Test Técnico

1. Requerimientos técnicos

- **Duración estimada:** De 4 a 6 horas de trabajo aproximadamente (repartido en un plazo de 3-5 días).
- **Formato:** Envío de un repositorio de código (preferiblemente en GitHub, GitLab o Bitbucket).
- **Entregables esperados**
    - Código funcional.
    - Documentación (README) explicando la solución, diagrama de arquitectura Amazon Web Services (AWS), decisiones técnicas y cómo ejecutar/desplegar el proyecto.
    - Tests unitarios o de integración, si corresponde.

2. Problema a resolver

Objetivo: Diseñar y desarrollar un microservicio serverless que procese órdenes de trabajo para un taller de servicio técnico.

Una orden de trabajo puede tener, entre otros, los siguientes atributos: *identificador*, *fecha de registro*, *descripción*, *fecha de entrega* y *estado*. El estado de una orden de trabajo puede ser *recibida*, *en proceso*, *completada* y *cancelada*. Las órdenes canceladas deben tener un motivo que describa el por qué de la cancelación. Las órdenes pueden llegar a este servicio en cualesquiera de los estados antes mencionados.

Descripción:

- Implemente un servicio serverless que permita recibir órdenes de trabajo con diferentes estados a través de una API REST.
- Los datos recibidos deben ser almacenados en una base de datos (BD), de preferencia NoSQL.
- El servicio debe incluir funcionalidades para:
    1. Validar los datos de entrada.
    2. Cada orden de trabajo, además de ser almacenada en BD, se debe enviar a un sistema de colas según el estado de la misma. En este sistema de colas, las órdenes en un estado A, no deben estar junto con las órdenes en otro estado.
        - Otro componente del servicio será el encargado de procesar las órdenes en el sistema de colas.
- Documente cómo escalaría esta solución para manejar un alto volumen de solicitudes, y qué cambios implementaría en la solución propuesta para procesar las órdenes en el mismo orden en que fueron recibidas.

Requisitos técnicos

- El código debe estar implementado en **Python** o **C#**.
- Utilizar **Serverless Framework** para la infraestructura como código (IaC).
- Implementar testing unitario para las funciones principales.
- Incluir un README en el repositorio de código con la documentación solicitada en "**Entregables esperados".**

Servicios de AWS a considerar

- AWS Step Functions
- Amazon ApiGateway
- Amazon EventBridge
- Amazon DynamoDB
- Amazon Simple Queue Service
- Amazon Simple Notification Service