a través de APIs y pueden desarrollarse, actualizarse y escalarse por separado, sin afectar el resto del sistema.

#### 3.3 Beneficios

* **Escalabilidad**: Cada servicio puede crecer de manera independiente según la demanda.
* **Resiliencia**: Un fallo en un servicio no afecta a toda la aplicación.
* **Flexibilidad tecnológica**: Se pueden usar distintos lenguajes y frameworks en cada servicio.

#### 3.4 Desafíos

* **Complejidad**: Se requieren estrategias avanzadas para gestionar múltiples servicios.
* **Latencia**: La comunicación entre servicios puede generar retrasos.
* **Seguridad**: Al haber más puntos de acceso, aumenta la superficie de ataque.

#### 3.5 Empresas que la utilizan

**Amazon** Su arquitectura maneja múltiples capas para gestionar los servicios de la tienda en línea, bases de datos, sistemas de pagos, etc.

**Netflix** Migró de un sistema monolítico a microservicios para mejorar la escalabilidad y ofrecer una mejor experiencia de streaming.

### **4.Arquitectura Multitenant.**

4.1 definición:

Es un modelo en el que una única instancia de una aplicación sirve a varios clientes (tenants), cada uno con sus propios datos y configuraciones, pero compartiendo la misma infraestructura y recursos. Los datos de cada cliente están aislados para garantizar la seguridad y privacidad, mientras se optimizan los costos y el uso de los recursos.

3.2 explica con tus palabras:

según mis palabras: cuando una sola aplicación sirve a varios clientes, pero cada uno tiene sus datos y configuraciones separadas. Todos comparten los mismos recursos, lo que ayuda a reducir costos, pero se asegura que los datos de cada cliente estén protegidos.

#### 4.3 Beneficios

**Eficiencia en costos**: Un solo sistema atiende a múltiples clientes sin duplicar infraestructura.

**Mantenimiento centralizado**: Las actualizaciones se aplican a todos los clientes a la vez.

**Escalabilidad**: Se pueden agregar nuevos clientes sin afectar el rendimiento general.

**Uso óptimo de recursos**: Se maximiza la capacidad del hardware y la base de datos.

#### 4.4 Desafíos

* **Seguridad**: Es fundamental evitar que los datos de un cliente sean accesibles por otro.
* **Personalización limitada**: La configuración suele ser estándar para todos los clientes.
* **Carga de trabajo desbalanceada**: Algunos clientes pueden consumir más recursos que otros.
* **Gestión de datos compleja**: Se requiere una estructura bien diseñada para evitar conflictos entre clientes.

#### 4.5 Empresas que la utilizan

**Salesforce** Implementa una arquitectura multitenant para ofrecer su CRM a múltiples empresas con datos aislados en una misma plataforma.

**Google Workspace** Gmail, Drive y Docs operan en un entorno multitenant, permitiendo que diferentes usuarios compartan la misma infraestructura de forma segura.

**5.Arquitectura Event-Driven**

5.1 definición:

La Arquitectura Event-Driven (o Dirigida por eventos) utiliza eventos para desencadenar y establecer comunicación entre servicios desacoplados. Un evento es un cambio de estado, o una actualización, como un elemento que se coloca en un carro de compras de un sitio web de comercio electrónico. Los eventos pueden llevar el estado (el elemento comprado, su precio y una dirección de entrega) o pueden ser identificadores (una notificación de que se envió una orden).

5.2 explica con tus palabras:

Básicamente, un sistema reacciona a eventos en lugar de seguir un orden fijo. Un evento puede ser, por ejemplo, hacer clic en un botón, recibir un mensaje o cambiar un dato.

#### 5.3 Beneficios

* Los sistemas pueden adaptarse mejor a los cambios.
* Los componentes no dependen directamente unos de otros, facilitando el mantenimiento.
* Solo se ejecutan tareas cuando ocurre un evento, reduciendo el uso de recursos.

#### 5.4 Desafíos

* Como los eventos ocurren en diferentes momentos, su ejecución es más complicada.
* Si hay demasiados eventos simultáneos, el sistema puede saturarse si no está bien diseñado.
* Difícil de diseñar.

#### 5.5 Empresas que la utilizan

**Amazon (AWS)**

Usa Amazon EventBridge, SQS y SNS para conectar servicios mediante eventos.

**DoorDash**

Usa la arquitectura para gestionar pedidos de comida y coordinar repartidores en tiempo real.

### **6.Arquitectura Micro-FrontEnds**

6.1 definición:

La arquitectura de micro-frontends concibe el sitio web como un conjunto de aplicaciones independientes, cada una de las cuales se desarrolla y despliega de forma autónoma por su respectivo equipo de desarrollo.

Como resultado, cada equipo forma parte de un área de negocio y desarrolla su funcionalidad de manera independiente al resto y sin dependencias externas. Esto permite una mayor flexibilidad y reduce el acoplamiento entre los diferentes módulos o aplicaciones que componen el sitio.

6.2 explica con tus palabras:

Es una forma de construir aplicaciones web dividiéndolas en partes más pequeñas. Cada parte se maneja de forma independiente, pero todas juntas hacen la aplicación completa.

#### 6.3 Beneficios

* Los equipos pueden desarrollar y desplegar micro-aplicaciones de manera autónoma, acelerando el tiempo de entrega.
* Cada micro-aplicación puede usar tecnologías diferentes, permitiendo la adopción de nuevas tecnologías sin afectar toda la aplicación
* Es más fácil de mantener y escalar, ya que cada micro-aplicación puede ser actualizada o reemplazada sin afectar el resto de la interfaz.

#### 

#### 6.4 Desafíos

* La gestión de múltiples micro frontends puede ser complicada, especialmente en cuanto a la integración y coordinación entre ellos.
* La carga de recursos y la comunicación entre micro frontends pueden afectar el rendimiento si no se manejan bien.
* Si no se controla, puede haber duplicación de funcionalidades entre micro frontends.

#### 6.5 Empresas que la utilizan

**Ticketmaster**

Usa micro frontends para permitir que diferentes equipos gestionen las partes de la plataforma, como la compra de entradas y el perfil de usuario, de manera autónoma.

**Amazon**

Implementa micro frontends para manejar sus diferentes servicios y equipos de forma separada, mejorando la flexibilidad.