

## Solución prueba para postulación a cargo desarrollador

**Nombre:**

Brayan David Pulido Romero

**Cedula:**

1007297022

**Fecha:**

16/03/2025

## Instrucciones

Te damos la bienvenida a esta prueba técnica que nos permitirá evaluar tus habilidades para el rol de Desarrollador Senior. A continuación, te proporcionamos las instrucciones para completarla:

**Duración:** Tienes un plazo de 24 horas a partir de la recepción de este correo para completar y enviar la prueba.

**Formato:** La prueba contiene preguntas de selección múltiple que deberás responder con la mayor precisión posible.

**Envío:** Una vez completada la prueba, por favor envíala como un archivo adjunto en formato PDF al correo electrónico

Asegúrate de verificar tus respuestas antes de enviar el archivo, ya que no se aceptarán envíos posteriores al plazo establecido.

¡Éxito en tu prueba!

## Conocimientos en Node

LOGÍSTICA | DIGITAL | BPO

Indica cuál o cuáles de las siguientes opciones describen correctamente el concepto de Node y sus ventajas:

- a) Node es un framework de JavaScript para construir aplicaciones web.
- b) Node es un entorno de tiempo de ejecución de JavaScript que permite ejecutar código JavaScript fuera del navegador web.
- c) Node es un lenguaje de programación que permite trabajar con bases de datos.
- d) Las principales ventajas de Node son la escalabilidad, la eficiencia en el manejo de peticiones y la posibilidad de trabajar con múltiples conexiones en tiempo real.
- e) Todas las anteriores.

Indica cuál o cuáles de las siguientes opciones describen correctamente la diferencia entre concurrencia y paralelismo:

- a) La concurrencia se refiere a la capacidad de un sistema para ejecutar varias tareas al mismo tiempo, mientras que el paralelismo se refiere a la ejecución simultánea de varias tareas en diferentes procesadores o núcleos de un sistema.
- b) La concurrencia y el paralelismo son términos sinónimos que se refieren a la capacidad de un sistema para ejecutar múltiples tareas de manera eficiente.
- c) La concurrencia se refiere a la capacidad de un sistema para ejecutar varias tareas en secuencia, mientras que el paralelismo se refiere a la ejecución simultánea de varias tareas en diferentes hilos de ejecución dentro de un mismo proceso.
- d) La concurrencia se refiere a la capacidad de un sistema para ejecutar varias tareas al mismo tiempo en diferentes procesadores o núcleos, mientras que el paralelismo se refiere a la ejecución secuencial de varias tareas en un solo hilo de ejecución.

Para qué se usa `async/await` y qué problemas resuelve:

- a) `async/await` es utilizado para permitir la ejecución de código asíncrono de manera secuencial y resuelve el problema de tener que anidar callbacks para manejar los resultados de operaciones asíncronas.
- b) `async/await` es utilizado para permitir la ejecución de código síncrono en ambientes asíncronos y resuelve el problema de tener que manejar el estado de varias operaciones asíncronas en paralelo.
- c) `async/await` es utilizado para permitir la ejecución de código asíncrono de manera paralela y resuelve el problema de tener que esperar a que una operación asíncrona termine para poder continuar con otra.
- d) `async/await` es utilizado para permitir la ejecución de código asíncrono de manera secuencial y resuelve el problema de tener que manejar el estado de varias operaciones asíncronas en paralelo.

**Indica cuál o cuáles de las siguientes opciones describen correctamente qué es NPM y package.json:**

- a) NPM es un lenguaje de programación utilizado en el desarrollo web, mientras que package.json es un paquete de software utilizado para el manejo de bases de datos.
- b) **NPM es un administrador de paquetes para el lenguaje de programación JavaScript, mientras que package.json es un archivo de configuración utilizado para definir las dependencias de un proyecto de Node.js.**
- c) NPM es un entorno de ejecución para el lenguaje de programación Node.js, mientras que package.json es un archivo que contiene la documentación del código fuente de un proyecto.
- d) NPM es un módulo de Node.js que permite crear aplicaciones web, mientras que package.json es un archivo utilizado para el manejo de versiones de un proyecto.

**Indica cuál o cuáles de las siguientes opciones describen correctamente qué son los Buffer y para qué se utilizan:**

- a) Los Buffer son una clase en JavaScript que se utiliza para almacenar datos binarios y se utilizan para manipular archivos.
- b) **Los Buffer son una clase en Node.js que se utiliza para almacenar datos binarios y se utilizan para manipular archivos, realizar operaciones de red y criptografía.**
- c) Los Buffer son una clase en React que se utiliza para almacenar datos binarios y se utilizan para manipular archivos.
- d) Los Buffer son una clase en TypeScript que se utiliza para almacenar datos binarios y se utilizan para manipular archivos.

**Qué es un stream y qué es autenticación y autorización:**

- a) Un stream es un conjunto de datos almacenados en una base de datos y la autenticación y autorización son técnicas utilizadas para la transmisión de datos.
- b) **Un stream es un flujo de datos en tiempo real que se envía y recibe en trozos pequeños y la autenticación y autorización son técnicas utilizadas para controlar el acceso a los recursos de una aplicación.**
- c) Un stream es un conjunto de datos almacenados en un archivo y la autenticación y autorización son técnicas utilizadas para la manipulación de archivos.
- d) Un stream es una técnica utilizada para la transmisión de datos y la autenticación y autorización son técnicas utilizadas para la gestión de bases de datos.

## Conocimientos de arquitectura de software

¿Cuál es un patrón de diseño arquitectónico utilizado para separar la interfaz de usuario de la lógica de negocio en una aplicación web?

- a) **Modelo-Vista-Controlador (MVC)**
- b) Modelo-Vista-Presentador (MVP)
- c) Modelo-Vista-ViewModel (MVVM)
- d) Component-Based Architecture (CBA)

¿Cuál de las siguientes opciones es una ventaja de la arquitectura de microservicios en comparación con la arquitectura monolítica?

- a) Mayor complejidad en el mantenimiento del código.
- b) Mayor acoplamiento entre los componentes del sistema.
- c) **Mayor escalabilidad y flexibilidad.**
- d) Mayor simplicidad en el diseño de la arquitectura.

¿Cuál de las siguientes afirmaciones es cierta sobre la arquitectura orientada a servicios (SOA)?

- a) SOA es un patrón arquitectónico que se centra en la integración de bases de datos.
- b) SOA es una metodología de desarrollo de software centrada en la seguridad.
- c) **SOA es una arquitectura en la que los componentes del sistema se comunican a través de mensajes.**
- d) SOA es un marco de trabajo para la gestión de proyectos de software.

¿Qué es la escalabilidad horizontal en el contexto de la arquitectura de software?

- a) La capacidad de un sistema para manejar grandes cantidades de datos.
- b) La capacidad de un sistema para adaptarse a diferentes entornos de ejecución.
- c) La capacidad de un sistema para procesar múltiples tareas simultáneamente.
- d) **La capacidad de un sistema para manejar un aumento en la carga de trabajo distribuyendo la carga en varios servidores.**

**¿Qué patrón de diseño se utiliza para definir la relación entre los objetos de una clase, de manera que un cambio en el estado de un objeto afecte a los objetos dependientes?**

- a) **Observador (Observer)**
- b) Singleton
- c) Proxy
- d) Decorador

**¿Cuál de las siguientes opciones es una característica de la arquitectura de software basada en microservicios?**

- a) Monolito de código fuente único.
- b) Fuerte acoplamiento entre los componentes del sistema.
- c) Comunicación sincrónica entre los microservicios.
- d) **Independencia en el despliegue y la escala de los microservicios.**

**¿Cuál de las siguientes afirmaciones es correcta sobre la arquitectura sin servidor (serverless)?**

- a) **En la arquitectura sin servidor, los desarrolladores se preocupan más por construir la aplicación y menos en donde esta será desplegada**
- b) La arquitectura sin servidor solo se utiliza en aplicaciones web.
- c) En la arquitectura sin servidor, los desarrolladores no tienen control sobre la infraestructura subyacente.
- d) La arquitectura sin servidor es más costosa que otras arquitecturas tradicionales.

**¿Cuál es un patrón arquitectónico que se utiliza para mejorar el rendimiento y la velocidad de respuesta de una aplicación web al almacenar en caché el contenido estático en servidores distribuidos cercanos a los usuarios finales?**

- a) **CDN (Content Delivery Network)**
- b) Proxy reverso
- c) Load balancer
- d) Escalado vertical



**¿Cuál de los siguientes patrones arquitectónicos se utiliza para garantizar que un microservicio pueda manejar peticiones simultáneas y distribuir la carga de trabajo de manera equitativa?**

- a) Servidor de aplicaciones
- b) Proxy inverso
- c) Escalado vertical
- d) **Load balancing**

**¿Cuál de los siguientes patrones arquitectónicos se utiliza para gestionar la persistencia de datos en un sistema de microservicios, donde cada microservicio tiene su propia base de datos?**

- a) Base de datos compartida
- b) Gateway de bases de datos
- c) **Patrón CQRS (Command Query Responsibility Segregation)**
- d) Patrón repositorio

**Sobre la tolerancia a fallos en la arquitectura de microservicios?**

- a) Los microservicios no necesitan ser tolerantes a fallos, ya que son pequeños y autónomos.
- b) Los microservicios dependen completamente de la tolerancia a fallos de la infraestructura subyacente.
- c) **Cada microservicio debe ser diseñado para ser tolerante a fallos y manejar errores de forma independiente.**
- d) La tolerancia a fallos en la arquitectura de microservicios es responsabilidad exclusiva del equipo de operaciones.

**¿Qué es el descubrimiento de servicios en el contexto de la arquitectura de microservicios?**

- a) La capacidad de un microservicio para descubrir automáticamente los servicios que se encuentran en su mismo servidor.
- b) La capacidad de un microservicio para descubrir los servicios disponibles en otros servidores del sistema.
- c) **La capacidad de un microservicio para registrar su disponibilidad en un registro centralizado.**
- d) La capacidad de un microservicio para compartir información de configuración con otros microservicios.

## Caso Parctico

Diseñe una aplicación con arquitectura orientada a micro servicios que permita gestionar la información del usuario y sus tareas:

Los artefactos a construir serán los siguientes:

1. Un micro servicio que permita el registro del usuario y posteriormente la autenticación (básica) del mismo
2. Un micro servicio que provee las funcionalidades de crear, actualizar y eliminar las tareas de un usuario
3. Desarrolle en el front de preferencia (React deseable) las interfaces de usuario que permita la captura y la visualización de datos
4. Puede trabajar con el gestor de base de datos de su preferencia

### Tecnologías a utilizar que debe tener instalado:

- Una versión reciente de node
- React deseable
- IDE de desarrollo de su preferencia
- RDBMS de su preferencia o en su defecto MongoDB

Una vez terminada esta prueba, subir sus artefactos a un repositorio de GIT de su preferencia y adjunte las instrucciones que considere necesarias para poder ejecutar los artefactos de forma exitosa.