AZURE LAB

Este laboratorio tiene como objetivo poder usar algunos de los conceptos aprendidos durante las sesiones como Azure Storage Blob, Azure Storage Queue, Azure Functions y Azure Event Grid.

El ejercicio propuesto es la eliminación de un registro de base de datos y las imágenes asociadas que estén contenidas en Azure Storage.

# EVALUACIÓN

La evaluación se realizará en función de qué servicios de Azure se usen para resolver el ejercicio siguiendo los criterios especificados a continuación:

* Aprobado: Mediante una llamada al método DeleteGame de la Api, usar el SDK de azure para eliminar las imágenes.
  + Azure Storage Blobs: Eliminar las imágenes asociadas a un registro de base de datos
* Notable: Mediante una llamada al método DeleteGame de la Api, guardar un mensaje en una cola que dispare la Azure Function para eliminar las imágenes
  + Azure Storage Blobs
  + Azure Storage Queue: Al eliminar un registro, guardar un mensaje en la cola que identifique el registro que se ha eliminado
  + Azure Functions: Con un trigger de cola, lee el mensaje que identifica el registro eliminado y realiza la eliminación de las imágenes asociadas
* Sobresaliente: Mediante una suscripción de Azure Event Grid a Blob Storage, llamar a una Azure Function que realice la eliminación de los thumbnails.
  + Azure Event Grid: Que realice el enlace entre el evento de eliminación de screenshots y una Azure Function
  + Azure Functions: Disparada con un trigger de Event Grid, elimina los thumbnails asociados al registro eliminado.

# AYUDA

## Eliminación de las imágenes

private BlobContainerClient GetBlobContainerClient(string container)

{

StorageSharedKeyCredential storageCredentials = new StorageSharedKeyCredential(storageOptions.AccountName, storageOptions.AccountKey);

BlobContainerClient blobContainerClient = new BlobContainerClient(storageOptions.ConnectionString, container);

return blobContainerClient;

}

public async Task DeleteFolderBlobs(string container, string folder)

{

var containerClient = GetBlobContainerClient(container);

var folderBlobs = containerClient.GetBlobsAsync(prefix: folder);

await foreach (var blobItem in folderBlobs)

{

BlobClient blobClient = containerClient.GetBlobClient(blobItem.Name);

await blobClient.DeleteIfExistsAsync();

}

}

## Guardar mensaje en una cola

Más información [Tutorial: Work with Azure Queue Storage queues in .NET | Microsoft Learn](https://learn.microsoft.com/en-us/azure/storage/queues/storage-tutorial-queues?tabs=dotnet%2Cenvironment-variable-windows)

QueueClient queue = new QueueClient(connectionString, "mystoragequeue");

static async Task InsertMessageAsync(QueueClient theQueue, string newMessage)

{

if (null != await theQueue.CreateIfNotExistsAsync())

{

Console.WriteLine("The queue was created.");

}

await theQueue.SendMessageAsync(newMessage);

}

## Leer mensajes de eventGrid

var myEvent = eventGridEvent.Data.ToObjectFromJson<StorageBlobDeletedEventData>();

[Azure Blob Storage as Event Grid source - Azure Event Grid | Microsoft Learn](https://learn.microsoft.com/en-us/azure/event-grid/event-schema-blob-storage?tabs=event-grid-event-schema)

[azure-sdk-for-net/Sample3\_ParseAndDeserializeEvents.md at main · Azure/azure-sdk-for-net · GitHub](https://github.com/Azure/azure-sdk-for-net/blob/main/sdk/eventgrid/Azure.Messaging.EventGrid/samples/Sample3_ParseAndDeserializeEvents.md)