

## Prueba Técnica: Automatización de Reportes Financieros

### Objetivo:

Diseñar un sistema (AWS Lambdas) que procese transacciones financieras almacenadas en MongoDB, genere reportes detallados en formato XLSX con gráficos, y los guarde para su consulta posterior.

### Escenario:

La empresa tiene una base de datos con transacciones financieras en MongoDB. Cada transacción contiene información sobre clientes, montos, fechas y categorías. Se necesita un sistema que genere un reporte mensual con análisis visual de las transacciones, incluyendo:

1. Total de ingresos y gastos por categoría.
2. CRUD de transacciones.
3. Resumen por cliente (totales, promedios).

El reporte debe incluir gráficos que visualicen estas métricas y ser almacenado para su distribución.

### Tareas Técnicas:

#### 1. Configurar MongoDB:

Crear una colección llamada 'transacciones' con la siguiente estructura de documento:

```
{
  "transaccion_id": "UUID",
  "cliente_id": "UUID",
  "cantidad": "float",
  "categoría": "string",
  "fecha": "ISODate",
  "tipo": "string" // "income" o "expense"
}
```

#### 2. Crear una función CRUD con Lambda para gestionar transacciones:

- Crear registro de transacción (createTransaction):
  - Método HTTP: POST
  - Endpoint: '/transaction'
  - Descripción: Permite registrar una nueva transacción.
- Obtener transacciones (readTransactions):
  - Método HTTP: GET
  - Endpoint: '/transactions'
  - Descripción: Recupera todas las transacciones.

- Obtener transacciones de cliente particular (readUserTransactions):
  - Método HTTP: GET
  - Endpoint: '/transactions/{cliente\_id}'
  - Descripción: Recupera todas las transacciones del usuario indicado.
- Actualizar transacción (updateTransaction):
  - Método HTTP: PUT
  - Endpoint: '/transaction/{transaccion\_id}'
  - Descripción: Permite modificar los detalles de una transacción existente.
- Desactivar transacción (deactivateTransaction):
  - Método HTTP: DELETE
  - Endpoint: '/transaction/{transaccion\_id}'
  - Descripción: Desactiva una transacción en lugar de eliminarla. Las transacciones desactivadas ya no pueden ser editadas.

## 2. Crear una función Lambda para procesar datos financieros:

- Entrada:
  - Rango de fechas (e.g., '2025-01-01' a '2025-01-31').
- Acciones:
  1. Conectar a la base de datos MongoDB.
  2. Consultar las transacciones en el rango de fechas especificado.
  3. Usar pandas para calcular:
    - Total de ingresos y gastos por categoría.
    - Resumen de transacciones por cliente (totales y promedios).

## 3. Crear una función Lambda para generar el reporte en XLSX:

- Entrada:
  - Datos procesados de la primera función Lambda.
- Acciones:
  1. Crear un archivo XLSX utilizando openpyxl o xlsxwriter.
  2. Incluir hojas para:
    - Resumen por categoría (ingresos y gastos).
    - Resumen por cliente.
  3. Agregar gráficos:
    - Gráfico de barras para ingresos/gastos por categoría.
    - Gráfico de línea para ingresos/gastos para un cliente seleccionado en el rango de fechas establecido.
  4. Guardar el archivo temporalmente.

## 4. Crear una función Lambda para almacenar el reporte:

- Entrada:
  - Archivo XLSX generado.

- Acciones:

1. Subir el archivo a un bucket de S3 (opcional) con un nombre que incluya el rango de fechas.
2. Generar un enlace de descarga firmado con un tiempo de expiración configurado (opcional).

5. Crear el archivo `serverless.yml`:

Configurar el proyecto para desplegar con Serverless Framework. El archivo `serverless.yml` debe incluir:

- Configuración de las funciones Lambda.
- Definición de los permisos necesarios para interactuar con MongoDB y S3.
- Variables para configurar los buckets de S3 (opcional), las URIs de MongoDB y otros parámetros.
- Configuración del entorno para cada función Lambda.

Requisitos Adicionales:

1. Datos de prueba:

Archivo JSON con transacciones de muestra para poblar la base de datos MongoDB.

2. Infraestructura:

- MongoDB: Puede ser local o un clúster en MongoDB Atlas.
- AWS: Usar AWS Lambda y S3 (opcional).
- Despliegue: Archivo `serverless.yml` para Serverless Framework.

3. Entrega:

- Código de las funciones Lambda en Python.
- Script para poblar la base de datos MongoDB.
- Ejemplo del archivo `serverless.yml`.
- Ejemplo del reporte XLSX generado.

Criterios de Evaluación:

1. Documentación.
2. Lógica, procesamiento de datos y generación del reporte.
3. Calidad visual y estructura del reporte en XLSX.
4. Uso eficiente de pandas para análisis de datos.
5. Integración de MongoDB, AWS Lambda y S3.
6. Despliegue exitoso utilizando Serverless Framework.

¡Buena suerte!