

Roles y Seguridad

<input checked="" type="checkbox"/> Checklist	<input checked="" type="checkbox"/>
# Día	3
Estado	Completado
Fecha	@21 de diciembre de 2025
Notas Técnicas	<p>1. Roles y nivel de alcance:</p> <p>En PostgreSQL, los roles son entidades globales al cluster. Sus atributos (LOGIN, SUPERUSER, CREATEDB, CREATEROLE) aplican a todo el cluster, mientras que los permisos se asignan a nivel de base de datos y objetos específicos.</p> <p>2. Atributos de rol vs permisos:</p> <p>Los atributos de un rol definen capacidades administrativas (como crear bases de datos o roles) y no deben confundirse con los permisos sobre objetos, los cuales se gestionan mediante GRANT y REVOKE.</p> <p>3. Herencia de roles:</p> <p>Asignar un rol a un usuario mediante GRANT no implica que sus privilegios se apliquen automáticamente. Para ello, el rol debe tener el atributo INHERIT o el usuario debe activar el rol con SET ROLE.</p> <p>4. Uso de psql y prompts:</p> <p>En psql, el prompt <code>=#</code> indica que el cliente está listo para ejecutar comandos, mientras que <code>-#</code> significa que una instrucción está incompleta y debe cerrarse o cancelarse antes de continuar.</p>

	5. Buenas prácticas de seguridad:
No es recomendable otorgar el atributo SUPERUSER a usuarios de aplicación. Es preferible usar roles específicos, permisos mínimos y control explícito del acceso a schemas y tablas.	
= Tiempo Invertido	6 horas

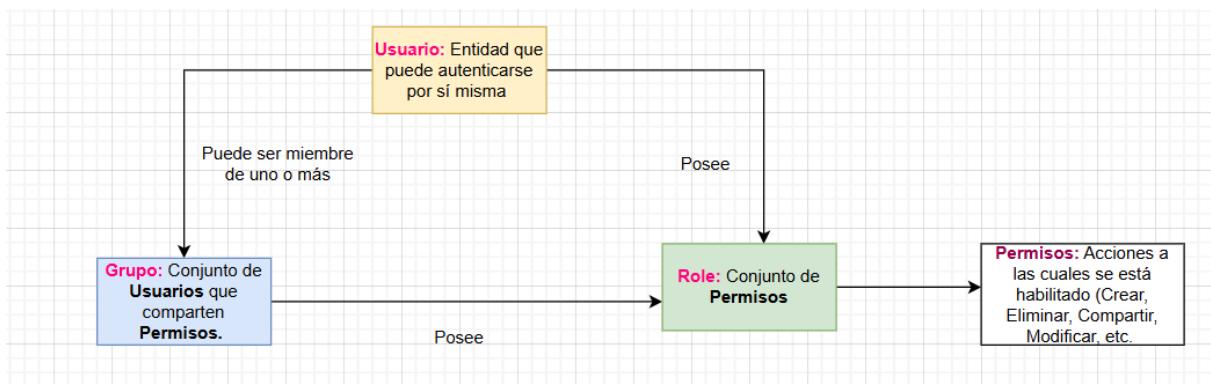
Día 3 – Roles, Usuarios y Permisos en PostgreSQL

🎯 Objetivo del día

Comprender y aplicar el **modelo de seguridad de PostgreSQL**, creando roles y usuarios, asignando permisos correctamente y entendiendo cómo proteger datos en entornos reales.

El **Usuario** se puede entender como un **ROL** con atributo de **Login**, comprendiendo que el **ROL** es una entidad de Seguridad que puede:

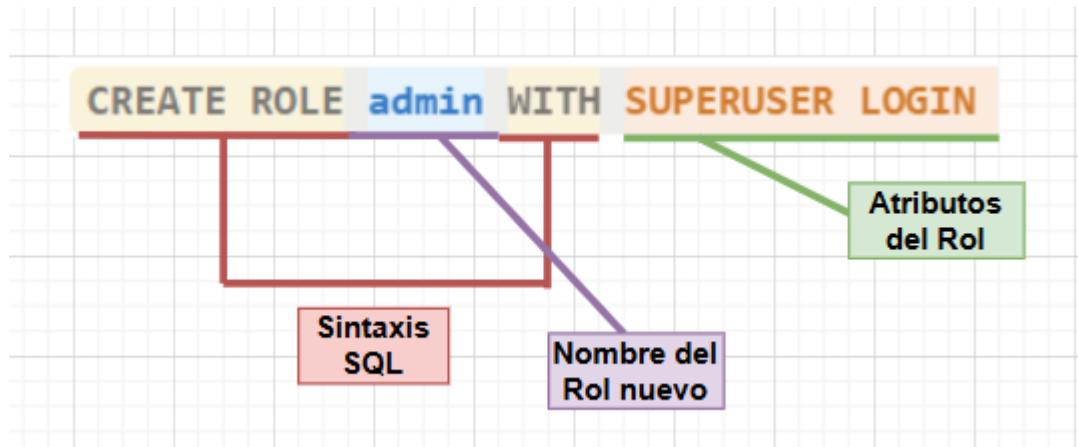
- Representar a una Persona (Usuario)
- Representar una aplicación
- Representar un conjunto de Permisos



Atributos del Rol (Role Attributes):

Los atributos de un Rol, son los que definen las capacidades administrativas de un Rol a nivel de Cluster.

CREATE ROLE admin WITH SUPERUSER LOGIN



- **SUPERUSER:** Puede sobre escribir todas las clases de acceso a las Bases de Datos, posee un acceso TOTAL al Cluster. (**Default = NOSUPERUSER**)
- **CREATEDB:** Permite la creación de Bases de Datos. (**Default = NOCREATEDB**)
- **CREATEROLE:** Permite la creación de nuevos Roles, además, posee la capacidad de eliminar o modificar permisos de otros Roles. (**Default = NOCREATEROLE**)
- **INHERIT:** Determina si el nuevo Rol hereda los permisos del grupo al cual pertenece. (**Default = INHERIT**)
- **LOGIN:** Permite que el nuevo Rol pueda hacer Login. (**Default = NOLOGIN**)

Parámetros de Conexión y Autenticación o Configuraciones del Rol: Estos determinan el comportamiento del Rol y como será su conexión.

- **CONNECTION LIMIT:** Permite configurar el límite de conexiones del Rol, si se establece como -1, signifiga que posee conexiones ilimitadas.
(`CONNECTION LIMIT -1`)

- **PASSWORD '.....'**: Le asigna una clave de acceso al Rol, para que un Rol pueda tener esta configuración de acceso, necesita poseer el Atributo de Rol **Login**.
- **VALID UNTIL 'aaa-mm-dd hh-mm-ss'**: Determina una fecha de expiración para le Password, si este parámetro no se agrega, el Password no vencerá.

Cláusulas de Gestión de Membresía: Definen la relación de los grupos.

- **IN ROLE**: El nuevo Rol será miembro de los Roles listados
- **ROLE**: Los Roles listados en la sentencia serán miembros del nuevo Rol, esta acción hace que el **nuevo Rol** sea un **GRUPO**.
- **ADMIN**: Cumple una función similar a **ROLE**, pero este le concede a los miembros el permiso de también ser administradores del **nuevo Rol**.



En PostgreSQL, los roles pueden tener **atributos administrativos** que definen sus capacidades a nivel del cluster (como **SUPERUSER**, **CREATEDB**, **CREATEROLE**, **LOGIN**) y **opciones de configuración** (como **PASSWORD**, **CONNECTION LIMIT**, **VALID UNTIL**).

Estos atributos son distintos de los **permisos sobre objetos**, como Tablas o Schemas, los cuales se gestionan mediante **GRANT** y **REVOKE**.

Los principales **Permisos** que hay en PostgreSQL:

- **SELECT**
- **INSERT**
- **UPDATE**
- **DELETE**
- **ALL**

IMPORTANTE: Los **roles** en PostgreSQL son **globales al cluster**, sus **atributos** (LOGIN, SUPERUSER, CREATEDB...) aplican a **todas las bases** y los **permisos** se otorgan por **base de datos, schemas, tablas**, o sea por **objeto** que se defina.

Conexión a la Base de Datos desde la Terminal de SQL Shell, es esta ocación me conecte a la BD creada en el día 2 de estudio Alquiler, pero si no informo la Base de Datos, por defecto se conecta a la BD Postgres

PSQL primero establece la **conexión** al **Servidor** y a la **Base de Datos**, después de eso permite ingresar comandos **SQL**.

The screenshot shows a terminal window titled "SQL Shell (psql)". It displays the following connection parameters:

- Server [localhost]: localhost
- Database [postgres]: alquiler
- Port [5432]: 5432
- Username [postgres]: postgres
- Contraseña para usuario postgres: password

Annotations explain these parameters:

- localhost: Es el Host donde corre PostgreSQL, en mi caso es en mi PC por eso es localhost
- alquiler: Base de Datos a la que deseo conectarme
- 5432: Es el Puerto donde escucha PostgreSQL
- postgres: El Rol con Login que se usa para conectarse
- password: Es el Password asignado

A callout box provides options to find the port:

Opciones para saber cual es el Puerto:
 1. pgAdmin --> Server --> Properties --> Connection
 2. Archivo postgresql.conf
 3. Durante la instalación

A second callout box provides options to find the username:

Opciones para saber el Username:
 1. pgAdmin --> Login/Group Roles
 2. Consulta con SQL:
 SELECT rolname FROM pg_roles;

```

SQL Shell (psql)
X + -
Server [localhost]: localhost Es el Host donde corre PostgreSQL, en mi caso es en mi PC por eso es localhost
Database [postgres]: alquiler Base de Datos a la que deseo conectarme
Port [5432]: 5432 Es el Puerto donde escucha PostgreSQL
Username [postgres]: postgres El Rol con Login que se usa para conectarse
Contraseña para usuario postgres: password Es el Password asignado

psql (16.11)
ADVERTENCIA: El código de página de la consola
de página de Windows (1252).
Los caracteres de 8 bits pueden fu
Vea la página de referencia de psql
para obtener más detalles.
Digite «help» para obtener ayuda.

alquiler=# |
  
```

Dato curioso: con el comando **\du** en la terminal de PostgreSQL (psql), se puede ver la lista de todos los Roles.

```

SQL Shell (psql)
postgres=> CREATE DATABASE lectura_libros;
ERROR: se ha denegado el permiso para crear la base de datos
postgres=> \du
          Lista de roles
  Nombre de rol   |           Atributos
-----+-----
    admin      | Superusuario
    +
escritura    | Contrasena vblida hasta 2025-12-31 19:30:59-05
lector       | No puede conectarse
lectura       | No puede conectarse
postgres      | Superusuario, Crear rol, Crear BD, Replicacion, Ignora
RLS
usuario_escritura | Crear rol, Crear BD
usuario_escritura2 |
usuario_lectura  |

```

Roles Unificados:

En otros sistemas (como SQL Server o el antiguo Postgres), tienes un **Usuario** para entrar y un **Grupo** para organizar permisos. En PostgreSQL, el sistema solo conoce el objeto **ROLE**.

Un rol puede actuar como usuario, como grupo, o como ambos a la vez, dependiendo de sus **atributos**:

- **Si tiene el atributo `LOGIN`**: Se comporta como un **Usuario** (puede iniciar sesión).
- **Si NO tiene el atributo `LOGIN`**: Se comporta como un **Grupo** (sirve para contener permisos y otros roles).
- **Si tiene miembros**: Se comporta como un **Grupo**.
- **Si es miembro de otro**: Se comporta como un **Usuario** dentro de un grupo.

Comando	Lo que realmente hace el sistema	Resultado Típico
<code>CREATE USER</code>	<code>CREATE ROLE ... WITH LOGIN</code>	Un rol que puede entrar a la DB.
<code>CREATE GROUP</code>	<code>CREATE ROLE ... NOLOGIN</code>	Un rol diseñado para agrupar permisos.
<code>CREATE ROLE</code>	<code>CREATE ROLE ... NOLOGIN</code> (por defecto)	La forma base y flexible.

La verdadera potencia de los roles unificados es la capacidad de crear "árboles" de permisos. Gracias a que son la misma entidad, un rol puede ser "padre" de otro mediante la cláusula `GRANT`.

- **Herencia (`INHERIT`):** Por defecto, si el rol "Juan" es miembro del rol "Ventas", Juan hereda automáticamente todos los permisos de Ventas.
- **Propiedad:** Un rol (ya sea usuario o grupo) puede ser dueño de tablas o bases de datos.

Las ventajas de este modelo son:

1. **Flexibilidad:** Puedes convertir un grupo en un usuario simplemente dándole permiso de `LOGIN`, o viceversa.
2. **Simplicidad:** Solo hay una tabla de sistema (`pg_authid`) para gestionar toda la seguridad del clúster.
3. **Recursividad:** Los grupos pueden estar dentro de otros grupos sin límites complejos, ya que técnicamente todos son la misma clase de objeto.

Autenticación vs Autorización

- **Autenticación:** *quién eres*
- **Autorización:** *qué puedes hacer*

Actividad práctica:

1 Crear roles

```

Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:

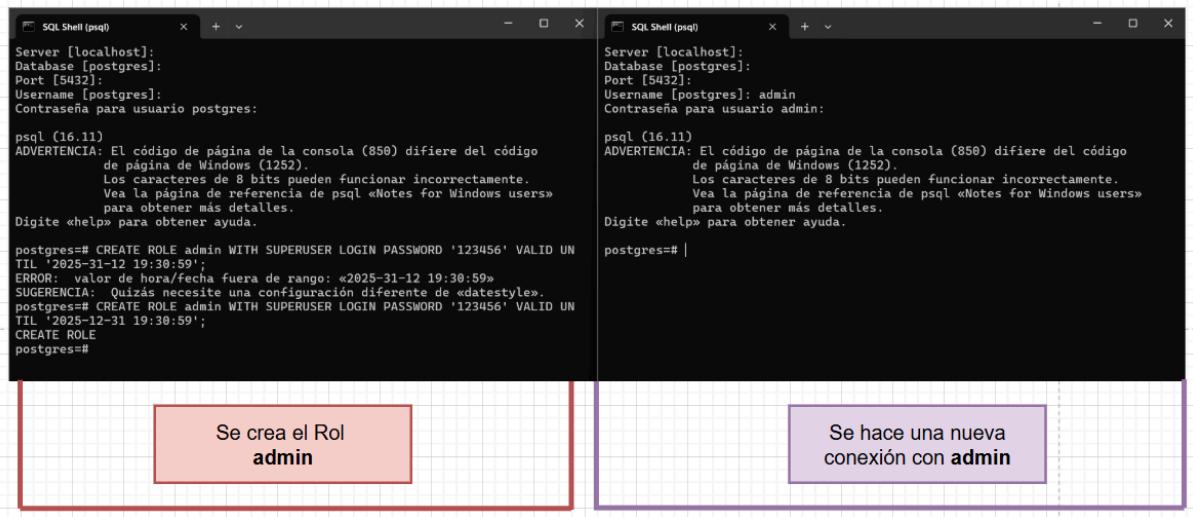
psql (16.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

postgres=# CREATE ROLE admin WITH SUPERUSER LOGIN PASSWORD '123456' VALID UNTIL '2025-31-12 19:30:59';
ERROR: valor de hora/fecha fuera de rango: «2025-31-12 19:30:59»
SUGERENCIA: Quizás necesite una configuración diferente de «datestyle».
postgres=# CREATE ROLE admin WITH SUPERUSER LOGIN PASSWORD '123456' VALID UNTIL '2025-12-31 19:30:59';
CREATE ROLE
postgres=#

```



Se crea el primer **ROL** llamado **ADMIN**, el cuál también es un **USUARIO** (porque es una entidad que posee **LOGIN**), pero al momento del su creación se presento un error de sintaxis de la fecha, ya que, PostgreSQL interpreta las fechas según datestyle, por tal motivo espera un formato **YYYY-MM-DD** y se escribió **YYYY-DD-MM**, después de hacer el cambio pertinente el Rol fue creado.



```

Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:

psql (16.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

postgres=# CREATE ROLE admin WITH SUPERUSER LOGIN PASSWORD '123456' VALID UNTIL '2025-31-12 19:30:59';
ERROR: valor de hora/fecha fuera de rango: «2025-31-12 19:30:59»
SUGERENCIA: Quizás necesite una configuración diferente de «datestyle».
postgres=# CREATE ROLE admin WITH SUPERUSER LOGIN PASSWORD '123456' VALID UNTIL '2025-12-31 19:30:59';
CREATE ROLE
postgres=#

```

Se crea el Rol
admin

Se hace una nueva
conexión con **admin**

-- Rol de solo lectura

CREATE ROLE lector;

-- Rol de escritura

CREATE ROLE escritor;

```
postgres=# CREATE ROLE lectura;  
CREATE ROLE  
postgres=# CREATE ROLE escritura;  
CREATE ROLE  
postgres=#
```

2 Crear usuarios

```
CREATE ROLE usuario_lectura  
LOGIN  
PASSWORD'789456';
```

```
CREATE ROLE usuario_escritura  
LOGIN  
PASSWORD'hola123' NOCREATEDB;
```

```
postgres=# CREATE ROLE usuario_lectura LOGIN PASSWORD '789456';  
CREATE ROLE  
postgres=# CREATE ROLE usuario_escritura LOGIN PASSWORD 'hola123' NOCREATEDB  
;  
CREATE ROLE  
postgres=# |
```

The screenshot shows two separate PostgreSQL sessions in SQL Shell (psql) windows.

Session 1 (Top Window):

```

Server [localhost];
Database [postgres];
Port [5432];
Username [postgres]: usuario_lectura
Contraseña para usuario usuario_lectura:

psql (16.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.

Digite «help» para obtener ayuda.

postgres=> CREATE DATABASE lectura.libros;
ERROR: se ha denegado el permiso para crear la base de datos
postgres=>

```

Session 2 (Bottom Window):

```

Server [localhost];
Database [postgres];
Port [5432];
Username [postgres]: usuario_escritura
Contraseña para usuario usuario_escritura:

psql (16.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.

Digite «help» para obtener ayuda.

postgres=> CREATE DATABASE escritura.libros;
ERROR: se ha denegado el permiso para crear la base de datos
postgres=>

```

Annotations:

- A purple callout box points to the error message in Session 1: "El usuario_lectura no puede crear una Base de Datos, porque en el momento en que fue creado este Rol, no se le otorgó este Atributo."
- An orange callout box points to the error message in Session 2: "El usuario_escritura no puede crear una Base de Datos, porque en el momento en que fue creado este Rol, se especificó que no poseía el Atributo de crear Bases de Datos."

--Creación de un Rol (usuario) donde se especifica los Atributos para crear Bases de Datos y Roles

CREATE ROLE usuario_escritura2

WITH CREATEDB CREATEROLE LOGIN PASSWORD '789456';

--El Rol (usuario) crea una Base de Datos y un Rol

CREATE DATABASE libros;

CREATE ROLE lector;

The screenshot shows two separate PostgreSQL sessions in SQL Shell (psql) windows.

Session 1 (Left Window):

```

Server [localhost];
Database [postgres];
Port [5432];
Username [postgres]: postgres
Contraseña para usuario postgres:

psql (16.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.

Digite «help» para obtener ayuda.

postgres=> CREATE ROLE usuario_escritura2 WITH CREATEDB CREATEROLE LOGIN PAS
SWORD '789456';
CREATE ROLE
postgres=>

```

Session 2 (Right Window):

```

Server [localhost];
Database [postgres];
Port [5432];
Username [postgres]: usuario_lectura
Contraseña para usuario usuario_lectura:

psql (16.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.

Digite «help» para obtener ayuda.

postgres=> CREATE DATABASE lectura.libros;
ERROR: se ha denegado el permiso para crear la base de datos
postgres=>

```

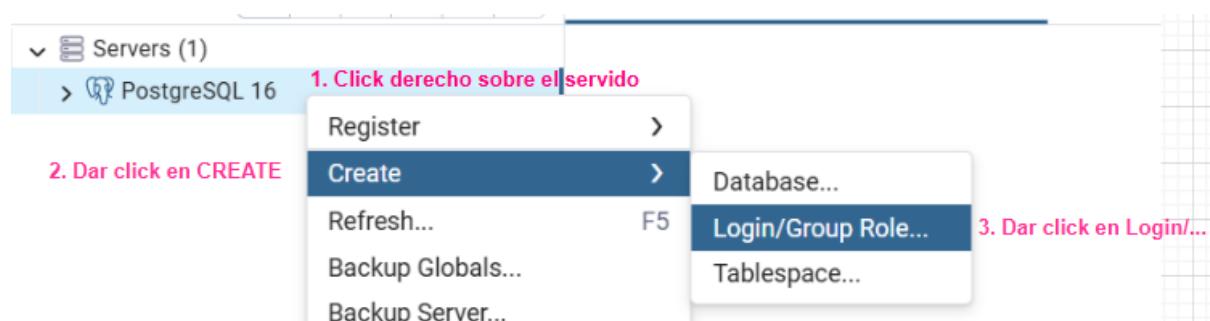
Annotations:

- A red box highlights the command "CREATE ROLE usuario_escritura2 WITH CREATEDB CREATEROLE LOGIN PAS SWORD '789456';" in Session 1.
- A green box highlights the error message in Session 2: "El usuario_escritura2, logró crear una Base de Datos y un Rol".

Creación de Roles y Usuarios por medio de la herramienta pgAdmin:

Desde la herramienta de pgAdmin también se pueden crear los Roles y Usuarios, esta es una opción más amigable pues a diferencia de la Terminal está posee una interfaz gráfica.

1. Ubicarse sobre el Servidor
2. Dar clik derecho sobre el Servidor
3. Seleccionar la opción de Create
4. Elegir la opción de Login/Group Role...



Posteriormente se abrirá una ventana, donde se debe:

5. En la pestaña **General** definir el nombre y agregar un comentario (opcional)

The screenshot shows the 'Group Role - PostgreSQL 16' configuration window. The tab bar at the top has 'General' selected, which is underlined. Below the tabs, there are two input fields: 'Name' with the value 'administrador_test' and 'Comments' with the value 'Crea el usuario de administrador por motivo de práctica |'. The 'Comments' field is currently active, indicated by a cursor in the text area.

6. Definir las configuraciones de acceso

En el ejemplo se agrega un Password, se determina una fecha de vencimiento para la contraseña y se le permite al usuario tener acceso ilimitado

 Group Role - PostgreSQL 16 X

General **Definition** Privileges Membership Parameters Security SQL

Password
Account expires	2025-12-31 13:00:00 -05:00 
Please note that if you leave this field blank, then the password will never expire.	
Connection limit	-1

7. Configurar los Privilegios que posee el usuario

 Group Role - PostgreSQL 16

General Definition **Privileges** Membership Parameters Security SQL

Can login?	<input checked="" type="checkbox"/>
Superuser?	<input type="checkbox"/>
Create roles?	<input type="checkbox"/>
Create databases?	<input checked="" type="checkbox"/>
Inherit rights from the parent roles?	<input checked="" type="checkbox"/>
Can initiate streaming replication and backups?	<input type="checkbox"/>
Bypass RLS?	<input type="checkbox"/>

8. Guardar la configuración

 Close  Reset  Save

3 Asignar roles a usuarios

```
GRANT lectorTO usuario_lectura;  
GRANT escritorTO usuario_escritura;
```

Al intentar asignar un Rol al **usuario_escritura2** me sale un error, pues se estaba haciendo desde el **usuario_lectura** y este no cuenta con el privilegio de **CREATEROLE**

```
postgres=> GRANT admin TO usuario_escritura2;  
ERROR: se ha denegado el permiso para otorgar el rol «admin»  
DETALLE: Sólo los roles con el atributo SUPERUSER pueden otorgar roles con  
el atributo SUPERUSER.  
postgres=> |
```

Pero al hacerlo desde el usuario **postgres** si me lo permitió, porque este es un **SUPERUSER** y cuenta con el Privilegio de **CREATEROLE**

```
postgres=# GRANT admin TO usuario_escritura2;  
GRANT ROLE  
postgres=# |
```

Dato curioso:

En psql, el prompt **-#** indica que una instrucción está incompleta; esta se debe cancelar con **Ctrl+C** o completar la sintaxis antes de ejecutar nuevos comandos.

Prompt	Significado
=# postgres=#	Listo para ejecutar comandos
-# postgres-#	Esperando que completes la instrucción
postgres=# SELECT rolname, rolinherit FROM pg _roles WHERE rolname = 'usuario_escritura' postgres-#	Falta el ';' de la consulta

```

postgres=# SELECT rolname, rolinherit FROM pg
_roles WHERE rolname = 'usuario_escritura'
postgres-# ;
      rolname      | rolinherit
-----+-----
 usuario_escritura | t
(1 fila)

postgres=#

```

Se ingresa la consulta sin ;

Solicita que ingrese el ; que hace falta

Permite otra Solicitud

Checklist:

- Estudio del tema
- Ejecución de scripts
- Evidencia guardada
- Notas técnicas escritas

Recursos:

- <https://www.postgresql.org/docs/current/user-manag.html>
- https://youtu.be/JAxgsKy8Q0k?si=o9_UaAieH5DfpoxD
- https://youtu.be/ojEYsse-d2M?si=j0_uYPnmISYPIrl
-