1. ¿Qué es el protocolo HTTP?

HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos. La estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web.

1. ¿cuáles son los métodos que implementa y cómo funciona?

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

El método DELETE borra un recurso en específico.

El método CONNECT establece un túnel hacia el servidor identificado por el recurso.

El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino.

El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

1. ¿cuáles son las respuestas, sus códigos y significados?

Respuestas informativas (100–199),

Respuestas satisfactorias (200–299),

Redirecciones (300–399),

Errores de los clientes (400–499),

Errores de los servidores (500–599).

1. ¿Qué es el formato JSON?

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. JSON es de fácil lectura y escritura para los usuarios.

1. ¿Qué es una API?

Una API, o interfaz de programación de aplicaciones, es un conjunto de reglas definidas que permiten que diferentes aplicaciones se comuniquen entre sí. Actúa como una capa intermedia que procesa las transferencias de datos entre sistemas.

1. ¿Cuántos tipos de API existen?

* **Las API abiertas** son interfaces de programación de aplicaciones de código abierto a las que puede acceder con el protocolo HTTP. También conocidas como API públicas, han definido puntos finales de API y formatos de solicitud y respuesta.
* **Las API de socios** conectan a socios comerciales estratégicos. Normalmente, los desarrolladores acceden a estas API en modo autoservicio a través de un [portal para desarrolladores de API](https://www.ibm.com/mx-es/products/api-connect/developer-portal) público. Aún así, deben completar un proceso de incorporación y obtener credenciales de inicio de sesión para acceder a las API de los socios.
* Las **API internas** permanecen ocultas para los usuarios externos. Estas API privadas no están disponibles para usuarios fuera de la empresa y están destinadas a mejorar la productividad y la comunicación entre diferentes equipos de desarrollo interno.
* **Las API compuestas** combinan múltiples API de datos o servicios. Permiten a los programadores acceder a varios puntos finales en una sola llamada. Las API compuestas son útiles en la arquitectura de microservicios donde realizar una sola tarea puede requerir información de varias fuentes

1. ¿Qué es una API REST?

Una API, o interfaz de programación de aplicaciones es un conjunto de reglas que definen la forma en que aplicaciones o dispositivos pueden conectarse y comunicarse entre sí. Una API REST es una API que se ajusta a los principios de diseño de REST, o el estilo de arquitectura de transferencia de estado representacional.

1. ¿Qué es un monolito?

Se refiere a una aplicación que ha sido desarrollada como un único y gran sistema, en lugar de estar dividida en componentes o servicios más pequeños y modularizados.

1. ¿Qué son los microservicios?

Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes.

1. ¿Qué es Docker y para qué se usa?

Docker es una plataforma de código abierto que permite a los desarrolladores crear, implementar, ejecutar, actualizar y gestionar contenedores, que son componentes estandarizados y ejecutables que combinan el código fuente de la aplicación con las bibliotecas y dependencias del sistema operativo (SO) necesarias para ejecutar ese código en cualquier entorno.

1. ¿Qué son los Kunernetes y cómo se usan?

Kubernetes, también conocido como "k8s" o "kube", es una plataforma de orquestación de contenedores para planificar y automatizar el despliegue, la gestión y el escalado de aplicaciones en contenedores.

1. ¿Qué es Terraform?

Terraform es una herramienta de codificación declarativa que permite a los desarrolladores utilizar un lenguaje de configuración denominado HCL (HashiCorp Configuration Language) para describir la nube de "estado final" deseada o la infraestructura local para ejecutar una aplicación.

1. ¿Qué es OpenShift?

OpenShift es una **plataforma de aplicaciones basada en Kubernetes**. El software se utiliza para construir entornos de desarrollo y aplicaciones descentralizadas y escalables.

1. ¿Qué es CI?

La integración continua es una práctica de [desarrollo de software](https://aws.amazon.com/es/devops/) mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas. La integración continua se refiere en su mayoría a la fase de creación o integración del proceso de publicación de software y conlleva un componente de automatización (p. ej., CI o servicio de versiones) y un componente cultural (p. ej., aprender a integrar con frecuencia). Los objetivos clave de la integración continua consisten en encontrar y arreglar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo que se tarda en validar y publicar nuevas actualizaciones de software.

1. ¿Qué es TDD?

TDD o Test-Driven Development (desarrollo dirigido por tests) es una práctica de programación que consiste en escribir primero las pruebas (generalmente unitarias), después escribir el código fuente que pase la prueba satisfactoriamente y, por último, refactorizar el código escrito.

1. ¿Qué es DEVOPS?

DevOps es un marco de trabajo y una filosofía en constante evolución que promueve un mejor desarrollo de aplicaciones en menos tiempo y la rápida publicación de nuevas o revisadas funciones de software o productos para los clientes. Con DevOps se promueve una comunicación continua más fluida, la colaboración, la integración, la visibilidad y la transparencia entre equipos de desarrollo de aplicaciones (Dev) y sus homólogos en operaciones tecnológicas (Ops).