Prueba Técnica Python

# Objetivo de la Prueba

Desarrollar una API RESTful utilizando Flask para gestionar el booking de tours. La API permitirá a los usuarios realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los tours y las reservas. La prueba evaluará la capacidad del candidato para trabajar con Flask, incluyendo la creación de modelos, rutas API, serializadores para la entrada y salida de datos, y pruebas unitarias.

# Requisitos Funcionales

## Modelo de Datos

- Tours: Deben incluir al menos los campos de nombre, descripción, fecha, y precio.  
- Reservas: Deben estar relacionadas con los tours e incluir los campos de usuario (nombre del usuario que reserva), fecha de reserva, y cantidad de personas.

## Operaciones API:

- Tours:  
 - Listar todos los tours disponibles.  
 - Obtener los detalles de un tour específico.  
 - Crear un nuevo tour.  
 - Actualizar un tour existente.  
 - Eliminar un tour.  
  
- Reservas:  
 - Crear una reserva para un tour.  
 - Listar todas las reservas de un usuario.  
 - Cancelar una reserva.

# Requisitos Técnicos

1. Modelos: Definir modelos de datos utilizando SQLAlchemy ORM.  
2. Serializadores: Implementar serializadores para transformar la entrada y salida de datos de la API.  
3. Rutas API: Definir las rutas API en Flask para acceder a las operaciones.  
4. Controladores: Desarrollar controladores para manejar las operaciones API.  
5. Pruebas Unitarias: Escribir pruebas unitarias para cada una de las operaciones API, asegurando que funcionen como se espera.  
6. Datos de Prueba: Opcionalmente, crear scripts para generar datos de prueba para los modelos.

# Instrucciones

## Configuración del Entorno:

- Asegurarse de tener instalado Python, pip y virtualenv.  
- Crear un nuevo entorno virtual y activarlo.  
- Instalar Flask, Flask-SQLAlchemy y otras dependencias necesarias.

## Desarrollo:

Seguir los requisitos técnicos para implementar las funcionalidades solicitadas. Utilizar las herramientas y características de Flask para lograr una estructura de código limpia y mantenible.

## Pruebas:

Realizar pruebas unitarias utilizando Pytest o unittest. Verificar que todas las operaciones API funcionen correctamente bajo diferentes escenarios.

## Documentación:

Incluir un archivo README.md con instrucciones sobre cómo configurar y ejecutar el proyecto, así como cómo ejecutar las pruebas unitarias.

## Entrega:

El código fuente debe ser subido a un repositorio Git (por ejemplo, GitHub, GitLab, Bitbucket) y compartir el enlace como parte de la entrega.

# Criterios de Evaluación

- Cumplimiento de los requisitos funcionales y técnicos.  
- Calidad del código y uso de las mejores prácticas de Flask.  
- Diseño y estructuración de la base de datos.  
- Cobertura y calidad de las pruebas unitarias.  
- Documentación y facilidad de uso de la API.

# Mejoras para Puntos Extras

Además de los requisitos básicos, los candidatos pueden considerar las siguientes mejoras para ganar puntos extras en su evaluación:  
- Implementación de autenticación y autorización (por ejemplo, JWT).  
- Integración de un sistema de caché para mejorar el rendimiento de la API.  
- Uso de contenedores (por ejemplo, Docker) para la implementación de la aplicación.  
- Creación de una interfaz de usuario para interactuar con la API.  
- Documentación de la API utilizando herramientas como Swagger o Postman.  
- Seguridad de la API, como la prevención de inyecciones SQL y la protección contra ataques de tipo CSRF o XSS.