

## Introducción

En este trabajo se desarrollará la práctica dos del laboratorio de bases de datos, en la cual veremos a detalle el manejo y la administración de una base de datos por medio de herramientas de modelado, manipulación y configuración de un servidor PostgreSQL en el entorno local del alumno, se especificará actividad por actividad acompañada de una captura de pantalla o fotografía del resultado de cada una de estas y en algunos casos se acompañará con un poco de teoría complementaria y únicamente cuando se requiera se añadirá un diagrama de entidad relación.

Por último se practicará con Docker para la ejecución de comandos permitiéndonos un mejor manejo de este, reforzando y ampliando nuestros conocimientos con esta herramienta.

## Objetivos

- **Realizar las configuraciones iniciales para acceder al servidor de base de datos PostgreSQL del Laboratorio de la Facultad de Ingeniería.**
- **Instalar/Configurar el sistema operativo como punto inicial para contar con un ambiente de base de datos en los equipos de los alumnos.**

## Contenido

- **C1.-Captura de pantalla de la ejecución del comando psql --version en el entorno local del alumno.**

```
t pipardo@pipardo-Predator-PHN16-71:~$ psql --version
psql (PostgreSQL) 16.11 (Ubuntu 16.11-0ubuntu0.24.04.1)
```

- **C2.-Sentencia para la creación del esquema, captura de pantalla de la ejecución y respuesta a ¿qué es un esquema y cuáles son sus principales usos?**

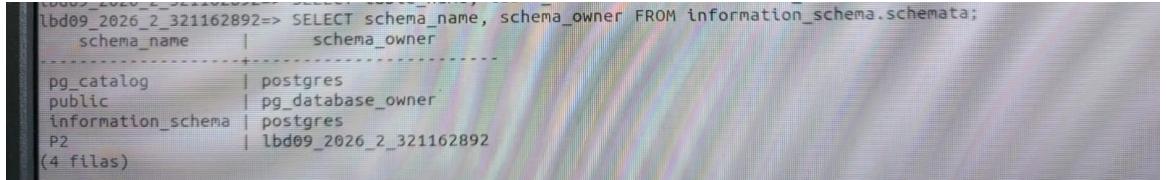
```
CREATE SCHEMA
lbd09_2026_2_321162892=> CREATE SCHEMA "P2" AUTHORIZATION "lbd09_2026_2_3211628
92";
CREATE SCHEMA
```

El esquema lo cree durante el laboratorio.

Un esquema es un espacio de nombres el cual contiene objetos de base de datos, tales como tablas, vistas, índices, tipos de datos, funciones y operadores, sirviendo como un contenedor lógico el cual nos permite una mejor organización de los objetos, facilitando a administración de permisos y mejora la seguridad y escalabilidad del sistema.

Como usos principales tenemos:

- a) Organización lógica: Nos permite agrupar objetos de la base de datos en compartimentos lógicos para que así sean más fáciles de administrar.

- b) Gestión de permisos y seguridad: Mediante este, múltiples usuarios pueden utilizar una misma base de datos entre sí, siendo que mediante la asignación de permisos específicos por esquema creemos un entorno multiusuario sin que interfieran entre sí.
  - c) Evitar colisiones de nombres: Diferentes esquemas dentro de la misma base de datos tengan objetos con el mismo nombre sin que estos entren en conflicto.
  - d) Estructuración de aplicaciones: Facilitan la organización de objetos en grupos lógicos, como por ejemplo al separar los datos de una aplicación mediante esquema\_app o administracion con esquema\_admin.
- **C3.-Captura de pantalla de la consulta a la vista information\_schema.schemata y respuesta a ¿cuántos esquemas se crean por defecto al crear una base de datos?**
- 
- ```

lbd09_2026_2_321162892=> SELECT schema_name, schema_owner FROM information_schema.schemata;
   schema_name    |    schema_owner
-----+-----
 pg_catalog | postgres
 public   | pg_database_owner
 information_schema | postgres
 p2       | lbd09_2026_2_321162892
(4 filas)

```
- Se crean 3 esquemas por defecto, pg\_catalog, public, information\_schema.
- **C4.-Captura de pantalla de la consulta a la vista information\_schema.views y respuesta a ¿cuál esquema no tiene creado por defecto ninguna vista?**

| table_name                      | table_schema |
|---------------------------------|--------------|
| pg_roles                        | pg_catalog   |
| pg_settings                     | pg_catalog   |
| pg_stat_activity                | pg_catalog   |
| pg_stat_replication             | pg_catalog   |
| pg_stat_slru                    | pg_catalog   |
| pg_stat_wal_receiver            | pg_catalog   |
| pg_stat_recovery_prefetch       | pg_catalog   |
| pg_group                        | pg_catalog   |
| pg_user                         | pg_catalog   |
| pg_policies                     | pg_catalog   |
| pg_rules                        | pg_catalog   |
| pg_views                        | pg_catalog   |
| pg_tables                       | pg_catalog   |
| pg_matviews                     | pg_catalog   |
| pg_indexes                      | pg_catalog   |
| pg_sequences                    | pg_catalog   |
| pg_stats                        | pg_catalog   |
| pg_stats_ext                    | pg_catalog   |
| pg_stats_ext_exprs              | pg_catalog   |
| pg_publication_tables           | pg_catalog   |
| pg_locks                        | pg_catalog   |
| pg_cursors                      | pg_catalog   |
| pg_available_extensions         | pg_catalog   |
| pg_available_extension_versions | pg_catalog   |
| pg_prepared_xacts               | pg_catalog   |
| pg_prepared_statements          | pg_catalog   |
| pg_seclabels                    | pg_catalog   |
| pg_timezone_abbrevs             | pg_catalog   |
| pg_timezone_names               | pg_catalog   |
| pg_stat_sys_tables              | pg_catalog   |
| pg_stat_xact_sys_tables         | pg_catalog   |
| pg_stat_user_tables             | pg_catalog   |
| pg_stat_all_tables              | pg_catalog   |
| pg_stat_xact_all_tables         | pg_catalog   |
| pg_stat_xact_user_tables        | pg_catalog   |
| pg_statio_all_tables            | pg_catalog   |
| pg_statio_sys_tables            | pg_catalog   |
| pg_statio_user_tables           | pg_catalog   |
| pg_stat_all_indexes             | pg_catalog   |
| pg_stat_sys_indexes             | pg_catalog   |
| pg_stat_user_indexes            | pg_catalog   |
| pg_statio_all_indexes           | pg_catalog   |
| pg_statio_sys_indexes           | pg_catalog   |
| pg_statio_user_indexes          | pg_catalog   |
| pg_statio_all_sequences         | pg_catalog   |
| pg_statio_sys_sequences         | pg_catalog   |
| pg_statio_user_sequences        | pg_catalog   |
| pg_stat_subscription            | pg_catalog   |
| pg_stat_ssl                     | pg_catalog   |
| pg_stat_gssapi                  | pg_catalog   |

El esquema que no tiene creada por defecto ninguna vista es “public” ya que este es el esquema de trabajo de nosotros y comienza vacío.

- **C5.-Captura de pantalla de la consulta a la vista information\_schema.tables.**

```
lbd09_2026_2_321162892=> SELECT table_name, table_type
FROM information_schema.tables WHERE table_schema = 'P2'
;
table_name | table_type
-----+-----
estudiante | BASE TABLE
(1 fila)
```

- **C6.-Captura de pantalla de la consulta a la vista information\_schema.sequences, las sentencias de inserción y respuesta a ¿qué es una secuencia?**

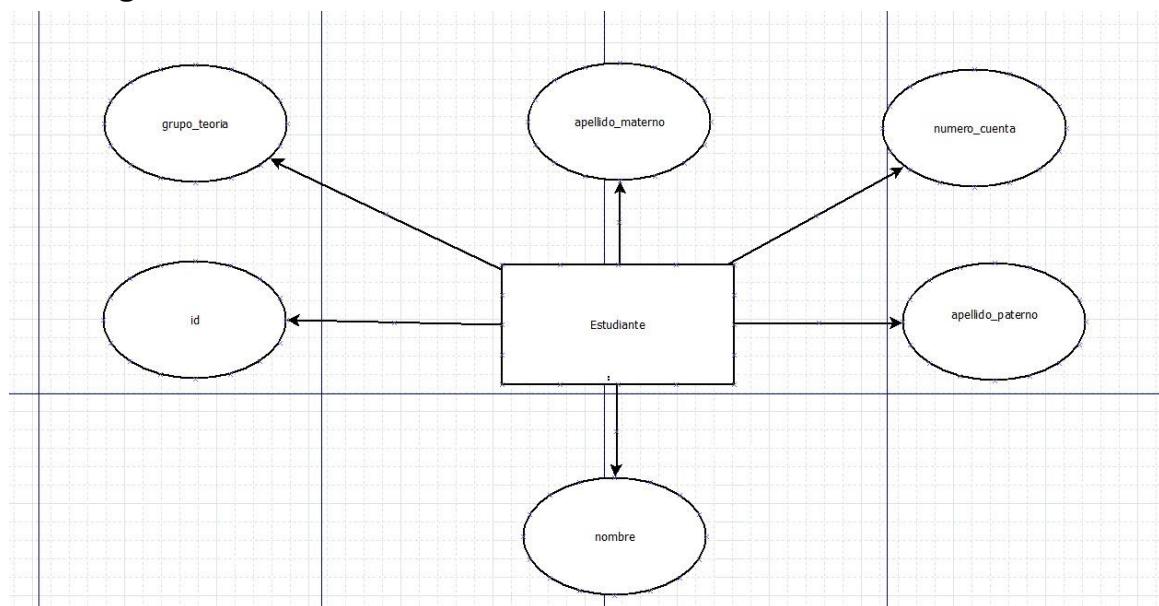
```
lbd09_2026_2_321162892=> SELECT sequence_schema, sequence_name, data_type, start_value, minimum_value, maximum_value, increment FROM information_schema.sequences;
sequence_schema | sequence_name | data_type | start_value | minimum_value | maximum_value | increment
-----+-----+-----+-----+-----+-----+-----+
P2 | estudiante_id_seq | integer | 1 | 1 | 2147483647 | 1
(1 fila)
```

Una secuencia es un objeto de base de datos el cual genera de manera automática valores numéricos consecutivos, generalmente empleados como identificadores únicos en nuestras tablas.

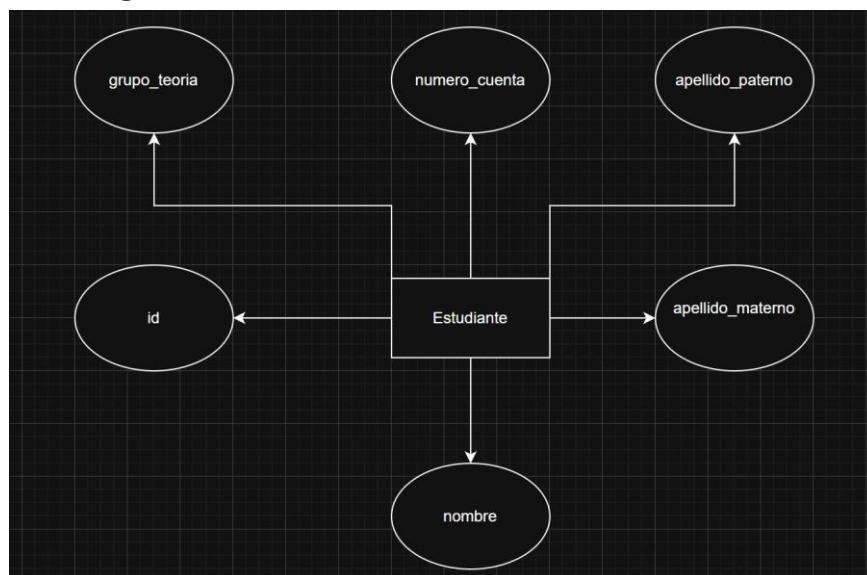
- **C7.-Captura de pantalla de la consulta a los registros insertados en la tabla estudiante.**

```
lbd09_2026_2_321162892=> SELECT * FROM "P2".estudiante;
+----+-----+-----+-----+-----+
| id | nombre | apellido_paterno | apellido_materno | num_cuenta |
+----+-----+-----+-----+-----+
| 0  | Mario  | Vidrio          | Lopez           | 321162892   |
| 1  | Edgar  | Soto            |                 | 1064511     |
+----+-----+-----+-----+-----+
(2 filas)
```

- **C8.-Diagrama ER de la tabla estudiante con DIA**



- **C9.-Diagrama ER de la tabla estudiante con Draw.io**



- **C10.-Captura de pantalla de la ejecución de comandos su –l postgres –c “/usr/pgsql-15/bin/initdb -D /u01/pgdata/15/data” en el contenedor.**

```
bash-5.1# su -l postgres -c "/usr/pgsql-15/bin/initdb -D /u01/pgdata/15/data"
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "C.UTF8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /u01/pgdata/15/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and
--auth-host, the next time you run initdb.

Success. You can now start the database server using:

    /usr/pgsql-15/bin/pg_ctl -D /u01/pgdata/15/data -l logfile start
bash-5.1# 
```

- **C11.-Captura de pantalla de la ejecución del comando su –l postgres –c “psql” en el contenedor.**

```
bash-5.1# su - postgres -c "/usr/pgsql-15/bin/pg_ctl -D /u01/pgdata/15/data -l logfile start"
waiting for server to start.... done
server started
bash-5.1# su - postgres -c "psql"
psql (15.16)
Type "help" for help.

postgres=# 
```

- **C12.-Captura de pantalla de la ejecucion del comando sudo dmidecode –t system en el entorno local del alumno.**

```
root@pipardo-Predator-PHN16-71:/# sudo dmidecode -t system
# dmidecode 3.5
Getting SMBIOS data from sysfs.
SMBIOS 3.4 present.

Handle 0x0001, DMI type 1, 27 bytes
System Information
    Manufacturer: Acer
    Product Name: Predator PHN16-71
    Version: V1.13
    Serial Number: NHQMBAA00132512EE87600
    UUID: 4c3ce671-8dac-be59-74d4dd08582a
    Wake-up Type: Power Switch
    SKU Number: 0000000000000000
    Family: Predator Helios Neo 16

Handle 0x000F, DMI type 12, 5 bytes
System Configuration Options
    Option 1: ConfigOptions1
    Option 2: ConfigOptions2
    Option 3: ConfigOptions3
    Option 4: ConfigOptions4
    Option 5: ConfigOptions5
    Option 6: ConfigOptions6

root@pipardo-Predator-PHN16-71:/# 
```

## Conclusiones

Tras realizar esta práctica puedo concluir que se han cumplido todos los objetivos de esta misma de manera satisfactoria, siendo que realizamos las configuraciones iniciales para ingresar al servidor de base de datos PostgreSQL del laboratorio de la Facultad de Ingeniería, al igual que instalamos y configuramos satisfactoriamente el sistema operativo como punto inicial para el ambiente en el que trabajaremos de manera local, también se logró identificar partes teóricas importantes, como saber que es un esquema y sus usos o también el uso de herramientas para la creación de diagramas ER y por último el uso de Docker, por lo mismo considero que se adquirieron todos los conocimientos esperados.

## Bibliografia

Inc, D. (2026a, enero 8). *Get started*. Docker Documentation. <https://docs.docker.com/get-started/>

*draw.io - free flowchart maker and diagrams online*. (s. f.). <https://www.draw.io/>

*Chapter 37. The Information Schema*. (2026, 12 febrero). PostgreSQL Documentation.

<https://www.postgresql.org/docs/15/information-schema.html>

*PostgreSQL 15.16 documentation*. (2026, 12 febrero). PostgreSQL Documentation.

<https://www.postgresql.org/docs/15/index.html>

*FAQ - CentOS Project shifts focus to CentOS Stream - The CentOS Project.* (s. f.).

<https://www.centos.org/distro-faq/>

Macke, S. (s. f.). *Dia draws your structured diagrams: Free Windows, Mac OS X and Linux*

*version of the popular open source program.* <http://dia-installer.de/>