

Prueba Técnica - Desarrollador Fullstack Junior/Tecnólogo
Alejandro Aguirre Ruiz

1. APIs REST y Arquitectura

- a. Explique las diferencias entre los métodos HTTP GET, POST, PUT, PATCH y DELETE. Proporcione un ejemplo práctico de cuándo usar cada uno.

Método	Función	Ejemplo
GET	Obtener datos	GET /empleados -> Devuelve lista de todos los empleados
POST	Crear un nuevo recurso	POST /empleados -> agrega el empleado a la BD
PUT	Actualiza el recurso completo	PUT /empleado/123 -> actualiza todo el empleado nombre, correo, fecha, etc..
PATCH	Actualiza un dato en específico	PATCH /empleado/123 con correo 'prueba@gmail.com' -> cambia solo el correo
DELETE	Borra un registro	DELETE /empleado/123 -> elimina el empleado 123

- b. ¿Qué es un código de estado HTTP? Explique el significado de los códigos: 200, 201, 400, 401, 403, 404, 500.

Código	Significado
200 Ok	Todo salió bien
201 Created	Se creó
400 Bad Request	Petición mal armada
401 Unauthorized	Credenciales invalidas, necesario iniciar sesión
403 Forbidden	Credenciales validas, pero sin permisos
404 Not Found	No existe lo que se pide
500 Internal Server Error	Error de servidor

- c. Describa qué es RESTful y mencione al menos 4 principios fundamentales de una API REST

Una API RESTful es simplemente una forma ordenada de hablar con un servidor usando lo que HTTP ya trae:

- URLs para señalar cosas (recursos).
- Verbos (GET, POST, etc.) para decir qué quieres hacer.
- Códigos de estado (200, 404...) para saber si salió bien o mal.

Principios básicos

- Cliente – servidor: Separa la interfaz del usuario con la lógica del negocio y almacenamiento de datos
- Sin estado (Stateless): Cada solicitud de cliente al servidor debe contener toda la información necesaria
- Caché: Datos que se puedan ser obtenidos y almacenados como cache

- Códigos HTTP bien usados
- d. ¿Qué es CORS y por qué es importante en el desarrollo web? ¿Cómo se configura?

Es una regla de seguridad del navegador, solo deja que la pagina pida datos al mismo dominio, protocolo y puerto.

Es importante porque protege al usuario evitando que cualquier pagina desconocida pueda leer datos senciales

La configuración básica se tiene desde las cabeceras *access-control-allow-origin*, *access-control-allow-methods*, *access-control-allow-headers*, *access-control-allow-credentials*

2. Microservicios y Arquitectura en la Nube

- a. Defina qué son los microservicios y mencione 3 ventajas y 2 desventajas frente a una arquitectura monolítica.

Una arquitectura de microservicios divide una aplicación grande en pequeños servicios independientes

- Se centran en una única función
- Se comunican entre si
- Pueden desplegarse y escalarse por separado

Ventajas

- Se pueden desplegar independientes
- Si un servicio recibe mucho tráfico, le das más instancias solo a ese servicio y no a todo el sistema
- Puede ser multi código, puedes tener una función en Python, otra en node o .net, etc

Desventajas

- Las llamadas entre servicios añaden latencia y posibles fallos de comunicación por lo que se sobre cargaría la red
- Puede demorar los procesos de despliegues, ya que son por separado cada uno y cada uno tiene que pasar por varios procesos

- b. Explique qué es un contenedor Docker y cómo difiere de una máquina virtual.

Un contenedor Docker es un paquete ligero con la app más las dependencias que se ejecuta usando el mismo kernel del host, arranca en segundos por que no carga un sistema operativo completo y el mismo container corre igual en un pc que en un servidor o en la nube.

Por lo contrario la maquina virtual es una computadora dentro de otra, simula todo ram, disco, bios y sus sistema operativo, arranca mas lento y consume más recurso y cada VW actpua como si fuera un servidor físico indenpendiente

- c. Mencione 3 servicios de computación en la nube (AWS, Azure, GCP) y explique brevemente para qué se usa cada uno.

AWS	Lambda	Ejecuta funciones sin tener que levantar servidores, solo subes el script y se corre cuando lo llamas
Azure	Blob Storage	“Disco duro” infinito en la nube para guardar archivos, imágenes, PDF’s, video, bakups, etc...
Google Cloud	BigQuery	Base de datos analítica súper rápida para procesar terabytes de datos con SQL

- d. ¿Qué es el despliegue continuo (CD) y cómo se diferencia de la integración continua (CI)?

Integración continua: Cada vez que alguien sube código, el sistema lo complica, corre pruebas y genera un artefacto.

Despliegue continuo: Toma el artefacto ya probado y lo lleva automáticamente a un entorno productivo

3. Metodologías Ágiles – SCRUM

- a. Defina los roles principales en SCRUM y sus responsabilidades.

Rol	¿Quién es?	Responsabilidades
Product owner (PO)	La persona que representa al negocio y a los usuarios; traduce necesidades en historias de valor.	<ul style="list-style-type: none"> Decide que se construye y en qué orden Mantiene y prioriza el Product backlog Habla con clientes o usuarios y pasa la visión al equipo

Scrum Master (SM)	Coach del equipo y guardián del proceso; se asegura de que todos trabajen sin bloqueos y sigan las reglas de Scrum.	<ul style="list-style-type: none"> • Asegura que se siga la metodología Scrum • Quita bloqueos al equipo • Facilita las reuniones (Daily, Review, Retrospective)
Development Team	El grupo que hace el trabajo dev, qa, ux, etc...	<ul style="list-style-type: none"> • Concreta el como: diseña, codifica, prueba y entrega • Se autoorganiza y estima esfuerzos

- b. Explique qué son los Sprints, Sprint Planning, Daily Standup, Sprint Review y Sprint Retrospective.

Evento	¿Qué es?
Sprint	Un bloque de tiempo fijo (1-4 semanas) donde el equipo se compromete a entregar
Planing	Reunión que abre el sprint
Daily	Check rápido cada día.
Review	Demo al final del Sprint.
Retrospective	Reunión interna justo después del Review.

- c. ¿Qué es un Product Backlog y un Sprint Backlog?

El **Product Backlog** es la lista completa de “todo lo deseado”; el **Sprint Backlog** es la parte de esa lista que el equipo va a convertir en producto funcionando durante el Sprint en curso.

4. Tecnologías Específicas

- a. ¿Qué es JBoss y para qué se utiliza en el desarrollo de aplicaciones empresariales?

JBoss es donde vive tu aplicación Java empresarial, dándote los servicios de infraestructura (transacciones, seguridad, clustering) que sería costoso re-inventar a mano.

- b. Explique brevemente qué es X-Road y en qué contexto se implementa

X-Road es una plataforma *open source* que funciona como “carretera” de datos entre sistemas: estandariza, cifra y firma los mensajes para que distintas organizaciones (ministerios, bancos, hospitales, empresas) puedan intercambiar información por Internet de forma segura y verificable, sin necesidad de crear bases de datos centrales.

Se implementa cuando un gobierno o un ecosistema que combine sector público y privado necesita un canal estandarizado, seguro y auditado para que múltiples sistemas intercambien datos manteniendo soberanía, trazabilidad y escalabilidad.