

PRUEBA TÉCNICA PARA DESARROLLADOR PYTHON DE INTEGRACIONES

Para el desarrollo de esta prueba puede utilizar todas las librerías de Python que crea convenientes, y organizar el script de la forma que crea más adecuada para su fácil entendimiento y ejecución, incluso creando sus propias funciones y clases y llamándolas dentro del script.

Recuerde que esta prueba no solo evaluará que su script funcione, sino que también la optimización de importaciones y recursos, la claridad y eficiencia del código, y la creatividad al resolver el problema de la forma más simple posible.

Esta prueba consta de 3 puntos, todos con igual importancia.

1. Implementa un flujo de procesamiento de pedidos en Python que se ejecute de forma asíncrona usando una cola de trabajo (queda a su consideración el uso de un framework).

El flujo debe incluir al menos cuatro etapas:

- a. Validación de los datos del pedido.
- b. Enriquecimiento de los productos consultando una API externa (usar <https://fakestoreapi.com>).
- c. Cálculo de totales y descuentos aplicando la lógica definida (ejemplo: 10% si el total supera 500). Adicional generar un Hash/ID único del pedido procesado
- d. Persistencia de los resultados en una base temporal (SQLite o en memoria). No duplicar por reintentos (usar pedido.id como clave).

El sistema debe soportar encolar múltiples pedidos y procesarlos de forma concurrente.

Notas:

- Ejemplo de entrada de pedido

```
{  
  "id": 123,  
  "cliente": "ACME Corp",  
  "productos": [  
    {"sku": "P001", "cantidad": 3, "precio_unitario": 10},  
    {"sku": "P002", "cantidad": 5, "precio_unitario": 20}  
  ],  
  "fecha": "2025-01-01T10:30:00Z"  
}
```

- Para completar la información del producto usando el **API**: Fake Store API (base <https://fakestoreapi.com>)
 - GET /products (lista), GET /products/{id} (detalle).
 - Campos: id, title, price, description, category.
- Se debe considerar el log de procesamiento de la cola y la gestión de reintentos en caso de error

2. Desde la plataforma web se le da al usuario la opción de seleccionar una operación para transformar los datos para la integración entre Fractal y su ERP. Estas funciones pueden ir desde convertir de mayúscula a minúscula hasta unir dos campos. El equipo de integraciones constantemente está implementando nuevas funciones.

NECESIDAD:

Escriba, en Python, 3 funciones (suma, resta, y mayuscula_a_minuscula) y una cuarta función main.py donde recibe 3 parámetros: el nombre de la función, var1 y var2. El código debe funcionar incluso si se crea una 4ta función sin necesidad de modificar el código.

3. En Fractal, un requerimiento constante que recibimos de parte de la mayoría de nuestros clientes es integrar Fractal con sus ERP. Para suplir esta necesidad, Fractal disponibiliza un conjunto de servicios web (Fractal API, cuya documentación se encuentra <https://api.fractal.com/>) y en el equipo de integraciones se realizan soluciones a la medida para integrar Fractal con los ERP. Sin embargo, hemos encontrado varias falencias a la forma de trabajo actual:
- a. La integración usando servicios web, actualmente se hace por pulling (consulta todos los datos o en un rango de tiempo), lo cual es poco eficiente para los momentos en los que no hay datos o ya se realizó la creación o actualización.
 - b. Nuestros clientes desean integraciones en tiempo real y la consulta por pulling de los servicios web no es una opción para lograrlo. (ejemplo: hacer pulling cada segundo no es viable)
 - c. Las soluciones a la medida son poco escalables, porque necesitamos de mucho personal para atender las integraciones de todos nuestros clientes.

En este punto, requerimos que diseñe una arquitectura de componentes (puede dibujarla en papel) en la cual nos plantee una solución a los problemas descrito anteriormente, en donde podamos reutilizar código de las integraciones realizadas anteriormente (tenga en cuenta que normalmente se deben realizar transformación a los datos) y donde Fractal pueda emitir eventos para reducir la cantidad de peticiones que se realizan a los servicios web.

Notas:

- Tenga en cuenta que Fractal es una aplicación como servicio y en una sola base de datos se encuentran todas las compañías (multitenant).
- En caso de pasar a la entrevista técnica, se te preguntarán temas de:
 - Solución al problema de la multicompañía (multitenant)



- Altos volúmenes de datos (empresas que solo pueden enviar toda la información)
- Manejo de errores y opción de reintentar un evento en caso de caída del servicio