

# Desafío en Python con Sanic

## Requisitos:

Python 3.9 o superior  
Sanic  
Docker  
MongoDB  
Conocimientos en arquitectura de microservicios orientada a eventos  
Experiencia con RESTful APIs  
Conocimientos en mensajería con RabbitMQ

## Descripción del Desafío:

### 1. Desarrollar dos microservicios usando Sanic y Docker:

#### -Microservicio de Cuentas:

Permite crear, consultar, actualizar y eliminar cuentas de usuarios.  
Expondrá una API RESTful para la gestión de cuentas.  
Almacenará la información de las cuentas en MongoDB.

#### -Microservicio de Transacciones:

Permite realizar transferencias de dinero entre cuentas.  
Se comunicará con el microservicio de cuentas para obtener información de las cuentas.  
Utilizará RabbitMQ para la comunicación entre microservicios.  
Almacenará las transacciones en MongoDB.

#### - Implementación:

Crear dos proyectos independientes con Sanic, uno para cada microservicio.  
Definir las APIs RESTful para cada microservicio.  
Implementar la lógica de negocio para cada microservicio.  
Utilizar Docker para contenerizar cada microservicio.  
Configurar MongoDB para el almacenamiento de datos.  
Conectar los microservicios a MongoDB.  
Implementar la mensajería con RabbitMQ para el microservicio de transacciones.  
Probar los microservicios de forma individual y conjunta.

#### - Evaluación:

Se evaluará la capacidad del postulante para:

Diseñar e implementar APIs RESTful con Sanic.  
Implementar la lógica de negocio para la gestión de cuentas y transacciones.

Contenerizar los microservicios con Docker.  
Configurar y utilizar MongoDB para el almacenamiento de datos.  
Conectar los microservicios a MongoDB.  
Implementar la mensajería con RabbitMQ para la comunicación entre microservicios.  
Escribir código limpio, modular y bien documentado.  
Probar los microservicios de forma individual y conjunta.

Recursos adicionales:

Documentación de Sanic: <https://sanic.dev/>  
Documentación de Docker: <https://docs.docker.com/>  
Documentación de MongoDB: <https://docs.mongodb.com/>  
Documentación de RabbitMQ: <https://www.rabbitmq.com/tutorials/tutorial-one-python>

### **Entrega:**

Los postulantes deberán entregar un repositorio que contenga:

El código fuente de los dos microservicios.  
Los archivos Dockerfile para cada microservicio y un Docker Compose para levantar los servicios.  
La configuración de MongoDB.  
La configuración de RabbitMQ.  
Instrucciones para ejecutar los microservicios.