

## Evaluación Técnica para Desarrolladores

Esta evaluación tiene como objetivo conocer tus habilidades y conocimientos, así como tu capacidad para resolver problemas y aplicar conceptos clave en situaciones prácticas.

Por favor, responde a las siguientes preguntas utilizando tus propias palabras y proporcionando detalles relevantes. No dudes en demostrar tu razonamiento y enfoque para abordar los desafíos planteados. ¡Buena suerte!

1. ¿Cuál es la diferencia entre arquitectura monolítica y microservicios?
2. Menciona algunos patrones de diseño comunes en la arquitectura de software.
3. ¿Qué significa el principio SOLID y cómo se aplica en el desarrollo de software?
4. Explica la diferencia entre acoplamiento y cohesión en términos de diseño de software.
5. ¿Qué es la prueba unitaria y por qué es importante?
6. ¿Cómo garantizas que tu sitio web sea compatible con dispositivos móviles?
7. ¿Cómo optimizarías el rendimiento de una página web?
8. ¿Cuáles son algunas prácticas recomendadas para asegurar una aplicación web?
9. ¿Qué es un ataque de inyección SQL y cómo se puede prevenir?
10. ¿Cuál es la diferencia entre una base de datos SQL y NoSQL?
11. ¿Qué es la normalización en bases de datos relacionales?
12. Describe la diferencia entre INNER JOIN y LEFT JOIN en SQL.
13. Explica el propósito de los comandos git clone, git pull y git push.
14. ¿Qué es una fusión (merge) y cómo se resuelven los conflictos de fusión en Git?
15. Explica la diferencia entre una rama (branch) y una etiqueta (tag) en Git.

### Contexto:

Imagina que estás trabajando en el desarrollo de un sistema de control de gastos para una empresa. El objetivo principal es permitir a los empleados registrar sus gastos de manera eficiente, incluyendo detalles como el monto, la categoría del gasto, el RUC del comercio, y la posibilidad de adjuntar una foto del comprobante del gasto. El sistema debe tener un FrontEnd, un BackEnd y utilizar una arquitectura de microservicios para garantizar la escalabilidad y la independencia de los servicios.

### Requisitos Funcionales:

#### 1. Autenticación y Autorización:

- Los usuarios deben poder registrarse e iniciar sesión.
- Solo los usuarios autenticados pueden registrar gastos.

#### 2. Registro de Gastos:

- Los usuarios deben poder registrar un gasto especificando el monto, la categoría, el RUC del comercio y adjuntando una foto del comprobante.
- La categoría del gasto puede ser: Alimentación, Transporte, Alojamiento, Otros.

#### 3. Visualización de Gastos:

- Los usuarios deben poder ver una lista de sus gastos anteriores, ordenados por fecha.
- Cada gasto en la lista debe mostrar el monto, la categoría, el RUC del comercio y la foto del comprobante.

#### 4. Microservicios:

- Desarrolla un microservicio para la gestión de usuarios (autenticación y autorización).
- Desarrolla un microservicio para el registro y recuperación de gastos.
- Asegúrate de que estos microservicios se comuniquen de manera eficiente.

#### 5. Almacenamiento de Datos:

- Utiliza una base de datos para almacenar la información de usuarios y gastos.
- La foto del comprobante debe almacenarse de manera segura y eficiente.

## 6. Interfaz de Usuario:

- Desarrolla un frontend interactivo que permita a los usuarios navegar fácilmente por el sistema.
- Asegúrate de que la interfaz de usuario sea amigable y responsive.

## Evaluación:

- La prueba técnica será evaluada en base a la implementación exitosa de los requisitos funcionales y la arquitectura de microservicios.
- Se valorará la eficiencia en el manejo de la base de datos y el almacenamiento seguro de las fotos de los comprobantes.
- La interfaz de usuario debe ser intuitiva y fácil de usar.

## Observaciones:

- Puedes utilizar las tecnologías y frameworks que consideres apropiados para el desarrollo.
- Proporciona documentación clara sobre cómo ejecutar y probar la aplicación.
- Enfócate en la modularidad y la escalabilidad de la solución.
- Súbelo a un repositorio de Git (GitHub o GitLab) y brinda acceso al siguiente usuario:  
**manueltemple@gmail.com**