

PRUEBA TECNICA

Desarrollador Backend Python / Flask – Integración API

Entrega: Código fuente + README + video corto explicativo

1. Objetivo General

Desarrollar un microservicio backend utilizando Python y Flask que actúe como intermediario entre un sistema interno y una API externa.

El objetivo de esta prueba es evaluar la capacidad del candidato para:

- Consumir APIs externas.
- Validar y transformar datos.
- Manejar errores correctamente.
- Organizar código backend de forma clara.
- Documentar decisiones técnicas.

La prueba simula un escenario real de integración entre sistemas mediante APIs REST.

2. Requerimientos Funcionales

2.1 Endpoint POST /clientes

Debe:

- Recibir datos desde Postman en formato JSON.
- Validar que existan al menos los campos:
 - name
 - email
- Enviar la información a una API externa mediante un POST.
- Retornar la respuesta al cliente.

Se debe manejar correctamente:

- errores de conexión
 - respuestas no exitosas de la API externa
 - validaciones faltantes
-

2.2 Endpoint GET /clientes/<id>

Debe:

- Consultar un cliente en la API externa.

- Transformar la respuesta antes de retornarla.

Ejemplo:

Si la API externa retorna múltiples campos, el endpoint local debe devolver únicamente:

- id
 - name
 - email
-

2.3 Endpoint PUT /clientes/<id>

Debe:

- Recibir datos desde Postman.
 - Enviar la actualización a la API externa.
 - Retornar el resultado de la operación.
 - Manejar errores si la API externa falla.
-

3. Requerimientos Técnicos Obligatorios

- Uso de Python y Flask.
- Uso de requests o httpx para consumo de API.
- Separación básica del código (no todo en un solo archivo).
- Manejo de excepciones.
- Uso de variables de entorno para URLs o configuraciones.
- Código legible y organizado.

Estructura sugerida (no obligatoria):

```
app/  
routes/  
services/  
utils/  
app.py
```

4. Documentación Requerida

Debe incluir un archivo README.md explicando:

1. Cómo ejecutar el proyecto.
2. Estructura del proyecto.
3. Qué hace cada endpoint.
4. Cómo se manejan los errores.

5. Qué cambios realizaría si la API externa requiriera autenticación Bearer Token.
 6. Qué mejoras implementaría en un entorno productivo.
-

5. Criterios de Evaluación

Se evaluará principalmente:

- Claridad y organización del código.
- Comprensión del flujo de integración API.
- Manejo de errores.
- Capacidad de transformar datos entre sistemas.
- Documentación y claridad técnica.

No se evalúa diseño frontend ni interfaz gráfica.

6. Video Explicativo

El candidato debe grabar un video de máximo 5 minutos donde explique:

1. Descripción general de la arquitectura de la aplicación.
2. Función de cada archivo principal.
3. Demostración de los endpoints utilizando Postman.
4. Explicación breve de las decisiones técnicas tomadas.

No es necesario explicar cada línea en detalle. El objetivo es verificar que el candidato entiende la arquitectura y el funcionamiento del código.

7. Entregables Finales

El candidato debe entregar:

1. Carpeta del proyecto con código fuente completo.
2. Archivo README.md claro y completo.
3. Colección de Postman utilizada para pruebas.
4. Video explicativo corto (máximo 5 minutos).