

Backend · Docker · CI/CD

1 ¿Cómo se relacionan frontend, backend y base de datos dentro de una arquitectura web moderna?

Explica el rol de cada uno y cómo se comunican entre sí.

El front muestra la interfaz y envía peticiones al back y el back consulta la base de dato para luego mostrarle los datos al front

2 ¿Qué ventajas ofrece una API REST frente a otros modelos de comunicación entre sistemas?

Menciona al menos dos ventajas desde el punto de vista de escalabilidad o mantenimiento.

Ofrece gran escalabilidad por su arquitectura sin estado que facilita distribuir la carga, y simplifica el mantenimiento al desacoplar totalmente cliente y servidor permitiendo que evolucionen de forma independiente.

3 ¿Qué implica diseñar un backend desacoplado del frontend?

Explica por qué es importante en proyectos grandes o a largo plazo.

Porque es escalable y mejora la optimización o mantenimiento ya que está dividido

4 ¿Qué diferencias conceptuales existen entre contenedores Docker y máquinas virtuales?

Indica al menos dos diferencias claras a nivel de recursos o arquitectura.

Las máquinas virtuales virtualiza hardware con su sistema operativo y el docker utiliza un núcleo. Los contenedores son más rápidos ya que no tienen que virtualizar el hardware.

5 ¿Qué significa que una aplicación sea *portable* y cómo ayuda Docker a conseguirlo?

Explica por qué esto es clave en entornos profesionales.

Que una aplicación funcione igual en cualquier entorno al empaquetar Docker todas sus dependencias, asegura transiciones fiables entre desarrollo y producción, evitando fallos inesperados por diferencias de configuración.

6 ¿Qué es la integración continua (CI) y qué problemas evita en un equipo de desarrollo?

Relaciona la respuesta con errores de código o coordinación.

Fusionar y validar automáticamente el código en un repositorio compartido varias veces.

7) ¿Qué es el despliegue continuo (CD) y qué requisitos debería cumplir un proyecto para poder usarlo con seguridad?

Habla de estabilidad, tests o control de versiones.

El despliegue continuo es que la aplicación se actualice automáticamente para el usuario final tras cada mejora; para hacerlo sin riesgo, necesitas pruebas automáticas muy estrictas que actúen como "frenos de emergencia" si algo falla.

8) ¿Por qué la automatización es un elemento clave en DevOps?

Explica cómo afecta al tiempo, a la calidad del software y al equipo.

Porque elimina las tareas manuales repetitivas donde los humanos suelen equivocarse, permitiendo entregar mejoras mucho más rápido y garantizando que el proceso siempre se haga igual.

9) ¿Qué diferencias existen entre un error detectado en fase de desarrollo y uno detectado en producción?

Explica las consecuencias técnicas y organizativas.

Un error en desarrollo es barato y fácil de corregir porque nadie lo ve; en producción es crítico porque afecta a usuarios reales, costando mucho dinero y dañando la imagen de la empresa.

10) ¿Por qué es importante separar la configuración del código en un backend?

Da ejemplos conceptuales (entornos, seguridad, mantenimiento).

Permite que el mismo código funcione en distintos lugares, cambiando solo variables externas.

Evita que datos sensibles, como contraseñas de bases de datos o claves privadas, queden expuestos en el historial del código fuente.