

Creación Base de Datos

María Ximena Marin Delgado

Emilia Gallo Alzate

Sena

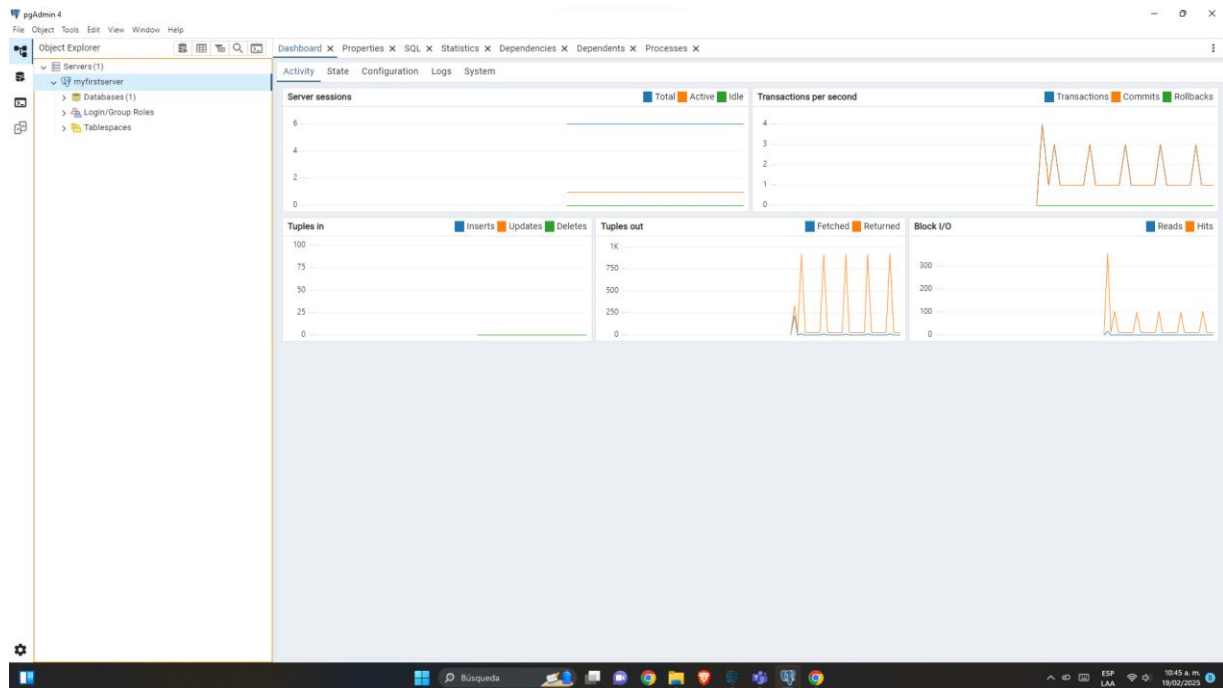
Analisis y Desarrollo de Software: Bases de Datos

Ingeniero Yusef Gonzales

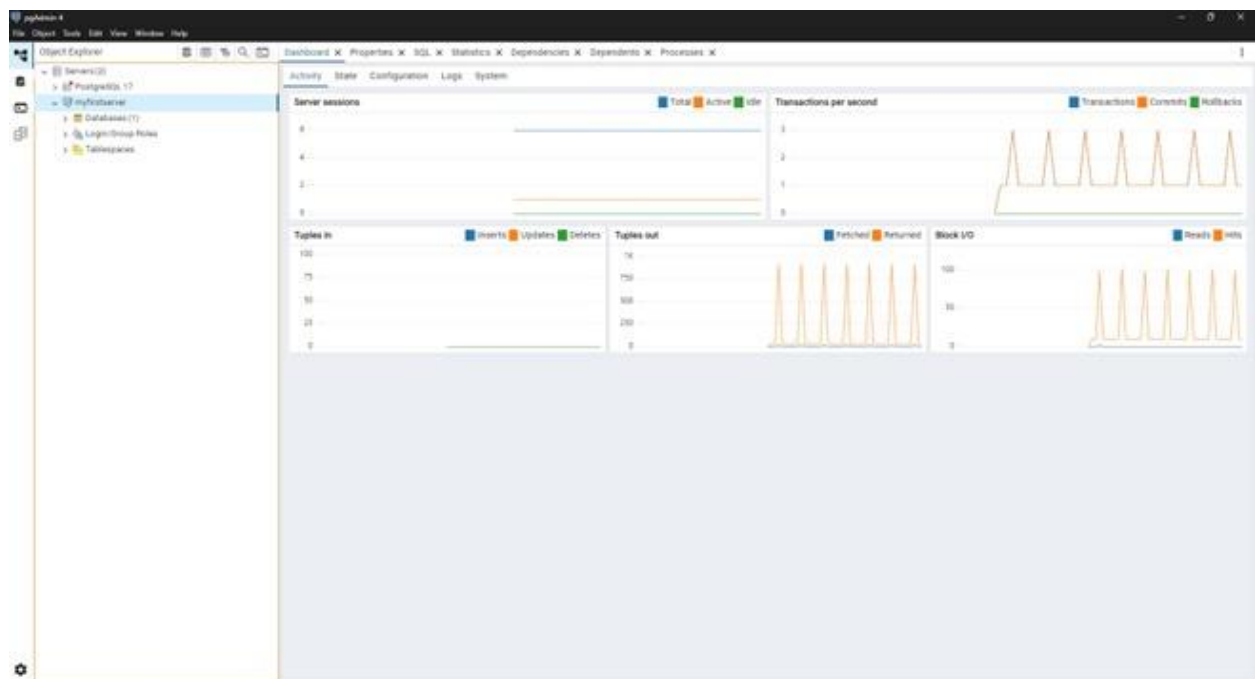
Itagüí, Antioquia

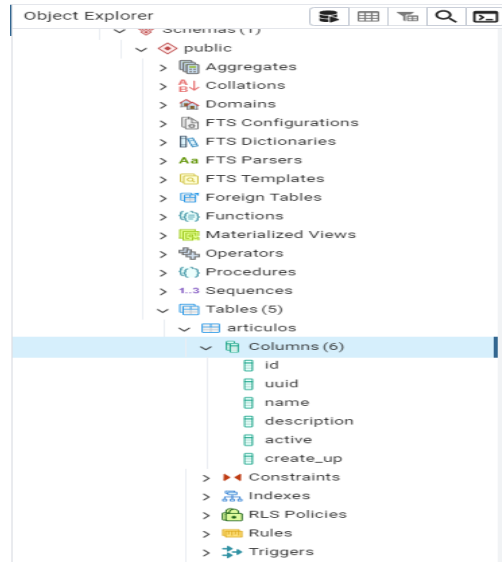
Marzo 7 de 2025

Primer base de datos (Emilia Gallo 19/02/2025):



Primer base de datos (María Ximena Marin Delgado19/02/2025):





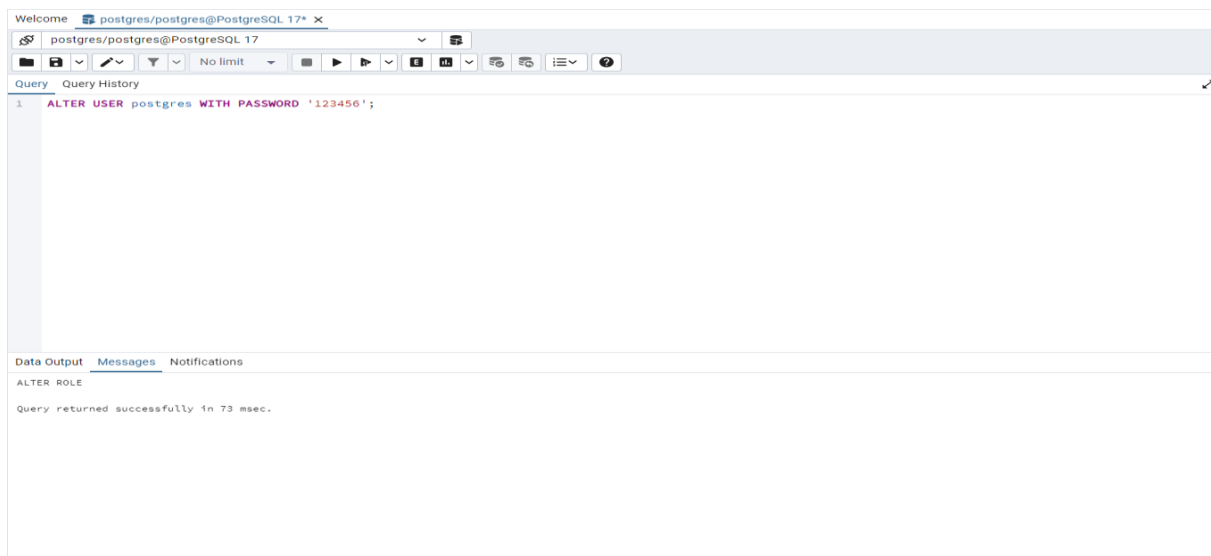
26/02/2025

Modificar la contraseña

```
postgres=# SELECT current_user;
current_user
-----
postgres
(1 fila)

postgres=# ALTER USER postgres WITH PASSWORD '123456';
```

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (17.2)
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.
```



Observar todos los usuarios configurados

```
SQL Shell (psql)

postgres=# SELECT rolname FROM pg_roles;
          rolname
-----
pg_database_owner
pg_read_all_data
pg_write_all_data
pg_monitor
pg_read_all_settings
pg_read_all_stats
pg_stat_scan_tables
pg_read_server_files
pg_write_server_files
pg_execute_server_program
pg_signal_backend
pg_checkpoint
pg_maintain
pg_use_reserved_connections
pg_create_subscription
postgres
(16 filas)
```

Welcome postgres/postgres@PostgreSQL 17* X

postgres/postgres@PostgreSQL 17

Query Query History

```
1 SELECT rolname FROM pg_roles;
```

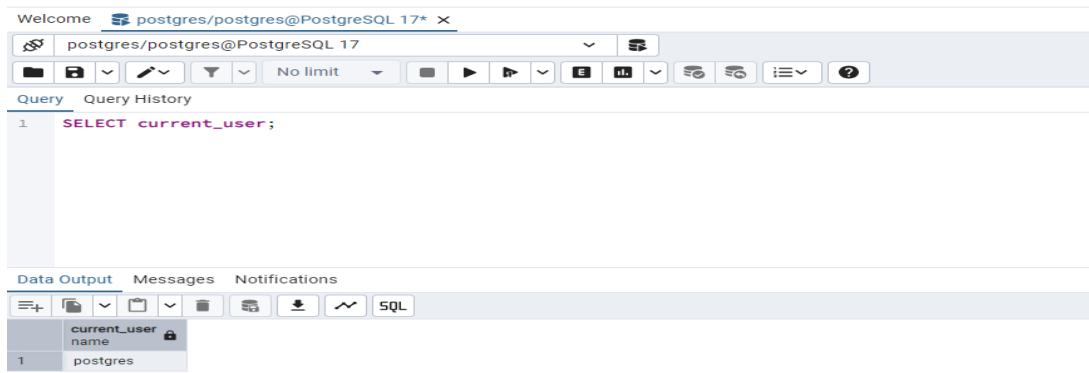
Data Output Messages Notifications

	rolname name	
1	pg_database_owner	
2	pg_read_all_data	
3	pg_write_all_data	
4	pg_monitor	
5	pg_read_all_settings	
6	pg_read_all_stats	
7	pg_stat_scan_tables	
8	pg_read_server_files	
9	pg_write_server_files	
10	pg_execute_server_program	
11	pg_signal_backend	
12	pg_checkpoint	
13	pg_maintain	
14	pg_use_reserved_connections	
15	pg_create_subscription	
16	postgres	

Seleccionar el usuario actual

```
SQL Shell (psql)

postgres=# SELECT current_user;
 current_user 
-----
 postgres
(1 fila)
```



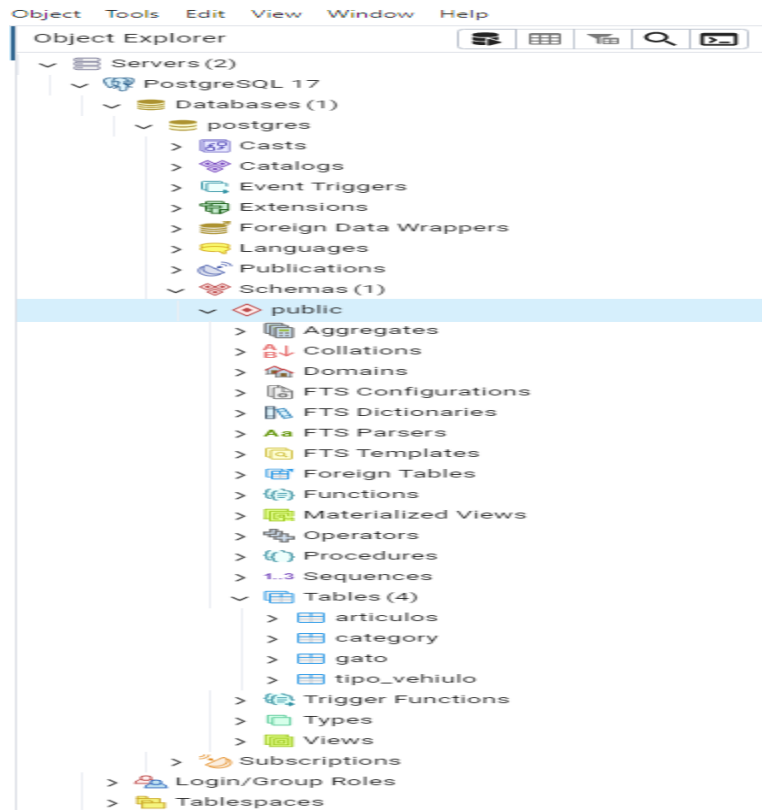
Obtener todas las tablas y sus shemas en la consola

```
SQL Shell (psql)

postgres=# SELECT current_user;
 current_user
-----
 postgres
(1 fila)

postgres=# \dt
Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
 public | articulos | tabla | postgres
(1 fila)

postgres=#
```



Crear una tabla

```
postgres=# CREATE TABLE category (id SERIAL PRIMARY KEY,clave VARCHAR(100) NOT NULL,lastname VARCHAR(100) UNIQUE NOT NULL);
postgres=#
```

```
postgres=# \dt
          Listado de relaciones
Esquema | Nombre   | Tipo | Dueño
-----+-----+-----+-----
public  | articulos | tabla | postgres
public  | category | tabla | postgres
(2 filas)
```

Welcome postgres/postgres@PostgreSQL 17* x

postgres/postgres@PostgreSQL 17

Query Query History

```
1 CREATE TABLE tipo_vehiculo (  
2     id SERIAL PRIMARY KEY,  
3     tipo VARCHAR(100) NOT NULL,  
4     vehiculo VARCHAR(100) UNIQUE NOT NULL  
5 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 32 msec.

Obtener todas las tablas y sus shemas v2

pgAdmin 4

File Object Tools Edit View Window Help

Welcome postgres/postgres@PostgreSQL 17* x postgres/postgres... x

postgres/postgres@PostgreSQL 17

Query Query History

```
1 SELECT table_schema, table_name  
2 FROM information_schema.tables  
3 WHERE table_type = 'BASE TABLE' AND table_schema NOT IN ('pg_catalog', 'information_schema');
```

Data Output Messages Notifications

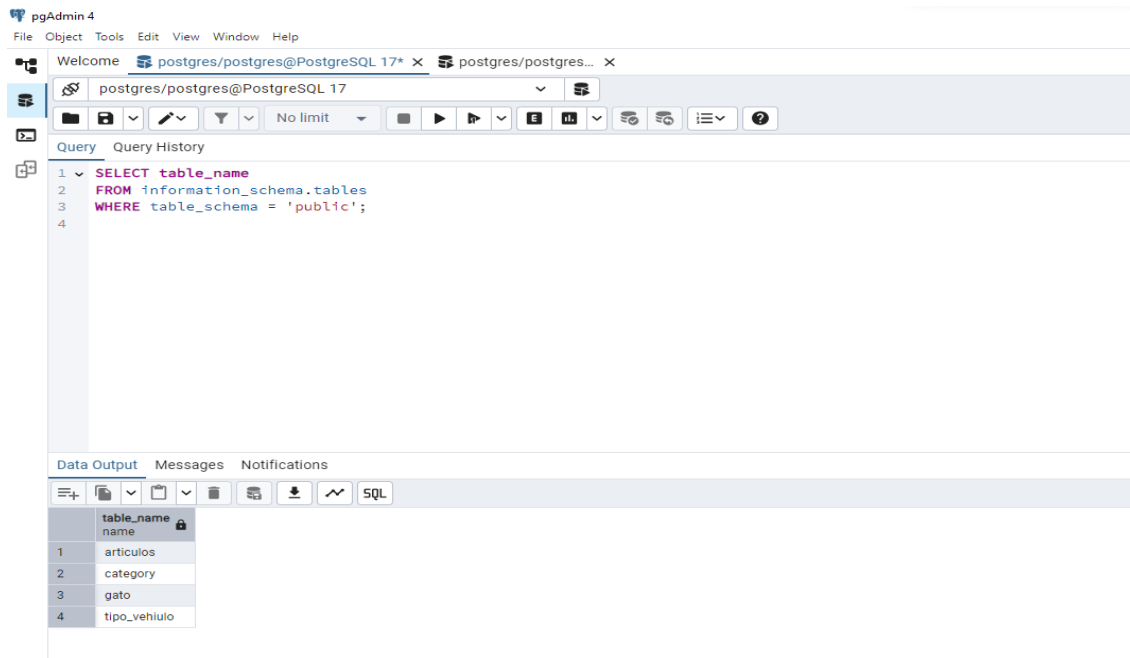
	table_schema name	table_name name
1	public	articulos
2	public	category
3	public	gato
4	public	tipo_vehiculo

Obtener el nombre de la tabla

```
SQL Shell (psql)

postgres=# FROM information_schema.tables
postgres=# WHERE table_schema = 'public';
  table_name
-----
 articulos
 category
      gato
 tipo_vehiculo
(4 filas)

postgres=#
```



Obtener todo el contenido de la tabla (only console)

```
postgres=# \d nombre_de_la_tabla
No se encontró relación llamada «nombre_de_la_tabla».
postgres=# \d gato
                                Tabla %public.gato%
Columna |          Tipo          | Ordenamiento | Nulable | Por omisi%
-----+-----+-----+-----+-----
id      | integer                |              | not null | nextval('gato_id_seq'::regclass)
raza    | character varying(100) |              | not null | 
edad    | character varying(100) |              | not null | 
Indices:
    "gato_pkey" PRIMARY KEY, btree (id)
    "gato_edad_key" UNIQUE CONSTRAINT, btree (edad)

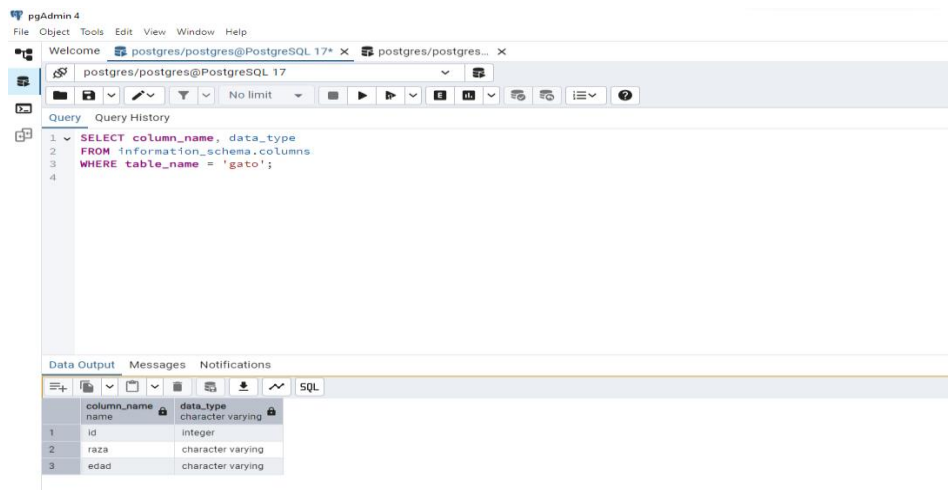
postgres=#
```

```
postgres=# \d tipo_vehiculo
No se encontró relación llamada «tipo_vehiculo».
postgres=# \d tipo_vehiculo
                                Tabla %public.tipo_vehiculo%
Columna |          Tipo          | Ordenamiento | Nulable | Por omisi%
-----+-----+-----+-----+-----
id      | integer                |              | not null | nextval('tipo_vehiculo_id_seq'::regclass)
tipo    | character varying(100) |              | not null | 
vehiculo | character varying(100) |              | not null | 
Indices:
    "tipo_vehiculo_pkey" PRIMARY KEY, btree (id)
    "tipo_vehiculo_vehiculo_key" UNIQUE CONSTRAINT, btree (vehiculo)

postgres=#
```

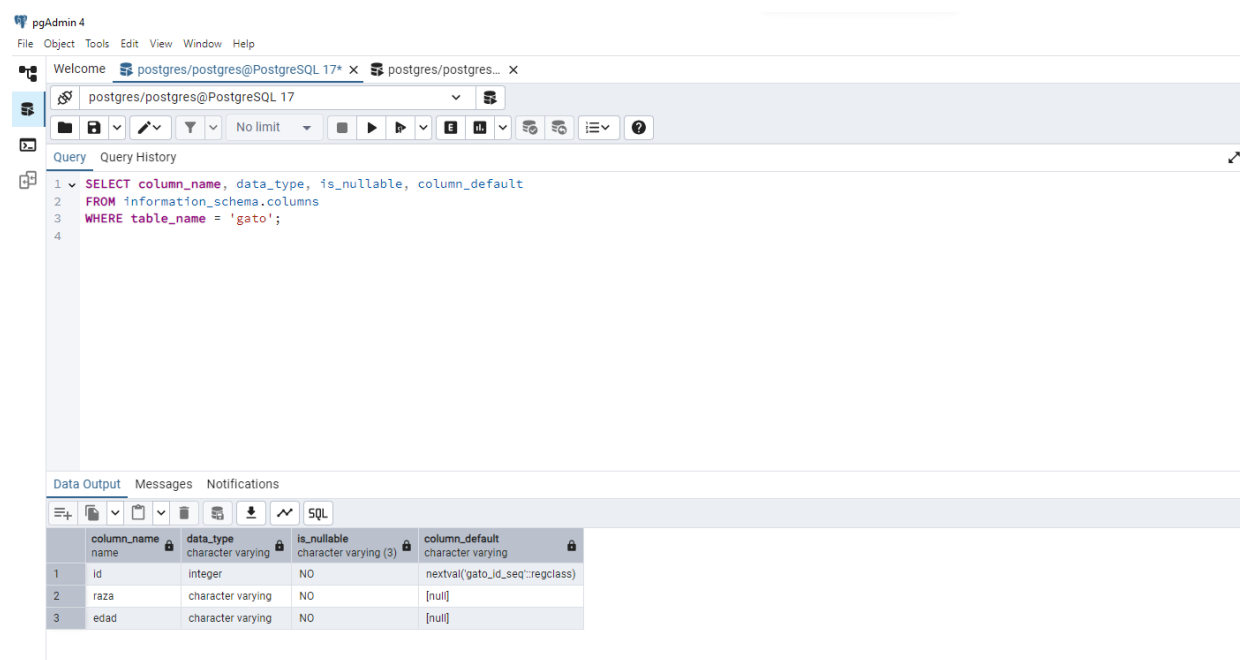
Obtener todo el contenido de la tabla sencillo

```
postgres=# SELECT column_name, data_type
postgres=# FROM information_schema.columns
postgres=# WHERE table_name = 'gato';
 column_name | data_type
-----+-----
id           | integer
raza         | character varying
edad         | character varying
(3 filas)
```



Obtener todo el contenido de la tabla más detallado

```
postgres=# SELECT column_name, data_type, is_nullable, column_default
postgres=# FROM information_schema.columns
postgres=# WHERE table_name = 'gato';
 column_name |      data_type      | is_nullable |      column_default
-----+-----+-----+-----
id           | integer             | NO          | nextval('gato_id_seq'::regclass)
raza         | character varying   | NO          |
edad         | character varying   | NO          |
(3 filas)
```



Crear una tabla (chatgpt)

Welcome postgres/postgres@PostgreSQL 17* x postgres/postgres... x

postgres/postgres@PostgreSQL 17

Query Query History

```
1 CREATE TABLE tipos_de_datos (  
2   id SERIAL PRIMARY KEY,  
3   nombre VARCHAR(100) NOT NULL,      -- Tipo de dato: character varying  
4   descripción TEXT,                  -- Tipo de dato: text  
5   edad INTEGER,                      -- Tipo de dato: integer  
6   salario NUMERIC(10, 2),            -- Tipo de dato: numeric  
7   activo BOOLEAN,                   -- Tipo de dato: boolean  
8   fecha_nacimiento DATE,             -- Tipo de dato: date  
9   ultima_conexion TIMESTAMP WITH TIME ZONE, -- Tipo de dato: timestamp with time zone  
10  direccion BYTEA,                   -- Tipo de dato: bytea  
11  datos_json JSON,                   -- Tipo de dato: json  
12  cantidad DECIMAL(10, 2),           -- Tipo de dato: decimal  
13  producto_id UUID,                  -- Tipo de dato: uuid  
14  imagen_url VARCHAR(255),            -- Tipo de dato: character varying  
15  geo_ubicacion POINT,               -- Tipo de dato: point  
16 );  
17
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 59 msec.

Servers (2)

PostgreSQL 17

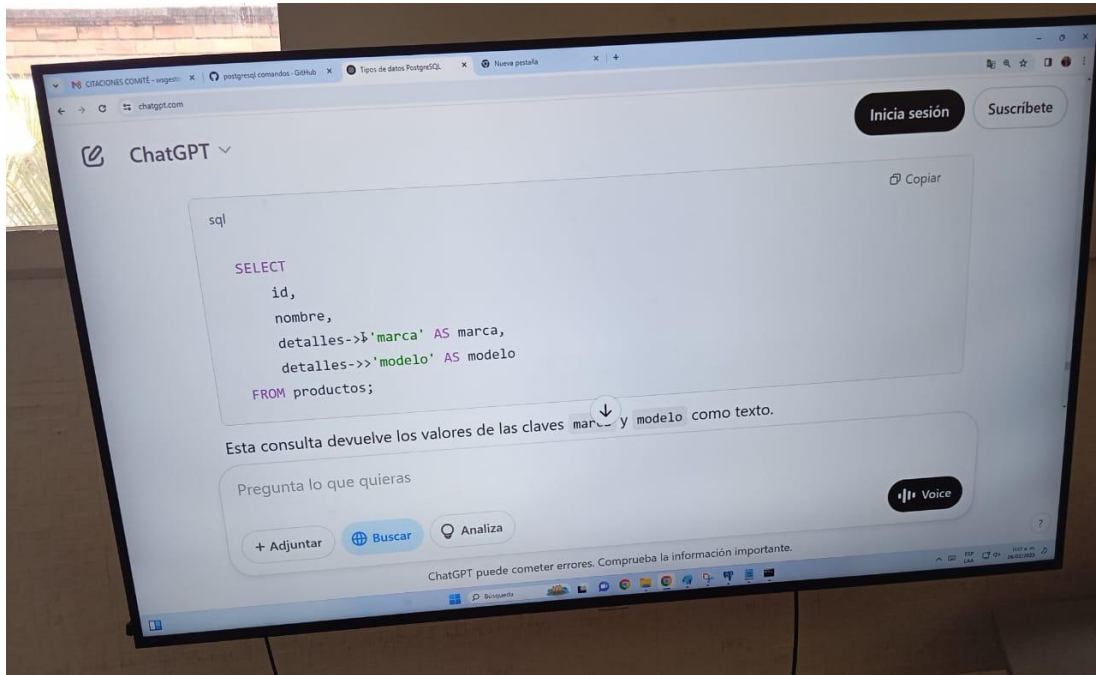
Databases (1)

postgres

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (1)
 - public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - Tables (5)
 - > articulos
 - > category
 - > gato
 - > tipo_vehiculo
 - > tipos_de_datos
 - > Trigger Functions
 - > Types
 - > Views
 - > Subscriptions
 - > Login/Group Roles
 - > Tablespaces

Tipo de dato (jsonb)

Exposición Emilia Gallo Alzate(4.5):



Tipo de dato (money)

Exposición María Ximena Marín Delgado(5):

