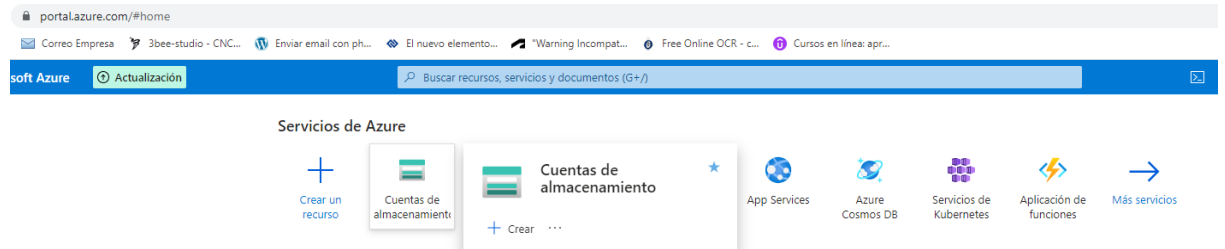
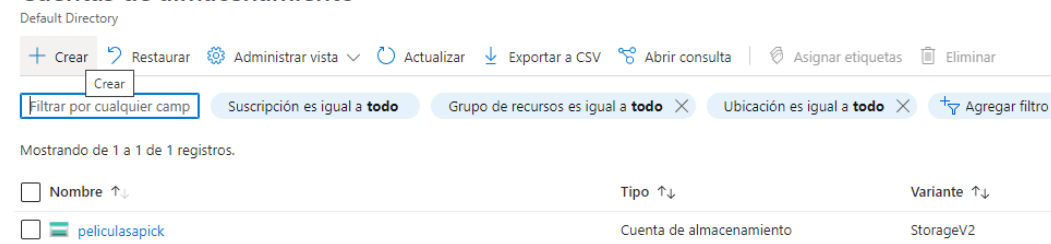


GUARDAR IMAGEN EN AZURE STORAGE

1. Vamos a <https://portal.azure.com/#home> y entramos a Cuentas de Almacenamiento



2. Vamos a crear una cuenta en el +
Cuentas de almacenamiento



3. Configuramos la cuenta de la siguiente manera, en recursos le damos a crear nuevo y en nombre de la cuenta de almacenamiento le damos un nombre. Lo demás lo dejamos como esta en la imagen.

Datos básicos | Opciones avanzadas | Redes | Protección de datos | Cifrado | Etiquetas | Revisar

Seleccione la suscripción en la que se creará la nueva cuenta de almacenamiento. Elija un grupo de recursos nuevo o uno ya existente para organizar y administrar la cuenta de almacenamiento junto con otros recursos.

Suscripción *

Grupo de recursos * [Crear nuevo](#)

Detalles de la instancia

Si necesita crear un tipo de cuenta de almacenamiento heredada, haga clic en [aquí](#).

Nombre de la cuenta de almacenamiento
 ⓘ * ✖ El nombre de cuenta de almacenamiento "películasapi" ya está en uso.

Región ⓘ *
 [Implementar en una zona perimetral](#)

Rendimiento ⓘ *
 ☒ **Estándar:** Opción recomendada para la mayoría de los escenarios (cuenta de uso general v2)
 ☐ **Prémium:** Se recomienda para escenarios que requieren una latencia baja.

Redundancia ⓘ *
 ☒ Habilite el acceso de lectura a los datos en el caso de que la región no esté disponible.

- En la pestaña de advanced dejar así excepto secure transfer required se deja en enabled para producción.

Create storage account

Basics Networking Data protection **Advanced** Tags Review + create

Security

Secure transfer required ☒ Disabled ☐ Enabled

Minimum TLS version

Infrastructure encryption ☒ Disabled ☐ Enabled

Sign up is currently required to enable infrastructure encryption on a per-subscription basis. [Sign up for infrastructure encryption](#)

Blob storage

Allow Blob public access ☐ Disabled ☒ Enabled

Blob access tier (default) ☐ Cool ☒ Hot

NFS v3 ☒ Disabled ☐ Enabled

Sign up is currently required to utilize the NFS v3 feature on a per-subscription basis. [Sign up for NFS v3](#)

- Vamos a review+create y después create.
- Vamos al menú de la cuenta de almacenamiento y nos dirigimos al apartado claves de acceso y copiamos la cadena de conexión.

Cuentas de almace... «

Default Directory

+ Crear ↻ Restaurar ...

Filtrar por cualquier campo...

Nombre ↑

películasapick ...

películasapick | Claves de acceso ☆ ...

Cuenta de almacenamiento

Buscar

Establecer recordatorio de rotación Actualizar Enviar comentarios

Las claves de acceso autentican las solicitudes de las aplicaciones en esta cuenta de almacenamiento. Guarden una ubicación segura, como Azure Key Vault, y reemplácelas a menudo por nuevas claves. Las dos claves le reemplazarán una mientras sigue usando la otra.

Recuerde actualizar las claves con cualquier recurso y aplicación de Azure que usen esta cuenta de almacenamiento. [Más información sobre la administración de claves de acceso de la cuenta de almacenamiento](#)

Nombre de la cuenta de almacenamiento

películasapick

key1 Girar clave

Última rotación: 24/4/2023 (hace 23 días)

Clave

Mostrar

Cadena de conexión

DefaultEndpointsProtocol=https;AccountName=películasapick;AccountKey=WTeL... Ocultar

key2 Girar clave

Última rotación: 24/4/2023 (hace 23 días)

Clave

Mostrar

Cadena de conexión

Mostrar

Información general

Registro de actividad

Etiquetas

Diagnosticar y solucionar problemas

Control de acceso (IAM)

Migración de datos

Eventos

Explorador de almacenamiento

Storage Mover

Almacenamiento de datos

Contenedores

Recursos compartidos de archivos

Colas

Tablas

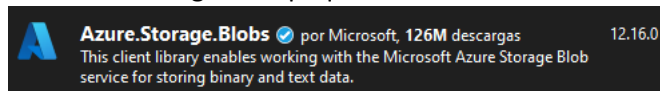
Seguridad y redes

Redes

CDN de Azure

Claves de acceso

7. Instalamos el siguiente paquete



8. Creamos una clase AlmacenadorAzureStorage y implementamos lo siguiente:

```
namespace backEnd.Utilidades
{
    1 referencia
    public class AlmacenamientoAzureStorage
    {
        private string connectionString;

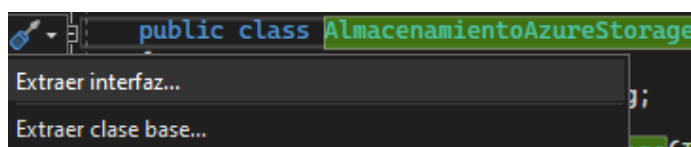
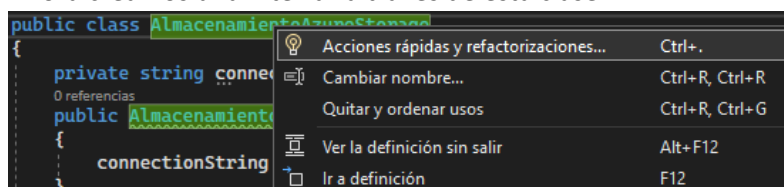
        0 referencias
        public AlmacenamientoAzureStorage(IConfiguration configuration)
        {
            connectionString = configuration.GetConnectionString("AzureStorage");
        }

        1 referencia
        public async Task<string> GuardarArchivo(string contenedor, IFormFile archivo)
        {
            var cliente = new BlobContainerClient(connectionString, contenedor);
            await cliente.CreateIfNotExistsAsync();
            cliente.SetAccessPolicy(Azure.Storage.Blobs.Models.PublicAccessType.Blob);
            var extension = Path.GetExtension(archivo.FileName);
            var archivoNombre = $"{Guid.NewGuid()}{extension}";
            var blob = cliente.GetBlobClient(archivoNombre);
            await blob.UploadAsync(archivo.OpenReadStream());
            return blob.Uri.ToString();
        }

        1 referencia
        public async Task BorrarArchivo(string contenedor, string ruta)
        {
            if (string.IsNullOrEmpty(ruta))
                return;
            var cliente = new BlobContainerClient(connectionString, contenedor);
            await cliente.CreateIfNotExistsAsync();
            var archivo = Path.GetFileName(ruta);
            var blob = cliente.GetBlobClient(archivo);
            await blob.DeleteIfExistsAsync();
        }

        0 referencias
        public async Task<string> EditarArchivo(string contenedor, IFormFile archivo, string ruta)
        {
            await BorrarArchivo(ruta, contenedor);
            return await GuardarArchivo(contenedor, archivo);
        }
    }
}
```

9. Ahora creamos una interfaz a través de esta clase.



Cambiamos el nombre , OK y se crea la interfaz con los métodos implementados

10. Vamos al program.cs o startup.cs y adicionamos el servicio.

```
builder.Services.AddTransient<IAmacenamientoArchivos, AlmacenamientoAzureStorage>();
```

11. Ahora vamos al controller donde se usa la imagen dentro de una entidad y usamos este código para crear la entidad

```
[HttpPost]
public async Task<ActionResult> Post([FromForm] ActorCreacionDTO actorCreacionDTO)
{
    var actor = mapper.Map<Actor>(actorCreacionDTO);

    if (actorCreacionDTO.Foto != null)
    {
        actor.Foto = await almacenadorArchivos.GuardarArchivo(contenedor, actorCreacionDTO.Foto);
    }

    context.Add(actor);
    await context.SaveChangesAsync();
    return NoContent();
}
```

12. Por ultimo agregamos la conexión con Azure en appsettings.json que copiamos en el numeral 6.

```
"connectionStrings": {
  "defaultConnection": "Data Source=(localdb)\\MSSQLLocalDB; Initial Catalog=UniversidadAPI; Integrated Security=True",
  "AzureStorage": "DefaultEndpointsProtocol=https;AccountName=peculasapick;AccountKey=WtLRJII/19YKP/jBRI6EGs0Eghc7N/kOh2kzASJfGy1N1fdgV7FTRN4aCaSfDuOtwSau1V2PaN+ASTe7z5Bg==;EndpointSuffix=core.windows.net"
```