Ejercicio 1 – API de Consulta de Clientes (HTTP Trigger)

Una empresa de e-commerce necesita una API serverless que permita consultar información de clientes por su ID. El endpoint debe devolver un JSON con el nombre, correo y estado del cliente. Debe desplegarse en Azure Function App (Consumption Plan), con autenticación de clave de función y monitoreo habilitado con Application Insights.

Pasos a seguir:

- Crear un proyecto Azure Functions con HTTP Trigger (authlevel function).
- Leer el parámetro 'id' desde la query string.
- Simular consulta a base de datos devolviendo datos en formato JSON.
- Configurar Application Insights para monitoreo de peticiones y errores.
- Publicar la Function App en Azure (Consumption Plan).
- Probar el endpoint con Postman usando la API Key.
- Configurar logging en el código para registrar cada solicitud recibida.

Ejercicio 2 – Proceso de Reportes Diarios (Timer Trigger)

Un banco requiere ejecutar una función todos los días a las 6:00 AM para generar un reporte de transacciones y guardarlo en Azure Blob Storage. Debe desplegarse en Azure y configurarse para que envíe logs detallados a Application Insights.

Pasos a seguir:

- Crear función con Timer Trigger usando expresión CRON para las 6:00 AM.
- Generar un archivo CSV simulado con datos de transacciones.
- Subir el archivo generado a un contenedor en Azure Blob Storage.
- Configurar conexión a Azure Storage mediante Application Settings.
- Publicar en Azure Function App y habilitar Application Insights.
- Validar que el archivo se genera diariamente y revisar logs en el portal.

Ejercicio 3 – Procesador de Imágenes (Blob Trigger + Output Binding)

Una agencia de marketing necesita que cada vez que un diseñador suba una imagen a un contenedor de Azure Blob Storage, esta sea procesada (por ejemplo, agregando una marca de agua) y guardada en otro contenedor. El sistema debe registrar en logs la ruta del archivo procesado.

Pasos a seguir:

- Crear función con Blob Trigger para escuchar subidas al contenedor 'incoming-images'.
- Procesar la imagen (simular con cambio de nombre o texto).
- Guardar resultado en contenedor 'processed-images' usando Output Binding.
- Configurar conexión a Blob Storage con variables en Application Settings.
- Publicar Function App en Azure y habilitar Application Insights.
- Subir una imagen de prueba y validar procesamiento.

Ejercicio 4 – Procesamiento de Mensajes (Queue Trigger)

Un sistema de pedidos online envía mensajes a una Azure Storage Queue con información de nuevas órdenes. Se necesita una función que procese cada mensaje, guarde los datos en Azure Table Storage y envíe logs al sistema de monitoreo.

Pasos a seguir:

- Crear función con Queue Trigger escuchando la cola 'orders'.
- Deserializar el mensaje JSON recibido.
- Guardar datos en Azure Table Storage usando Output Binding.
- Configurar Application Insights para registrar cada orden procesada.
- Publicar en Azure Function App y probar enviando mensajes con Azure CLI.

Ejercicio 5 – Orquestación de Tareas en Paralelo (Durable Functions)

Una empresa de logística requiere procesar múltiples rutas de entrega en paralelo para calcular el tiempo estimado total. La solución debe usar Durable Functions con patrón fan-out/fan-in, registrar tiempos de ejecución y mostrar estado de la orquestación en el portal de Azure.

Pasos a seguir:

- Crear Durable Functions con un orchestrator que llame en paralelo a múltiples activity functions.
- Cada activity function simula el cálculo de una ruta y devuelve un tiempo estimado.
- El orchestrator suma los tiempos y devuelve el total.
- Publicar Function App en Azure con Application Insights.
- Monitorear estado de ejecución desde Azure Portal.
- Configurar seguridad para que solo usuarios autenticados en Azure AD puedan iniciar la orquestación.