

## Actividad | 2 | Gestor de Bases de Datos

### Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez Vega

ALUMNO: Christopher Dávila Rojas

FECHA: 03/09/2024

## Índice

<b>Introducción</b>	<b>1-2</b>
<b>Descripción</b>	<b>2-3</b>
<b>Justificación</b>	<b>3-4</b>
<b>Desarrollo</b>	<b>4-13</b>
<b>Repasando lo aprendido</b>	<b>4</b>
<b>Iniciando la base datos</b>	<b>5</b>
<b>Creación de tablas y tipos de datos</b>	<b>9</b>
<b>Conclusión</b>	<b>13-14</b>
<b>Referencias</b>	<b>14</b>

## **Introducción**

En esta actividad, abordaremos la creación de una base de datos en PostgreSQL para una tienda departamental que requiere un sistema eficiente para administrar su información.

PostgreSQL es un sistema de gestión de bases de datos que se destaca por su sencillez, escalabilidad y cumplimiento de estándares, lo que me pareció en una opción ideal para manejar la complejidad y el volumen de datos que maneja una tienda departamental.

La creación de una base de datos en PostgreSQL implica estructurar y organizar los datos de manera que se facilite su acceso, manipulación y protección, garantizando así que la tienda pueda operar de manera eficiente y segura.

### **Descripción**

La naturaleza de una tienda de este tipo implica manejar una gran cantidad de datos relacionados con productos, clientes, ventas, inventarios y proveedores. Hasta ahora, es posible que la tienda haya estado utilizando métodos menos eficientes o manuales para gestionar esta información, lo cual no solo es propenso a errores, sino que también limita la capacidad de la tienda para crecer y adaptarse a nuevas necesidades del mercado.

Por lo tanto, se ha planteado la necesidad de desarrollar una base de datos utilizando PostgreSQL, un sistema conocido por su fiabilidad y versatilidad. En esta actividad, se espera interpretar los requerimientos específicos de la tienda, como la gestión de productos y ventas, para luego traducir estos requerimientos en una estructura de base de datos relacional.

Esta estructura permitirá no solo la centralización de la información, sino también su manipulación eficiente y segura, lo que es crucial para el éxito operativo de la tienda.

### **Justificación**

La elección de PostgreSQL como el sistema de gestión de bases de datos para esta actividad está justificada por varias razones clave.

En primer lugar, PostgreSQL es un sistema de código abierto, lo que significa que es accesible sin costo alguno y cuenta con una gran comunidad de desarrolladores que contribuyen a su mejora continua.

Además, es conocido por su capacidad para manejar grandes volúmenes de datos y por su compatibilidad con múltiples plataformas, lo que lo hace ideal para una tienda departamental que puede estar creciendo o planeando expandirse.

Otra ventaja significativa es la seguridad que ofrece; con funciones avanzadas de control de acceso y protección de datos, garantiza que la información sensible de la tienda esté bien protegida y admite la creación de bases de datos relacionales complejas que pueden satisfacer las diversas necesidades de la tienda, desde la gestión de inventarios hasta el seguimiento de las ventas y la fidelización de los clientes.

## Desarrollo

### Repasando lo aprendido

Para el desarrollo de esta actividad, se retomó las bases de datos normalizadas del ejercicio anterior, que serían las siguientes.

Primera tabla:

Base de datos : Tienda/ Clientes			
ID_cliente	Nombre_cliente	Direccion_cliente	Telefono_cliente

Segunda tabla:

Base de datos : Tienda/ Producto			
ID_producto	Nombre_producto	Precio_producto	Marca_producto

Tercera tabla:

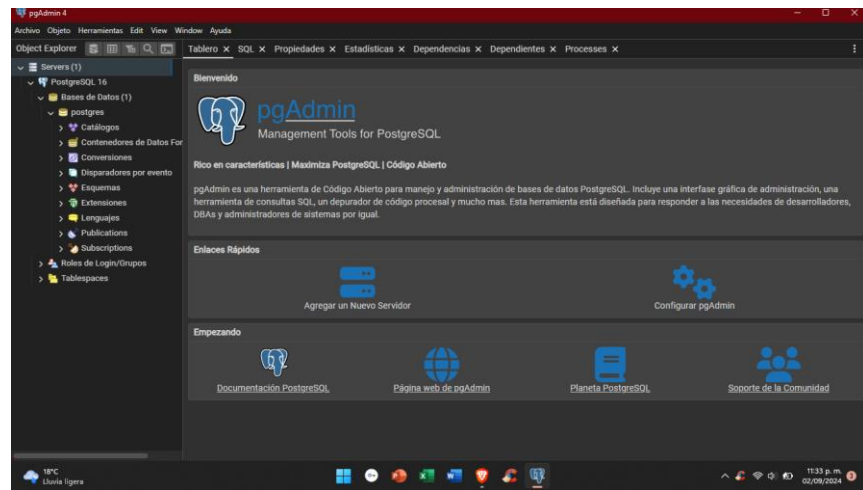
Base de datos : Tienda/ Venta	
ID_venta	Fecha_venta

Cuarta tabla:

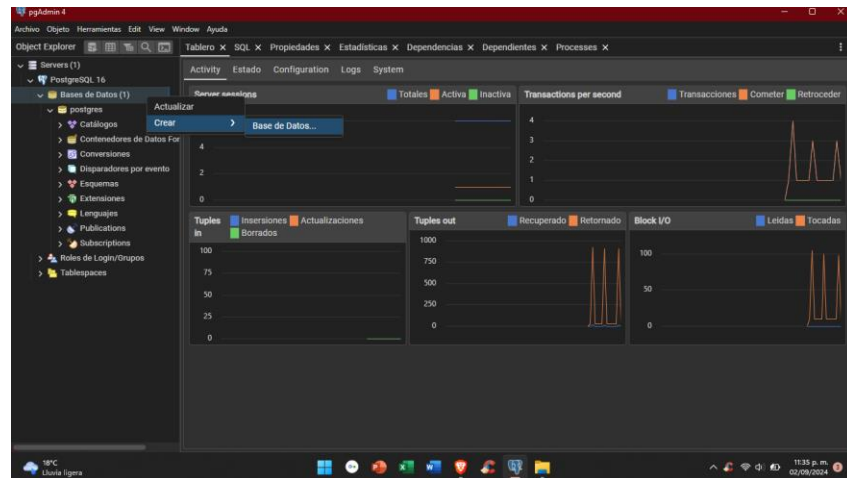
Base de datos : Tienda/ Marca	
ID_marca	Marca_producto

## Iniciando la base de datos

Con esta información como base, iniciamos abriendo PostgreSQL:



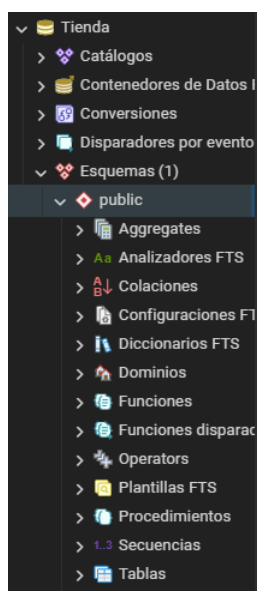
Una vez abierto, vamos a crear nuestra base de datos, para ello seleccionamos el apartado de base de datos ubicado en el menú de objetos, con el cursor seleccionando bases de datos, damos click derecho y elegimos la opción crear, para posteriormente dar click en base de datos.





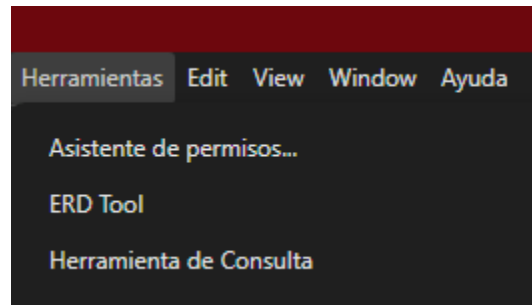


Ahora, seleccionaremos el apartado esquemas y nos mostrará la siguiente información:

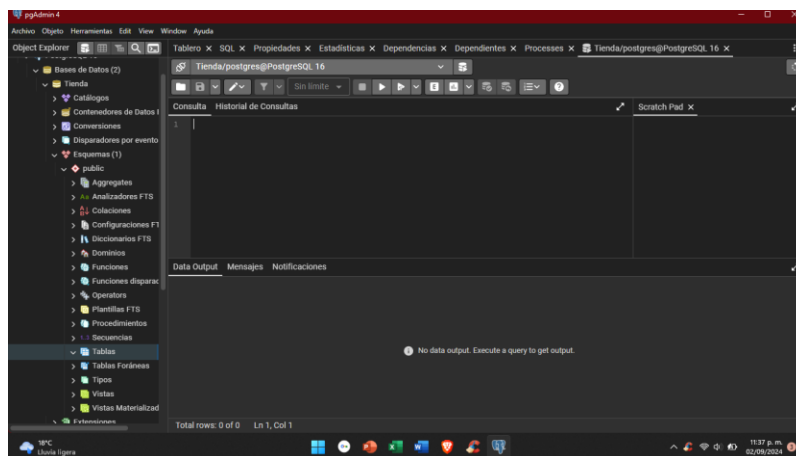


El apartado que nos interesa de este espacio es “Tablas”.

Con el espacio tablas seleccionado, iremos a herramientas daremos click y seleccionaremos herramienta de consulta.



Con ello, nos abrirá la consola.



## Creación de tablas y tipos de datos

Ahora, vamos a iniciar con la creación de las tablas. Para ello, nos valdremos del comando CREATE TABLE. Este comando nos permite crear las tablas, darles un nombre y definir las pautas que seguirá nuestra base de datos.

Para las pautas, ocuparemos los operadores lógicos: VARCHAR el cuál nos permite ingresar texto y números, INTEGER y BIGINT para caracteres puramente números de menor y mayor longitud respectivamente, DECIMAL para cantidades y DATE para fechas.

Los códigos para cada tabla quedaron de la siguiente forma.

Primera tabla:

```
CREATE TABLE clientes (  
  
  id_cliente INTEGER,  
  nombre_cliente VARCHAR (20),  
  direccion_cliente VARCHAR (60),  
  telefono_cliente BIGINT  
  
);
```

Segunda tabla:

```
10 CREATE TABLE producto (  
11  
12   id_producto INTEGER,  
13   nombre_producto VARCHAR (20),  
14   precio_producto DECIMAL (8,2),  
15   marca_producto VARCHAR (20)  
16  
17 );
```

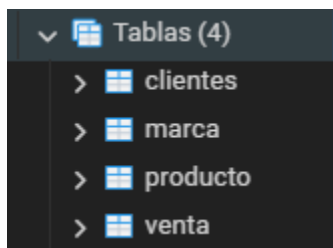
Tercera tabla:

```
19 CREATE TABLE venta (  
20  
21   id_producto INTEGER,  
22   fecha_venta DATE  
23  
24 );  
25
```

Cuarta tabla:

```
26 CREATE TABLE marca (  
27  
28   id_marca INTEGER,  
29   marca_producto VARCHAR (20)  
30  
31 );
```

Una vez creadas con la tecla F5, actualizaremos el espacio tablas, y podremos ver que se han creado de forma exitosa:



Una vez creadas las tablas, procederemos a agregar información a los registros. Esto lo conseguiremos por medio del comando INSERT INTO, seguido de los atributos que queramos agregar según la tabla y posteriormente, VALUES, seguido por los registros.

Por ejemplo, en el caso de la tabla “clientes”, el código quedó del siguiente modo:

```
4 INSERT INTO clientes (id_cliente, nombre_cliente, direccion_cliente, telefono_cliente)
5 VALUES (1501, 'Alondra Cruz Perez', 'Dalia 15 Los Reyes Ixtacala', 5571424748),
6 (1502, 'Alejandra Tellez', 'Sacerdotes 5, Ecatepec', 5535760059),
7 (1503, 'Emma Hernandez', 'Cofradia 7, Cuatitlan Izcalli', 5545760059),
8 (1504, 'Raul Ojeda Nuñez', 'Benito Juarez 25, Atizapan', 5520422010),
9 (1505, 'Alejandro Fonseca', 'Arboles 5, Cuautemoc', 5519758021)
```

Aquí podemos apreciar la estructura del código. Recordemos que en id\_cliente solo se aceptarían números, ya que elegimos que la columna fuera tipo INTEGER, mientras que en nombre y dirección, se aceptan datos alfanuméricos. Por último, en teléfono de cliente se uso el tipo de dato BIGINT, ya que al ser números telefónicos, el sistema podría detectarlos como números demasiado grandes para ser válidos en INTEGER.

Una vez hecho esto, por medio de SELECT\*FROM, vamos a seleccionar nuestra tabla, para ver que la información este correctamente insertada. Posteriormente daremos F5 para indicarle a la consola que deseamos ver esa parte. El resultado es el siguiente:

	id_cliente integer	nombre_cliente character varying (20)	direccion_cliente character varying (60)	telefono_cliente bigint
1	1501	Alondra Cruz Perez	Dalia 15 Los Reyes Ixtacala	5571424748
2	1502	Alejandra Tellez	Sacerdotes 5, Ecatepec	5535760059
3	1503	Emma Hernandez	Cofradia 7, Cuatitlan Izcalli	5545760059
4	1504	Raul Ojeda Nuñez	Benito Juarez 25, Atizapan	5520422010
5	1505	Alejandro Fonseca	Arboles 5, Cuautemoc	5519758021

Esto mismo aplica para todas las demás tablas, tal como marca, producto o venta, solo hay que considerar que tipo de dato es el que hemos elegido y, siguiendo la misma estructura, ir añadiendo los registros necesarios.

A continuación, mostraré el código y resultado de cada tabla.

Código y tabla “producto”:

```

4  INSERT INTO marca (id_marca, marca_producto)
5  VALUES
6  (1, 'Movistar'),
7  (2, 'ComputeX'),
8  (3, 'TabMaster'),
9  (4, 'SoundPro'),
10 (5, 'WristTech');
11

```

	id_marca integer	marca_producto character varying (20)
1	1	Movistar
2	2	ComputeX
3	3	TabMaster
4	4	SoundPro
5	5	WristTech

Código y tabla “venta”:

```

INSERT INTO venta (id_producto, fecha_venta)
VALUES
(1250, '2024-01-15'),
(2045, '2024-02-20'),
(3023, '2024-03-25'),
(4789, '2024-04-30'),
(5190, '2024-05-10'),
(1250, '2024-06-15'),
(3023, '2024-07-20'),
(2045, '2024-08-25');

```

	id_producto integer	fecha_venta date
1	1250	2024-01-15
2	2045	2024-02-20
3	3023	2024-03-25
4	4789	2024-04-30
5	5190	2024-05-10

Código y tabla “marca”:

```
4 INSERT INTO marca (id_marca, marca_producto)
5 VALUES
6 (1, 'Movistar'),
7 (2, 'ComputeX'),
8 (3, 'TabMaster'),
9 (4, 'SoundPro'),
10 (5, 'WristTech');
```

	id_marca integer	marca_producto character varying (20)
1	1	Movistar
2	2	ComputeX
3	3	TabMaster
4	4	SoundPro
5	5	WristTech

## **Conclusión**

Para mí, la actividad subraya la importancia de contar con un sistema de gestión de datos bien diseñado y eficaz.

En el ámbito laboral, la capacidad de organizar y manejar grandes volúmenes de información de manera eficiente es crucial para el éxito de cualquier empresa, especialmente en un entorno competitivo como el del comercio minorista.

A través de este ejercicio, no solo adquirí habilidades técnicas relacionadas con PostgreSQL, sino que también comprendí cómo estas habilidades pueden ser aplicadas para resolver problemas del mundo real.

La base de datos resultante no solo mejorará la operación diaria de la tienda, sino que también facilitará la toma de decisiones basada en datos, lo que es esencial para cualquier negocio que aspire a crecer y adaptarse a un mercado en constante cambio, así mismo, para multitud de trabajos que uno aspire a tener, incluso siendo útil para emprendimientos y/o proyectos personales.

En la vida cotidiana, el aprendizaje de estas técnicas refuerza la importancia de la organización y la eficiencia, valores que son aplicables en múltiples contextos más allá del entorno laboral.



## Referencias

*W3Schools.com*. (n.d.). [https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)