

## Prueba técnica Backend NodeJS

### Preguntas de Selección Múltiple:

1. ¿Qué es un microservicio?
  - a) Una aplicación monolítica.
  - b) Un componente independiente y autónomo de software.
  - c) Un lenguaje de programación.
  - d) Un tipo de base de datos.
  
2. ¿Qué es la arquitectura orientada a eventos?
  - a) Una arquitectura que no utiliza eventos.
  - b) Una arquitectura que se basa en la comunicación síncrona.
  - c) Una arquitectura que utiliza eventos para comunicar componentes.
  - d) Una arquitectura que no se utiliza en el desarrollo de software.
  
3. ¿Qué es Docker?
  - a) Un sistema operativo.
  - b) Una herramienta de virtualización.
  - c) Una plataforma de contenedores.
  - d) Un lenguaje de programación.
  
4. ¿Cuál de las siguientes tecnologías se utiliza comúnmente para orquestar contenedores en un clúster?
  - a) Docker Compose.
  - b) Kubernetes.
  - c) Apache Kafka.
  - d) Redis.
  
5. ¿Qué es un patrón de diseño Singleton?
  - a) Un patrón de diseño que garantiza la creación de una única instancia de una clase.
  - b) Un patrón de diseño para la comunicación entre microservicios.
  - c) Un patrón de diseño para implementar la autenticación.
  - d) Un patrón de diseño para la gestión de eventos.
  
6. ¿Qué significa REST en el contexto de servicios web?
  - a) Representational State Transfer.
  - b) Remote Service Transfer.

- c) Request State Transfer.
- d) Resource State Transmission.

7. ¿Cuál de las siguientes afirmaciones es verdadera sobre HTTP?

- a) HTTP es un protocolo de transferencia de archivos.
- b) HTTP es un protocolo de transferencia de correo electrónico.
- c) HTTP es un protocolo de aplicación basado en TCP/IP utilizado para la transferencia de datos en la web.
- d) HTTP es un protocolo de seguridad.

8. ¿Qué es un mensaje en el contexto de la programación orientada a eventos?

- a) Un error en el código.
- b) Una unidad de código fuente.
- c) Un evento que indica que algo ha sucedido.
- d) Una variable global.

9. ¿Cuál es el propósito de un balanceador de carga en una arquitectura de microservicios?

- a) Garantizar que todos los microservicios tengan la misma carga de trabajo.
- b) Distribuir el tráfico de manera uniforme entre los microservicios para mejorar la escalabilidad y la disponibilidad.
- c) Aumentar la latencia de los microservicios.
- d) Desplegar microservicios en diferentes regiones geográficas.

10. ¿Cuál es el propósito de un sistema de mensajería en una arquitectura orientada a eventos?

- a) Gestionar la autenticación y autorización.
- b) Facilitar la comunicación asíncrona entre componentes.
- c) Almacenar registros de eventos para auditoría.
- d) Ejecutar código en microservicios.

11. ¿Qué es DevOps?

- a) Una metodología de desarrollo de software.
- b) Un lenguaje de programación.
- c) Una herramienta de automatización de pruebas.
- d) Una cultura y conjunto de prácticas que integran desarrollo y operaciones.

12. ¿Cuál es el propósito de un manejador de eventos en un sistema orientado a eventos?

- a) Producir eventos.
- b) Consumir eventos y tomar acciones en respuesta a ellos.
- c) Almacenar eventos en una base de datos.

d) Definir eventos en un sistema.

13. ¿Qué es un contenedor en el contexto de Docker?

- a) Una máquina virtual.
- b) Un archivo ZIP.
- c) Una instancia de una imagen de Docker que se ejecuta en un sistema operativo.
- d) Una aplicación web.

14. ¿Qué es un webhook?

- a) Una técnica para realizar pruebas unitarias.
- b) Una URL que permite recibir notificaciones o eventos desde una fuente externa.
- c) Una base de datos en memoria.
- d) Un servicio de autenticación.

15. ¿Cuál es el propósito principal de las pruebas de integración?

- a) Identificar errores en un componente individual.
- b) Validar la interacción entre diferentes componentes del sistema.
- c) Probar la funcionalidad de una aplicación en producción.
- d) Evaluar el rendimiento de un servidor web.

16. ¿Qué es un servicio RESTful?

- a) Un servicio web que utiliza SOAP como protocolo de comunicación.
- b) Un servicio web que sigue los principios y restricciones de REST.
- c) Un servicio web que no admite solicitudes HTTP GET.
- d) Un servicio web que solo funciona en entornos locales.

17. ¿Cuál es la ventaja de utilizar un sistema de control de versiones como Git en el desarrollo de software?

- a) Facilita la programación concurrente.
- b) Proporciona seguridad en línea.
- c) Elimina la necesidad de escribir código.
- d) Reduce la velocidad de desarrollo.

18. ¿Cuál es el propósito de un diagrama de secuencia en UML (Unified Modeling Language)?

- a) Representar la estructura de una base de datos.
- b) Mostrar la interacción entre objetos en un sistema.

- c) Describir la arquitectura de hardware de un servidor.
- d) Documentar los requisitos de un proyecto.

19. ¿Cuál es el objetivo de la programación orientada a objetos (POO)?

- a) Facilitar la comunicación entre microservicios.
- b) Organizar el código en funciones.
- c) Modelar el mundo real mediante la creación de clases y objetos.
- d) Utilizar solo funciones en lugar de objetos.

20. ¿Cuál es la diferencia entre una base de datos SQL y una base de datos NoSQL?

- a) Una base de datos SQL utiliza una estructura de datos fija, mientras que una base de datos NoSQL es flexible en su estructura.
- b) Una base de datos SQL es más rápida que una base de datos NoSQL.
- c) Una base de datos SQL no admite consultas, mientras que una base de datos NoSQL sí lo hace.
- d) Una base de datos SQL solo almacena datos numéricos.

## Ejercicio Práctico

Supongamos que estás desarrollando un servicio de entrega de comida en línea. Diseña una arquitectura de microservicios para este sistema y describe brevemente cómo se comunicarían estos microservicios a través de eventos.

Para un servicio de entrega de comida en línea, diseñar una arquitectura de microservicios de la siguiente manera:

1. Servicio de Pedidos: Este microservicio sería responsable de recibir y gestionar los pedidos de comida de los clientes. Cuando un cliente realiza un pedido, este servicio crea un evento "PedidoCreado" que contiene los detalles del pedido.
2. Servicio de Restaurantes: Este microservicio gestionaría la información de los restaurantes asociados. Cuando un restaurante actualiza su menú o estado de disponibilidad, emite un evento "ActualizaciónDeRestaurante" que contiene la información relevante.
3. Servicio de Entrega: Este microservicio se encargaría de asignar entregadores a los pedidos y rastrear el estado de entrega. Cuando se asigna un entregador a un pedido, emite un evento "EntregadorAsignado" con los detalles pertinentes.

4. Servicio de Pagos: Este microservicio maneja las transacciones de pago. Cuando se completa un pago exitoso, emite un evento "PagoCompletado" que incluye la información de la transacción.

5. Servicio de Notificaciones: Este microservicio se encarga de enviar notificaciones a los clientes sobre el estado de sus pedidos. Escucha los eventos generados por otros microservicios y envía notificaciones push o por correo electrónico según sea necesario.

Estos microservicios se comunicarían entre sí a través de eventos utilizando un sistema de mensajería como Apache Kafka o RabbitMQ. Cada vez que un evento significativo ocurre en un microservicio, se emite un evento que otros microservicios pueden escuchar y reaccionar en consecuencia. Esto permite una arquitectura altamente escalable y desacoplada para el sistema de entrega de comida en línea.

Tecnologías a utilizar:

- NodeJS
- PostgreSQL
- Kafka o RabbitMQ
- NestJS – Opcional
- Docker

Entregar la URL del Repositorio Publico (Github – Gitlab – Bitbucket) y en el README.md especificar la solución, Documentación de los Servicios y como correr el proyecto