



Universidad Autónoma De Chiapas

Licenciatura en ingeniería en desarrollo y tecnologías de software

Act. 1.1 Definir los siguientes conceptos: Microservicios

Taller De Desarrollo 4

Dr. Luis Gutiérrez Alfaro

Alumno: Manuel Antonio Nuñez Sánchez

Matricula: A200884 "6°M"

Investigar lo siguientes términos:

API: “Interfaz de Programación de Aplicaciones” es un conjunto de reglas y protocolos que permiten que diferentes softwares se comuniquen entre sí. Es una forma estandarizada en la que las aplicaciones pueden solicitar y compartir datos o funcionalidades entre sí, sin necesidad de conocer los detalles internos de la implementación de la otra aplicación.

Las API son esenciales en el desarrollo de software, ya que proporcionan una manera de acceder y utilizar las características y datos de una aplicación sin tener que escribir todo el código desde cero. En lugar de eso, los desarrolladores pueden utilizar las API proporcionadas por otros programas o servicios para integrar funcionalidades específicas en sus propias aplicaciones.

Arquitectura: La arquitectura de microservicios es un enfoque de diseño de software en el que una aplicación se construye como un conjunto de servicios independientes y autónomos que se ejecutan en su propio proceso y se comunican entre sí a través de interfaces. Cada servicio se enfoca en una funcionalidad específica y puede ser desarrollado, desplegado y escalado de forma independiente.

Las características claves de la arquitectura de microservicios son:

Desacoplamiento, Escalabilidad, Desarrollo y Despliegue Independientes, Resistencia a Fallos, Escalabilidad Horizontal, Facilita la Innovación, Gestión de Datos, Complejidad de Orquestación.

la arquitectura de microservicios tiene ventajas claras en términos de escalabilidad, mantenimiento y flexibilidad, también introduce desafíos en cuanto a la gestión de la comunicación entre servicios, la seguridad y la administración de la complejidad. La elección de la arquitectura debe basarse en las necesidades específicas del proyecto y la comprensión de los beneficios y desafíos que conlleva cada enfoque.

Back End: se refiere a la parte de una aplicación de software que se encarga de la lógica y el procesamiento en segundo plano. Es la porción que no es visible para los usuarios finales, pero que es fundamental para el funcionamiento y la interacción efectiva de la aplicación. El back end se encarga de manejar los datos, la lógica de negocio, la autenticación, la seguridad y otros aspectos clave que permiten que una aplicación funcione correctamente.

es una parte esencial de cualquier aplicación de software, ya que maneja la lógica detrás de la interfaz de usuario y permite que la aplicación funcione de manera coherente, segura y eficiente. Su diseño y construcción es fundamental para el impacto en la experiencia del usuario y en la eficacia general de la aplicación.

Bifurcación: se refiere al acto de crear una copia de un proyecto de software existente con el propósito de continuar su desarrollo de manera independiente. Esta copia puede modificarse y evolucionar de manera separada del proyecto original. La bifurcación se utiliza a menudo en proyectos de código abierto y colaborativos, donde diferentes desarrolladores pueden tener visiones o enfoques distintos para el futuro del software.

es una estrategia en el desarrollo de software que permite a los desarrolladores crear una versión independiente de un proyecto existente.

Escalabilidad: se refiere a la capacidad de un sistema, aplicación o infraestructura para manejar un aumento en la carga de trabajo o la demanda sin comprometer su rendimiento, calidad o disponibilidad. En el contexto de la tecnología y la informática, la escalabilidad es esencial para garantizar que los sistemas puedan crecer y adaptarse a medida que aumentan los usuarios, los datos y las operaciones.

Hay dos tipos principales de escalabilidad:

Escalabilidad Vertical (Escalabilidad Up): En este enfoque, se mejora el rendimiento de un sistema aumentando la capacidad de los recursos existentes, como añadir más potencia de procesamiento, memoria RAM o recursos de almacenamiento a un único servidor. Sin embargo, este enfoque tiene un límite físico y puede volverse costoso a medida que se necesitan componentes más potentes.

Escalabilidad Horizontal (Escalabilidad Out): En este enfoque, se maneja un aumento en la carga de trabajo distribuyendo tareas entre múltiples instancias de servidores o máquinas. Este método es más flexible y permite un mayor grado de escalabilidad, ya que nuevos servidores pueden agregarse según sea necesario.

Flexibilidad: se refiere a la capacidad de un programa o sistema para adaptarse a diferentes situaciones, cambios en los requisitos y escenarios imprevistos sin requerir una reescritura completa o una modificación extensa del código. La flexibilidad es una característica deseable en el desarrollo de software, ya que permite a las aplicaciones ser más adaptables, mantenibles y escalables a medida que evolucionan con el tiempo.

Framework: es un conjunto predefinido de herramientas, bibliotecas, patrones de diseño y directrices que proporcionan una estructura para el desarrollo de software. Los frameworks permiten a los desarrolladores construir aplicaciones de manera más rápida y eficiente al proporcionar bloques de construcción listos para usar y establecer una arquitectura común.

Front End: se refiere a la parte de una aplicación de software que los usuarios interactúan directamente. Es la interfaz visible y la experiencia del usuario. En otras palabras, es la parte de una aplicación que se presenta en la pantalla y con la que los usuarios pueden interactuar visualmente y funcionalmente.

IaaS: es un modelo de entrega de servicios en la nube que proporciona a las organizaciones recursos informáticos virtualizados a través de Internet. En lugar de comprar y administrar hardware físico y recursos de infraestructura, las empresas pueden alquilar recursos de cómputo, almacenamiento, redes y otros componentes de infraestructura de un proveedor de servicios en la nube.

Microservicios: son un enfoque arquitectónico en el desarrollo de software donde las aplicaciones se descomponen en componentes pequeños, independientes y autónomos que se conocen como "microservicios".

Cada microservicio se encarga de una función específica y se comunica con otros microservicios a través de interfaces bien definidas, generalmente a través de APIs, son una arquitectura que promueve la modularidad, la escalabilidad y la independencia en el desarrollo de software. Permiten a las organizaciones construir aplicaciones más flexibles y adaptativas, facilitando la gestión y el mantenimiento a medida que la aplicación evoluciona con el tiempo.

PaaS: es un modelo de servicios en la nube que proporciona a los desarrolladores un entorno completo para construir, implementar y administrar aplicaciones sin preocuparse por la infraestructura subyacente. En lugar de preocuparse por la configuración y el mantenimiento de servidores, sistemas operativos y redes, los desarrolladores se centran en la codificación y el despliegue de sus aplicaciones.

Servicio: se refiere a una unidad de funcionalidad o recurso que se proporciona a través de una interfaz bien definida y accesible a través de una red o una API. Los servicios

permiten a las aplicaciones interactuar y compartir datos y funcionalidad de manera modular y distribuida.

Algunos puntos clave sobre los servicios en programación:

Abstracción: Los servicios abstractizan y encapsulan una funcionalidad específica, lo que permite a los desarrolladores utilizar esa funcionalidad sin necesidad de conocer los detalles internos de cómo funciona.

Comunicación: Los servicios se comunican entre sí o con otras aplicaciones a través de protocolos de comunicación estándar, como HTTP, REST, SOAP, entre otros.

Arquitecturas orientadas a servicios: en arquitecturas orientadas a servicios, las aplicaciones se dividen en componentes independientes y autónomos que se comunican para realizar tareas específicas.

Reutilización: Los servicios pueden ser reutilizados por múltiples aplicaciones. Esto promueve la eficiencia y evita la duplicación de esfuerzos.

Desacoplamiento: Los servicios permiten un acoplamiento flexible entre aplicaciones, lo que significa que los cambios en un servicio no necesariamente afectarán a otras partes del sistema.

Servicios web: Los servicios web son servicios que se exponen a través de la web y se acceden a través de protocolos web estándar como HTTP. Los formatos de intercambio de datos comunes en los servicios web incluyen XML y JSON.

Seguridad: Los servicios deben ser seguros para proteger los datos y la funcionalidad que proporcionan. Esto puede incluir autenticación, autorización y cifrado de datos.

APIs de servicio: Las APIs definen cómo los desarrolladores pueden interactuar con un servicio y acceder a su funcionalidad.

Servidor: es una computadora o sistema que proporciona recursos, servicios y funcionalidades a otras computadoras, conocidas como "clientes", a través de una red. Los servidores desempeñan un papel fundamental en el almacenamiento, procesamiento y distribución de datos y aplicaciones en entornos de red.

Soap: es un protocolo de comunicación utilizado en aplicaciones distribuidas y sistemas web para el intercambio de mensajes estructurados. Fue desarrollado como un estándar para permitir que diferentes sistemas, independientemente de sus plataformas y lenguajes de programación, puedan comunicarse entre sí de manera efectiva.

XML: es un lenguaje de marcado diseñado para almacenar y transmitir datos de manera legible tanto para humanos como para máquinas. Fue creado en la década de 1990 como un estándar por el World Wide Web Consortium. Aunque su popularidad ha disminuido un poco en favor de otros formatos como JSON en ciertas aplicaciones, XML todavía se utiliza ampliamente en diversas áreas, como la industria, la web semántica y la configuración de datos.

Web Services: son una tecnología ampliamente utilizada para permitir la comunicación y la interoperabilidad entre diferentes aplicaciones y sistemas a través de la red, especialmente en el contexto de la World Wide Web. Estos servicios permiten que aplicaciones escritas en diferentes lenguajes de programación y que se ejecutan en diferentes plataformas puedan intercambiar datos y realizar acciones sin necesidad de conocer los detalles internos de cada una.

Web: es un sistema de información en línea que permite acceder a documentos y recursos multimedia a través de la Internet. La World Wide Web, también conocida como "web", fue inventada por Tim Berners-Lee en 1989 y se ha convertido en una parte integral de la vida cotidiana de muchas personas en todo el mundo.

Balanceador: dispositivo o software que distribuye el tráfico de red entrante entre varios servidores o recursos para mejorar el rendimiento, la disponibilidad y la eficiencia del sistema. El objetivo principal de un balanceador es evitar la sobrecarga de un único servidor y asegurarse de que el tráfico se maneje de manera equitativa y eficiente en un entorno en el que haya múltiples recursos disponibles para atender las solicitudes.

Beneficios de usar balanceadores incluyen:

Mejora de la escalabilidad y el rendimiento de aplicaciones y servicios.

Distribución uniforme del tráfico para evitar la sobrecarga de servidores individuales.

Mayor disponibilidad y resistencia a fallos, ya que, si un servidor falla, el tráfico puede redirigirse a otros servidores en funcionamiento.

Capacidad de realizar mantenimiento en un servidor sin interrumpir el servicio, ya que el tráfico puede dirigirse a otros servidores mientras uno está fuera de línea.

Hay diferentes tipos de balanceadores, cada uno con sus propias características y usos. Algunos de los tipos más comunes incluyen: Balanceador de Carga Local, Balanceador de Carga Global, Balanceador de Carga de Aplicaciones, Balanceador de Carga de Servidores Web, Balanceador de Carga TCP/UDP, Balanceador de Carga basado en DNS, Balanceador de Carga de Contenedores

REST: es un estilo arquitectónico utilizado para diseñar servicios web y aplicaciones distribuidas en la World Wide Web. REST es un conjunto de principios y restricciones que fueron propuestos por Roy Fielding en su tesis doctoral en 2000, y se ha convertido en un enfoque popular para desarrollar sistemas web escalables y eficientes.

AWS: es una plataforma de servicios en la nube ofrecida por Amazon, que proporciona una amplia gama de servicios y herramientas para ayudar a las organizaciones y empresas a construir y gestionar aplicaciones y recursos en línea de manera eficiente. AWS es uno de los proveedores líderes en servicios en la nube y ha tenido un impacto significativo en la forma en que se desarrollan, implementan y operan aplicaciones y servicios en la actualidad.