

Nombre: Roberto Carlos Aruquipa Arzala

Materia: Lenguaje Audiovisual ISI-356

Fecha: 30/09/2023

Trabajo de Investigación

Arquitecturas de desarrollo de software más usadas en la actualidad y dar una explicación de cómo funcionan

La arquitectura de desarrollo de software se refiere a la estructura organizativa y el diseño general de un sistema de software. Esta arquitectura define cómo se componen y relacionan las diferentes partes del software, incluyendo componentes, módulos, capas y servicios, para lograr los objetivos funcionales y no funcionales del sistema. En esencia, es la base sobre la cual se construye y se organiza todo el software.

La arquitectura de desarrollo de software aborda preguntas clave sobre cómo se organizarán y comunicarán los componentes del sistema, cómo se gestionarán los datos, cómo se garantizará la escalabilidad, el rendimiento y la seguridad, y cómo se mantendrá y extenderá el sistema con el tiempo. También puede definir patrones de diseño, estándares y mejores prácticas a seguir durante el proceso de desarrollo.

Una arquitectura de desarrollo de software efectiva puede conducir a sistemas más robustos, mantenibles y escalables, mientras que una arquitectura deficiente puede generar problemas de rendimiento, dificultades en la implementación de cambios y problemas de mantenimiento. Por lo tanto, la elección de una arquitectura adecuada es una parte crítica del proceso de desarrollo de software.

En la actualidad, existen diversas arquitecturas de desarrollo de software ampliamente utilizadas para crear aplicaciones de alta calidad y escalables. A continuación, se describen algunas de las arquitecturas más comunes y una breve explicación de cómo funcionan:

1. Arquitectura Modelo-Vista-Controlador (MVC):

- **Descripción:** MVC es uno de los patrones de diseño más conocidos y se utiliza para separar la lógica de la aplicación en tres componentes principales: Modelo, Vista y Controlador. El Modelo se encarga de los datos y la lógica de negocio, la Vista es responsable de la interfaz de usuario y el Controlador maneja las interacciones entre el Modelo y la Vista.

- **Funcionamiento:** Cuando un usuario interactúa con la Vista, el Controlador procesa la entrada y actualiza el Modelo según sea necesario. Luego, la Vista se actualiza automáticamente para reflejar los cambios en el Modelo. Esta separación permite una fácil mantenibilidad y escalabilidad, ya que los componentes pueden modificarse independientemente.

2. Arquitectura Cliente-Servidor:

- **Descripción:** En la arquitectura cliente-servidor, las aplicaciones se dividen en dos partes principales: el Cliente y el Servidor. El Cliente es la interfaz de usuario que interactúa con el usuario final, mientras que el Servidor maneja la lógica de negocio y la gestión de datos.
- **Funcionamiento:** El Cliente realiza solicitudes al Servidor para obtener o enviar datos. El Servidor procesa estas solicitudes y envía respuestas al Cliente. Esto permite la distribución de la carga y la escalabilidad, ya que múltiples Clientes pueden conectarse al mismo Servidor.

3. Arquitectura de Microservicios:

- **Descripción:** Los microservicios son una arquitectura moderna que divide una aplicación en pequeños servicios independientes. Cada servicio realiza una función específica y se comunica con otros a través de API o protocolos definidos.
- **Funcionamiento:** Cada microservicio se ejecuta en su propio entorno y puede desarrollarse, implementarse y escalarse de manera independiente. Esto mejora la flexibilidad, la escalabilidad y la facilidad de mantenimiento. Sin embargo, requiere una gestión cuidadosa de las comunicaciones entre microservicios.

4. Arquitectura de Capas:

- **Descripción:** La arquitectura de capas organiza una aplicación en capas lógicas que se comunican entre sí de manera jerárquica. Las capas comunes incluyen la capa de presentación, la capa de lógica de negocio y la capa de acceso a datos.
- **Funcionamiento:** Cada capa tiene una función específica, y las capas superiores pueden acceder a las inferiores, pero no al revés. Esto facilita la modularidad y la separación de preocupaciones. Las aplicaciones de capas suelen ser altamente estructuradas y fáciles de mantener.

5. Arquitectura Orientada a Servicios (SOA):

- **Descripción:** SOA es una arquitectura que se basa en servicios independientes que se comunican a través de interfaces estándar. Cada servicio representa una funcionalidad específica de la aplicación.
- **Funcionamiento:** Los servicios se registran en un directorio o registro y pueden ser invocados por otros componentes de la aplicación cuando sea necesario. Esto promueve la reutilización de servicios y la interoperabilidad.