

Act. 1.1 Definir los siguientes conceptos: Microservicios

API: significa "Interfaz de Programación de Aplicaciones" (por sus siglas en inglés, Application Programming Interface). Es un conjunto de reglas y protocolos que permiten a diferentes aplicaciones o sistemas comunicarse entre sí. Una API define los métodos y datos que una aplicación puede usar para solicitar y enviar información a otra aplicación, sin necesidad de conocer los detalles internos de cómo está construida esa otra aplicación.

Arquitectura: La arquitectura de software se refiere a la estructura fundamental y al diseño de alto nivel de un sistema o aplicación de software. Es como el plano maestro que guía la construcción de un edificio, pero en el contexto del desarrollo de software. La arquitectura de software define cómo los diferentes componentes del sistema interactúan entre sí, cómo se organizan y cómo se comunican para lograr los objetivos funcionales y no funcionales del software.

AWS: Amazon Web Services (AWS) es una plataforma de servicios en la nube ofrecida por Amazon. Se trata de un conjunto integral de servicios y soluciones de infraestructura informática, almacenamiento, análisis, machine learning, Internet de las cosas (IoT) y más, que permiten a las empresas y a los individuos acceder y utilizar recursos de tecnología de la información de manera flexible y escalable a través de Internet.

Back End: El término "backend" se refiere a la parte de un sistema informático o aplicación que se encarga del procesamiento y almacenamiento de datos, la lógica de negocio y la interacción con bases de datos y servicios externos. En el contexto del desarrollo de software, el backend es la columna vertebral que permite que una aplicación funcione y se comuniquen con los usuarios a través de la interfaz de usuario (frontend).

Bifurcación: En el contexto de control de versiones y desarrollo de software, una bifurcación (fork en inglés) se refiere a la creación de una copia independiente de un proyecto de software. Esto suele ocurrir cuando un grupo de desarrolladores toma una versión de un proyecto y comienza a desarrollarla por separado, a menudo para implementar nuevas características o abordar diferentes objetivos.

Escalabilidad: La escalabilidad se refiere a la capacidad de un sistema, aplicación o recurso para crecer y adaptarse eficientemente a un aumento en la demanda, sin perder rendimiento o funcionalidad. En otras palabras, un sistema escalable es capaz de manejar un aumento en la carga de trabajo o en el número de usuarios sin degradar su rendimiento o causar problemas en su funcionamiento.

Flexibilidad: La flexibilidad se refiere a la capacidad de adaptarse, ajustarse o cambiar en respuesta a diferentes situaciones, necesidades o condiciones. En diversos contextos, la flexibilidad implica la capacidad de modificar, extender o reconfigurar algo para cumplir con nuevos requisitos o circunstancias sin tener que hacer cambios drásticos o costosos.

Framework: Un framework (marco de trabajo o estructura en español) es una base preestablecida y estructurada que proporciona un conjunto de herramientas, bibliotecas y patrones de diseño que facilitan el desarrollo y la implementación de aplicaciones, sistemas o módulos de software. Los frameworks ofrecen una estructura básica que puede ser personalizada y extendida según las necesidades específicas del proyecto, lo que acelera el proceso de desarrollo y ayuda a mantener buenas prácticas de programación.

Front End: El frontend (o interfaz de usuario) es la parte visible y con la que interactúan los usuarios en una aplicación, sitio web o software. Es la capa del sistema que presenta la información y permite la interacción directa del usuario con la aplicación. En otras palabras, el frontend es la cara de un producto digital que los usuarios pueden ver, tocar y utilizar.

IaaS: "Infrastructure as a Service" (Infraestructura como Servicio) en español, es un modelo de entrega de servicios en la nube que proporciona a los usuarios recursos de infraestructura de TI a través de Internet. En lugar de comprar y mantener hardware y equipos físicos, los usuarios pueden alquilar recursos como servidores virtuales, almacenamiento, redes y otros componentes de infraestructura en la nube, de manera escalable y bajo demanda.

Microservicios: Los microservicios son un enfoque arquitectónico en el desarrollo de software en el que una aplicación grande se divide en pequeños componentes independientes, llamados microservicios, que se ejecutan de manera autónoma y se comunican entre sí a través de interfaces bien definidas. Cada microservicio se enfoca en una función o característica específica de la aplicación y se desarrolla, implementa y escala de manera independiente.

PaaS: "Platform as a Service" (Plataforma como Servicio) en español, es un modelo de entrega de servicios en la nube que proporciona a los desarrolladores una plataforma completa para crear, desarrollar, implementar y gestionar aplicaciones sin tener que preocuparse por la infraestructura subyacente. En lugar

de gestionar servidores, sistemas operativos y configuraciones de red, los desarrolladores pueden centrarse en escribir código y construir aplicaciones utilizando herramientas y servicios proporcionados por el proveedor de PaaS.

Servicio: En el contexto de la tecnología y la informática, un servicio se refiere a una función o característica que se proporciona a través de una aplicación, sistema o plataforma. Por ejemplo, un servicio web puede ofrecer acceso a datos o funcionalidades a través de una API.

Servidor: un servidor es un sistema informático dedicado a proporcionar servicios, recursos y funciones a otros dispositivos o programas llamados "clientes". Los servidores están diseñados para manejar solicitudes y proporcionar respuestas a través de una red, como Internet o una red local. Actúan como centros de almacenamiento y procesamiento de datos que facilitan la comunicación y el intercambio de información entre usuarios y aplicaciones.

SOAP: acrónimo de "Simple Object Access Protocol" (Protocolo Simple de Acceso a Objetos), es un protocolo de comunicación utilizado en entornos distribuidos para intercambiar información estructurada entre sistemas y aplicaciones a través de la red. Fue diseñado para permitir la comunicación entre componentes de software independientes y heterogéneos, a menudo utilizando el formato XML para representar los datos.

XML: "eXtensible Markup Language" (Lenguaje de Marcado Extensible) en español, es un lenguaje de marcas diseñado para describir datos y facilitar la estructuración, almacenamiento y transmisión de información entre sistemas de manera legible tanto para humanos como para máquinas. XML se utiliza ampliamente para representar y transmitir datos en una variedad de aplicaciones y contextos.

Web Services: Los servicios web son un conjunto de estándares y protocolos que permiten que aplicaciones y sistemas se comuniquen y compartan datos a través de Internet o una red. Estos servicios permiten que diferentes aplicaciones, independientemente de su lenguaje de programación o plataforma, interactúen entre sí y accedan a funcionalidades y datos a través de interfaces estandarizadas.

Web: "World Wide Web" (Red Mundial) es un sistema de información global que se encuentra en Internet y que permite a los usuarios acceder y compartir documentos, recursos multimedia y otros tipos de contenido a través de enlaces hipertexto. Fue creado por Tim Berners-Lee en la década de 1980 y se ha convertido en una parte integral de la vida cotidiana de las personas en todo el mundo.

Balanceador: Un balanceador, en el contexto de la tecnología y la informática, se refiere a un dispositivo o software que distribuye la carga de trabajo entre múltiples recursos, como servidores, redes o aplicaciones. El propósito principal de un balanceador es asegurarse de que los recursos se utilicen de manera eficiente y equitativa, evitando la sobrecarga en un recurso específico y mejorando el rendimiento y la disponibilidad del sistema en general.

REST: "Representational State Transfer" (Transferencia de Estado Representacional) en español, es un estilo de arquitectura de software que se utiliza para diseñar servicios web y sistemas distribuidos que son eficientes, escalables y fáciles de mantener. REST es un enfoque popular en el diseño de APIs (Interfaces de Programación de Aplicaciones) debido a su simplicidad y su énfasis en la interoperabilidad y la escalabilidad.