

# Prueba Técnica para Desarrollador Python en Celes

## Objetivo:

Desarrollar un microservicio en Python que interactúe con un Datamart y proporcione una interfaz para realizar consultas y operaciones específicas. Además, el microservicio deberá integrar la autenticación con Firebase y tener un enfoque en pruebas unitarias y CI/CD.

## Tareas:

### Conexión con Datamart:

Para realizar esta prueba, deberás descargar el archivo .zip en esta ruta [https://drive.google.com/file/d/1s0irlrngQVeRDXY8F5gizkttG9Rqshg0/view?usp=drive\\_link](https://drive.google.com/file/d/1s0irlrngQVeRDXY8F5gizkttG9Rqshg0/view?usp=drive_link) y descomprimir el archivo en una carpeta local donde crearás el microservicio. Esta carpeta debe estar excluida de git.

### Endpoints expuestos:

Deberás crear un proyecto en Python usando Flask o FastAPI para exponer los siguientes endpoints

- Consultar las ventas en un periodo por empleado (KeyEmployee)
- Consultar las ventas en un periodo por producto (KeyProduct)
- Consultar las ventas en un periodo por tienda (KeyStore).

- Consultar la venta total y promedio por tienda (KeyStore)
- Consultar la venta total y promedio por producto (KeyProduct)
- Consultar la venta total y promedio por empleado (KeyEmployee)

En este punto se evaluará la manera como manejas errores y logs.

## Implementación de Seguridad: Autenticación con JWT

### Generación del Token JWT:

Implementar un endpoint que permita la autenticación de usuarios. Este endpoint puede recibir credenciales (como un nombre de usuario y contraseña) y validarlas.

Una vez autenticado, el servicio generará un token JWT que incluye la identidad del usuario y un tiempo de expiración.

## Uso del Token:

Los endpoints para consultar ventas requerirán que las solicitudes incluyan el token JWT en el encabezado de autorización.

El microservicio validará el token en cada solicitud antes de procesarla.

## Seguridad y Manejo de Tokens:

Asegurarse de que los tokens se generen y manejen de manera segura, utilizando librerías establecidas.

Implementar medidas para evitar vulnerabilidades comunes como la exposición de tokens o ataques de inyección.

## Pruebas Unitarias:

Desarrollar pruebas unitarias para las funciones críticas.

## Documentación:

Documentar el diseño y uso del microservicio.

Incluir instrucciones para configurar y desplegar el servicio.

## Puntos adicionales:

Diseñar un flujo básico de CI/CD utilizando herramientas como GitHub Actions o GitLab CI.

El flujo debe incluir: ejecución de pruebas unitarias, análisis de código y despliegue automatizado en un entorno de pruebas de su propuesta.

## Criterios de Evaluación:

Tu código debe ser entregado en github, desde donde será clonado y ejecutado localmente para hacer la validación y evaluación.

Debes demostrar buenas prácticas de Git. Esto incluye mensajes de commit significativos y commits lógicos y pequeños que faciliten la comprensión del proceso de desarrollo que seguiste.

Calidad del código (claridad, mantenibilidad, uso de buenas prácticas).

Funcionalidad del microservicio.

Cobertura y calidad de las pruebas unitarias.

Correcta implementación de la autenticación y seguridad.