



Instituto Politécnico Nacional
Escuela Superior de Cómputo



Unidad de aprendizaje:
Bases de datos

PRÁCTICA 2: *Introducción a PostgreSQL*

Alumno: González Cárdenas Ángel Aquilez

Boleta: 2016630152

Grupo: 3CV1

Profesor: Blanco Almazán Iván Eduardo

1. Objetivos

1. Instalar de forma adecuada el sistema gestor de base de datos *PostgreSQL*.
2. Crear una base de datos que contenga una tabla y registros.

2. Introducción

El sistema gestor de bases de datos que ahora conocemos como *PostgreSQL* tiene sus orígenes en el proyecto *POSTGRES*, creados en la Universidad de California en Berkeley.

Para poder trabajar con *PostgreSQL*, primero se instaló el software en el sistema operativo GNU/Linux *Xubuntu*, nombre compuesto de el sistema operativo base Ubuntu y el entorno de escritorio XFCE.

Ejecutamos el siguiente comando en la terminal:

```
sudo apt install postgresql
```

Una vez terminada la instalación, procedemos a verificar el estatus del *daemon* con el siguiente comando:

```
service postgresql status
```

el cual nos devuelve la siguiente información:

```
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sat 2023-04-01 10:30:13 CST; 3h 11min ago
     Main PID: 68103 (code=exited, status=0/SUCCESS)
        CPU: 2ms

abr 01 10:30:13 NeoTosti systemd[1]: Starting PostgreSQL RDBMS...
abr 01 10:30:13 NeoTosti systemd[1]: Finished PostgreSQL RDBMS.
```

Figura 1: Estatus del *daemon*

Finalmente ingresamos a la línea de comandos introduciendo lo siguiente:

```
sudo -u postgres psql
```

Una vez dentro, verificamos el estado de la conexión mediante el comando `\conninfo` que en conjunto, nos arrojaron lo siguiente:

```
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sat 2023-04-01 10:30:13 CST; 3h 11min ago
     Main PID: 68103 (code=exited, status=0/SUCCESS)
        CPU: 2ms

abr 01 10:30:13 NeoTosti systemd[1]: Starting PostgreSQL RDBMS...
abr 01 10:30:13 NeoTosti systemd[1]: Finished PostgreSQL RDBMS.
```

Figura 2: Ingreso a la línea de comandos de *PostgreSQL*

3. Desarrollo

3.1. Creación de la base de datos

Comenzamos creando una base de datos simple, la cual denominamos *springfield* mediante el siguiente comando de SQL:

CREATE DATABASE SPRINGFIELD;

Después comprobamos su existencia con el comando \l:

```
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sat 2023-04-01 10:30:13 CST; 3h 11min ago
     Main PID: 68103 (code=exited, status=0/SUCCESS)
        CPU: 2ms

abr 01 10:30:13 NeoTosti systemd[1]: Starting PostgreSQL RDBMS...
abr 01 10:30:13 NeoTosti systemd[1]: Finished PostgreSQL RDBMS.
```

Figura 3: Listado de todas las bases de datos disponibles

Ingresamos a nuestra base de datos con el comando \c seguido del nombre, en este caso:

```
\c      springfield
```

y nos arrojó el siguiente mensaje:

```
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sat 2023-04-01 10:30:13 CST; 3h 11min ago
     Main PID: 68103 (code=exited, status=0/SUCCESS)
        CPU: 2ms

abr 01 10:30:13 NeoTosti systemd[1]: Starting PostgreSQL RDBMS...
abr 01 10:30:13 NeoTosti systemd[1]: Finished PostgreSQL RDBMS.
```

Figura 4: Conexión exitosa a la base de datos

3.2. Creación de una tabla y elementos

Dentro de la base de datos, creamos la tabla **persona**, que contiene los siguientes campos:

1. id SERIAL PRIMARY KEY,
2. nombres VARCHAR,
3. ap_paterno VARCHAR,
4. ap_materno VARCHAR,
5. fecha_nacimiento DATE,
6. sexo CHAR

Con el comando CREATE TABLE y nuestros atributos definidos, ingresamos:

```
CREATE TABLE PERSONA (
    id SERIAL PRIMARY KEY,
    nombres VARCHAR,
    ap_paterno VARCHAR,
    ap_materno VARCHAR,
    fecha_nacimiento DATE,
    sexo CHAR
);
```

Y con el comando \d seguido del nombre de la tabla, comprobamos que se creo de manera adecuada:

Luego, agregamos los siguientes registros:

```

• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sat 2023-04-01 10:30:13 CST; 3h 11min ago
     Main PID: 68103 (code=exited, status=0/SUCCESS)
        CPU: 2ms

abr 01 10:30:13 NeoTosti systemd[1]: Starting PostgreSQL RDBMS...
abr 01 10:30:13 NeoTosti systemd[1]: Finished PostgreSQL RDBMS.

```

Figura 5: Creación de la tabla personas

INSERT INTO PERSONA VALUES

(1, 'Homero', 'Simpson', 'J', '12-05-1956', 'M');

INSERT INTO PERSONA VALUES

(2, 'Marge', 'Simpson', 'Bouvier', '19-03-1956', 'F');

y para comprobar su existencia, se consultan mediante la sentencia:

SELECT * FROM PERSONA;

produciendo así el siguiente resultado:

```

• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sat 2023-04-01 10:30:13 CST; 3h 11min ago
     Main PID: 68103 (code=exited, status=0/SUCCESS)
        CPU: 2ms

abr 01 10:30:13 NeoTosti systemd[1]: Starting PostgreSQL RDBMS...
abr 01 10:30:13 NeoTosti systemd[1]: Finished PostgreSQL RDBMS.

```

Figura 6: Registros en la tabla persona

4. Conclusión

Al termino de la práctica, se instaló el sistema gestor de base de datos PostgreSQL, se creo una base de datos con una tabla y se ingresaron registros, permitiendo así cumplir con los objetivos planteados.

5. Referencias

- A Brief History of PostgreSQL. (2023, febrero 9). PostgreSQL Documentation. <https://www.postgresql.org/docs/current/history.html>
- Levinas, M. (s/f). How to Install and Setup PostgreSQL server on Ubuntu 20.04. Cherry Servers. Recuperado el 1 de abril de 2023, de <https://www.cherryservers.com/blog/how-to-install-and-setup-postgresql-server-on-ubuntu-20-04>