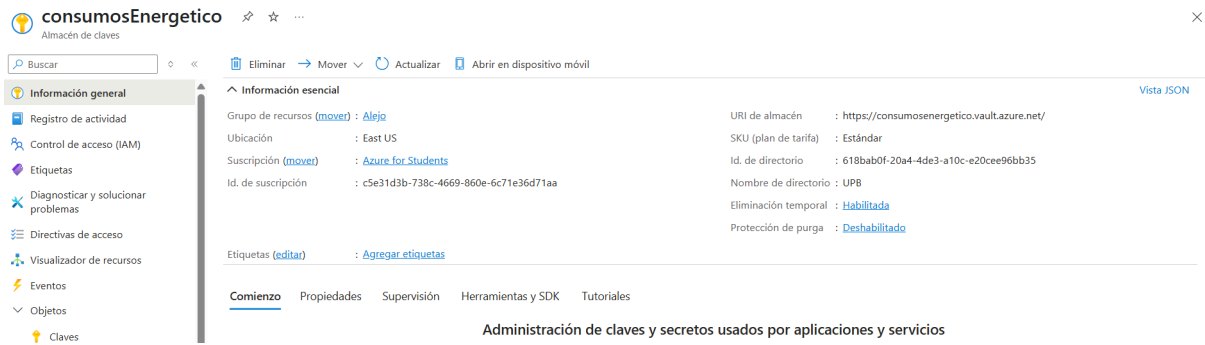
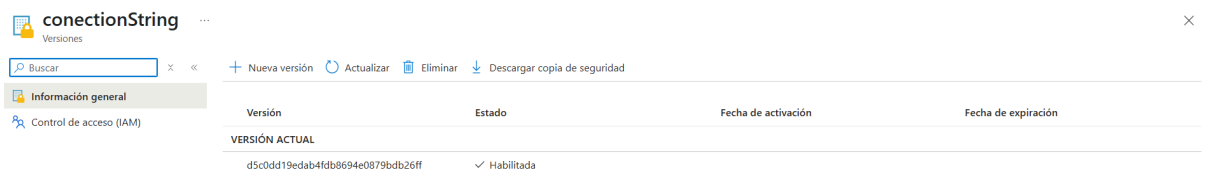


Se parte de la creación de una cuenta en Azure y se debe dirigir al producto “Almacén de claves”.

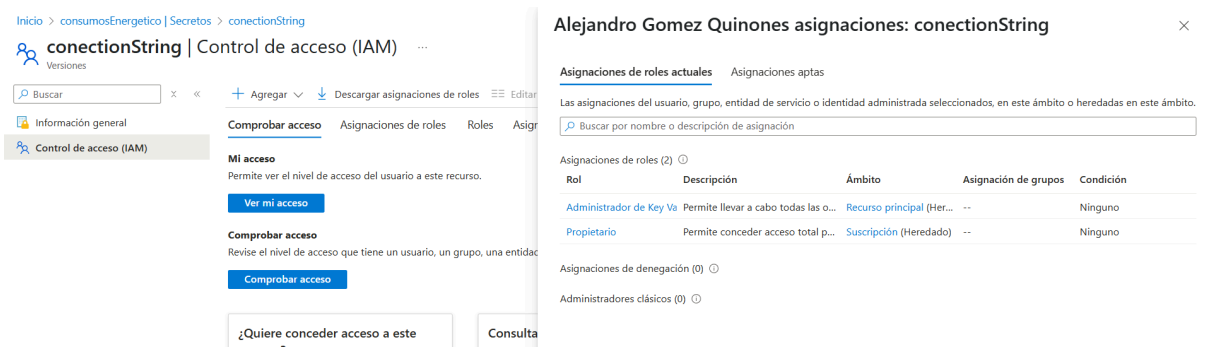
Luego se debe crear un almacén de claves con el nombre que se desee, elegir la ubicación y el grupo de recursos al que pertenece.



Dentro del almacén de claves, en la sección de secretos, se suben los secretos que se desean almacenar.



De manera adicional, se debe configurar el control de acceso para permitir el acceso a los usuarios que pueden interactuar con dicho secreto.



## Asignación de rol

### Adición de la asignación de roles ...

Rol	Miembros		
	Condiciones		
	Revisión y asignación		
Rol	Ninguno		
Ámbito	/subscriptions/c5e31d3b-738c-4669-860e-6c71e36d71aa/resourceGroups/Alejo/providers/Microsoft.KeyVault/vaults/consumosEnergetico/secrets/connectionString		
Miembros	Nombre	Id. de objeto	Tipo
	Juan Dario Rodas Marin	3a4c29dc-b4a1-4036-936e-0cf5821799f8	Usuario
Descripción	Sin descripción		

Ahora, para permitir la conexión a los secretos desde el proyecto, se deben instalar las siguientes extensiones utilizando NuGet. Puede hacerse desde la interfaz de usuario o desde la terminal.

Consola

```
dotnet add package Azure.Identity
dotnet add package Azure.Security.KeyVault.Secrets
```

Copiar

Dentro de DbContext, se agregar las dependencias a usar de Azure, las instaladas anteriormente.

```
using Npgsql;
using System.Data;
using Azure.Identity;
using Azure.Security.KeyVault.Secrets;
using Azure.Core;
```

Ahora dentro de la implementación agregamos algunas configuraciones como el delay, número de reintentos y demás.

También, desde Azure se obtiene el string de conexión, que permite conectarse al gestor de secretos. Es importante tener en cuenta que solo se pueden conectar a dicho gestor de secretos aquellos usuarios que previamente agregamos en el control de acceso.

Finalmente, se hace que la aplicación lea el string de conexión a la base de datos desde el gestor de secretos.

```

namespace CONSUMOS_ENERGETICOS_CS_REST_SQL_API.DbContexts
{
    11 referencias
    public class PgsqlDbContext(IConfiguration unaConfiguracion)
    {
        33 referencias
        public IDbConnection CreateConnection()
        {
            SecretClientOptions options = new SecretClientOptions()
            {
                Retry =
                {
                    Delay= TimeSpan.FromSeconds(2),
                    MaxDelay = TimeSpan.FromSeconds(16),
                    MaxRetries = 5,
                    Mode = RetryMode.Exponential
                }
            };
            var client = new SecretClient(new Uri("https://consumosenergetico.vault.azure.net/"), new DefaultAzureCredential(), options);

            KeyVaultSecret secret = client.GetSecret("conectionString");

            string secretValue = secret.Value;

            return new NpgsqlConnection(secretValue);
        }
    }
}

```