**1. ¿Qué es el protocolo http?**

El protocolo **HTTP (Hypertext Transfer Protocol)** es un protocolo de comunicación utilizado en la World Wide Web para la transferencia de datos entre un cliente (por lo general, un navegador web) y un servidor web. HTTP es la base de la comunicación en la web y se utiliza para solicitar y entregar recursos como páginas web, imágenes, videos, etc.

**2. ¿Cuáles son los métodos que implementan y cómo funcionan?**

Los **métodos HTTP** son verbos que indican la acción que se debe realizar en un recurso identificado por una URL. Algunos de los métodos más comunes son:

* **GET**: Recupera información de un recurso.
* **POST**: Envía datos para que sean procesados por el servidor.
* **PUT**: Actualiza un recurso o crea uno nuevo si no existe.
* **DELETE**: Elimina un recurso.
* **PATCH**: Realiza una actualización parcial de un recurso.

**3. Respuestas 200, 300, 400, 500, ¿sus códigos y sus significados?**

Las **respuestas HTTP** se dividen en diferentes grupos según su código de estado:

* **200:** Éxito (por ejemplo, 200 OK).
* **300:** Redirección (por ejemplo, 301 Moved Permanently, 302 Found).
* **400:** Errores del cliente (por ejemplo, 400 Bad Request, 404 Not Found).
* **500:** Errores del servidor (por ejemplo, 500 Internal Server Error).

**4. ¿Qué es el formato JSON?**

**JSON (JavaScript Object Notation)** es un formato de intercambio de datos ligero y legible por humanos. Se utiliza para representar datos estructurados en la forma de objetos y arreglos, y es comúnmente utilizado en aplicaciones web y servicios para transmitir datos entre un servidor y un cliente.

**5. ¿Qué es una API?**

**Una API (Application Programming Interface)** es un conjunto de reglas y protocolos que permite a diferentes aplicaciones o componentes de software comunicarse entre sí. Las API definen cómo solicitar ciertas acciones o acceder a datos en un sistema o servicio.

**6. ¿Cuántos tipos de API existen?**

Hay varios tipos de **API**, pero los principales son:

* **API web:** Utilizada para la comunicación a través de Internet, a menudo utilizando HTTP o HTTPS.
* **API de biblioteca**: Proporciona funciones y procedimientos para interactuar con una biblioteca de software.
* **API de sistema operativo:** Permite a las aplicaciones interactuar con el sistema operativo subyacente.
* **API de base de datos:** Proporciona acceso a una base de datos y la manipulación de datos.

**7. ¿Qué es una API REST?**

Una **API REST (Representational State Transfer)** es un estilo arquitectónico para el diseño de sistemas de software que utilizan HTTP y sus verbos **(GET, POST, PUT, DELETE)** para acceder y manipular recursos a través de URLs. Se basa en principios de simplicidad y utiliza el estado de la aplicación para representar el recurso en lugar de mantener un estado en el servidor.

**8. ¿Qué es un monolito?**

Un **monolito** es una arquitectura de software en la que todas las funcionalidades de una aplicación se desarrollan y despliegan como un solo bloque de código. En un monolito, todos los componentes de la aplicación están interconectados y se ejecutan en un solo proceso.

**9. ¿Qué son los microservicios?**

Los **microservicios** son una arquitectura de software en la que una aplicación se divide en pequeños servicios independientes que pueden ser desarrollados, desplegados y escalados de manera individual. Cada microservicio se enfoca en una función específica de la aplicación y se comunica con otros a través de API.

**10. ¿Qué es Docker y para qué se usa?**

**Docker** es una plataforma de contenedorización que permite empaquetar aplicaciones y sus dependencias en contenedores, que son entornos de ejecución aislados y portátiles. Se utiliza para facilitar el despliegue y la gestión de aplicaciones en entornos de desarrollo y producción.

**11. ¿Que son los Kubernetes y cómo se usan?**

**Kubernetes** es un sistema de orquestación de contenedores de código abierto que permite automatizar la implementación, escalabilidad y administración de aplicaciones en contenedores. Ayuda a gestionar la alta disponibilidad y la escalabilidad de aplicaciones en entornos de contenedor.

**12. ¿Qué es Terraform?**

**Terraform** es una herramienta de infraestructura como código (IaC) que permite definir y gestionar la infraestructura de manera declarativa. Los usuarios describen su infraestructura en archivos de configuración y Terraform se encarga de crear y gestionar los recursos en la nube de acuerdo con esas especificaciones.

**13. ¿Qué es OpenShift?**

**OpenShift** es una plataforma de contenedores de código abierto que se basa en Kubernetes y que facilita la implementación y administración de aplicaciones en contenedores en entornos de nube híbrida o local.

**14. ¿Qué es CI?**

**CI (Integración Continua)** es una práctica de desarrollo de software en la que los cambios de código se integran regularmente en un repositorio compartido. Cada integración desencadena una serie de pruebas automáticas para asegurar la calidad del código.

**15. ¿Qué es TDD?**

**TDD (Desarrollo Guiado por Pruebas)** es una metodología de desarrollo de software en la que se escriben pruebas automatizadas antes de escribir el código de la funcionalidad. El desarrollo se guía por el objetivo de que todas las pruebas pasen correctamente.

**16. ¿Qué es DEVOPS?**

**DevOps** es una cultura y conjunto de prácticas que busca la colaboración entre equipos de desarrollo (**Dev**) y operaciones (**Ops**) para automatizar y agilizar la entrega de software, desde la planificación hasta la implementación y la operación, con un enfoque en la mejora continua.