

## Tarea 2: Base de datos en PostgreSQL

Alan Andrés Mérida Morales, 202100023<sup>1,\*</sup>

<sup>1</sup>*Facultad de Ingeniería, Escuela de Ingeniería Mecánica Eléctrica,  
Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.*

### I. OBJETIVOS

- Instalar un software para realizar bases de datos.
- Crear una base de datos en PostgreSQL.
- Conectar el software de Octave con el software de PostgreSQL.

### II. INTRODUCCIÓN

En el siguiente reporte se muestran los resultados que se obtuvieron para esta práctica. En dicha práctica se buscó instalar un software para realizar bases de datos, por lo tanto, se instaló PostgreSQL. Se muestra el código que se realizó para crear la tabla de base de datos en PostgreSQL y de igual forma insertar valores en dicha tabla. También se utilizó el software de Octave de la práctica anterior para conectar con la base de datos creada, de esta forma se puede visualizar desde otro ambiente y se puede editar desde Octave.

### III. MARCO TEÓRICO

#### A. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, reconocido por su robustez, extensibilidad y cumplimiento de estándares SQL. Fue desarrollado originalmente en 1986 en el Departamento de Ciencias de la Computación de la Universidad de California, Berkeley, y desde entonces ha evolucionado hasta convertirse en una de las bases de datos más avanzadas y completas disponibles en la actualidad. Sus características claves son:

- PostgreSQL es de código abierto bajo la licencia PostgreSQL, que permite su uso, modificación y distribución gratuita.
- Cuenta con una comunidad activa de desarrolladores y usuarios que contribuyen al desarrollo continuo y al soporte de la plataforma.

- PostgreSQL es conocido por su alto cumplimiento de estándares SQL, lo que garantiza la portabilidad de aplicaciones entre diferentes sistemas de bases de datos.
- Soporta el estándar SQL ANSI/ISO, proporcionando una gran compatibilidad y facilidad de uso para desarrolladores que ya están familiarizados con SQL.
- Una de las características más destacadas de PostgreSQL es su capacidad para ser extendido mediante el uso de funciones personalizadas, tipos de datos, operadores y procedimientos almacenados.
- Los usuarios pueden definir sus propios tipos de datos, funciones, operadores y lenguajes de procedimiento.
- Ofrece mecanismos avanzados de control de transacciones, incluyendo soporte completo para ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), asegurando la integridad y fiabilidad de los datos.
- Soporte para JSON: Permite almacenar y consultar datos JSON, facilitando la integración con aplicaciones que manejan documentos JSON.
- Indexación Avanzada: Incluye diversos métodos de indexación, como B-tree, Hash, GiST, SP-GiST, GIN y BRIN, que mejoran significativamente el rendimiento de las consultas.
- Gestión de Concurrency: Implementa el control de concurrencia multiversion (MVCC), permitiendo que múltiples usuarios accedan a la base de datos simultáneamente sin bloqueos.

#### 1. Arquitectura

La arquitectura de PostgreSQL se puede dividir en varias capas y componentes clave:

- Procesos del Servidor.
  - Postmaster: Es el proceso principal que maneja la inicialización, las conexiones entrantes y la creación de procesos hijos.
  - Procesos Hijos: Incluyen procesos de fondo como autovacuum, wal writer, archiver, entre otros, que gestionan tareas de mantenimiento y optimización.

---

\* 3690273450101@ingenieria.usac.edu.gt

- **Gestión de Almacenamiento:** PostgreSQL utiliza un sistema de almacenamiento orientado a páginas, donde los datos se almacenan en bloques de tamaño fijo. Esto permite un acceso eficiente y rápido a los datos.
- **Sistema de Transacciones:** Implementa el control de concurrencia multiversion (MVCC) para manejar múltiples transacciones de forma simultánea, evitando conflictos de bloqueo.
- **Lenguaje y Parser:** Integra un parser SQL que traduce las consultas SQL en un plan de ejecución, optimizando la forma en que se accede a los datos.

## 2. Uso y aplicaciones

PostgreSQL es utilizado en una amplia gama de aplicaciones debido a sus características avanzadas y su capacidad para manejar grandes volúmenes de datos. Algunas de las áreas de aplicación incluyen:

- **Aplicaciones Web y Móviles:** Es una opción popular para el backend de aplicaciones web y móviles, gracias a su capacidad para manejar transacciones complejas y grandes volúmenes de datos.
- **Análisis de Datos:** Ofrece capacidades de análisis y almacenamiento de datos masivos, siendo una opción preferida para soluciones de Business Intelligence (BI).
- **Sistemas Geoespaciales:** Con el soporte de la extensión PostGIS, PostgreSQL se utiliza ampliamente en sistemas de información geográfica (GIS).
- **Finanzas y Comercio Electrónico:** La integridad y seguridad de datos que ofrece PostgreSQL lo hace ideal para sistemas financieros y plataformas de comercio electrónico.

## IV. RESULTADOS

### A. Códigos realizados

#### 1. PostgreSQL

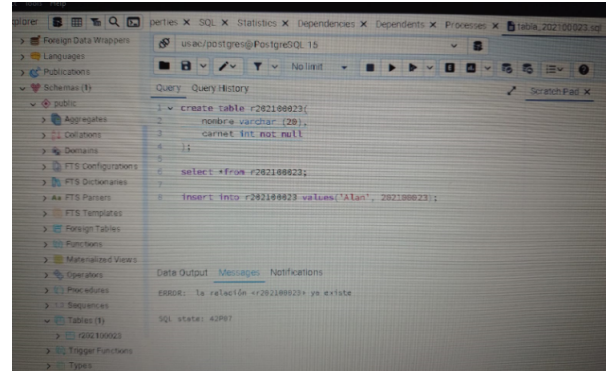


Figura 1: Código utilizado para crear la tabla de base de datos.

#### 2. Octave

```
octave> pg_load database
octave> conn = pg_connect(setdbopts('dbname','usac','host','localhost','port','5432','user','postgres','password','A@merida123'))
conn = (dbconn object)
octave> N=pg_exec_params(conn, "insert into r202100023 values ('Roberto','20150789')");
N = 1
octave> N=pg_exec_params(conn, 'select * from r202100023;')
N =
```

Figura 2: Código utilizado para conectar octave con la base da datos de PostgreSQL

### B. Explicación de los códigos

#### 1. Código en PostgreSQL

##### Línea create table r202100023(

Esta línea crea una nueva tabla en la base de datos llamada r202100023.

##### nombre varchar(20):

Define una columna llamada nombre que puede almacenar cadenas de texto de hasta 20 caracteres.

##### carnet int not null:

Define una columna llamada carnet que almacena números enteros y que no puede tener valores nulos (es obligatorio proporcionar un valor para esta columna).

**select \* from r202100023;:** Esta línea selecciona y muestra todos los registros de la tabla r202100023. El asterisco (\*) indica que se quieren seleccionar todas las columnas de la tabla.

**insert into r202100023 values('Alan', 202100023);:**

Esta línea inserta un nuevo registro en la tabla r202100023. Los valores 'Alan' y 202100023 se insertan en las columnas nombre y carnet, respectivamente.

## 2. Código en Octave

### pkg load database:

Esta línea carga el paquete database en Octave, que proporciona funciones para conectarse y realizar operaciones en bases de datos, incluidas las bases de datos PostgreSQL.

### pq\_connect:

Esta función establece una conexión a una base de datos PostgreSQL.

### setdbopts:

Especifica las opciones de conexión para la base de datos.

- 'dbname','usac': El nombre de la base de datos a la que se conecta es usac.
- 'host','localhost': El servidor de la base de datos está en la misma máquina donde se está ejecutando el script, lo que se indica con localhost.
- 'port','5432': Utiliza el puerto estándar de PostgreSQL, que es el 5432.
- 'user','postgres': Se conecta con el usuario postgres, que es el usuario administrador por defecto de PostgreSQL.
- 'password','A@merida123': La contraseña para el usuario postgres es A@merida123.

### pq\_exec\_params:

Esta función ejecuta una consulta SQL en la conexión proporcionada. Se utiliza para ejecutar comandos de inserción, selección, actualización o eliminación de registros.

**insert into r202100023 values ('Roberto','201556789');**

Esta consulta inserta un nuevo registro en la tabla r202100023 con los valores 'Roberto' para la columna nombre y '201556789' para la columna carnet.

**N:**

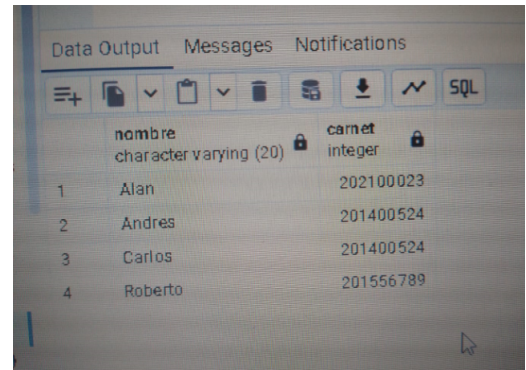
La variable N almacenará el resultado de la operación de inserción, que puede incluir información sobre el éxito o el fracaso de la operación.

**N: select \* from r202100023;**

Esta consulta selecciona todos los registros de la tabla r202100023. El resultado incluirá todos los datos actualmente almacenados en la tabla, incluidas las inserciones previas como la de 'Roberto'. La variable N ahora almacenará los resultados de la consulta de selección, que serán todos los registros en la tabla después de la inserción.

## C. Tabla de datos final

### 1. PostgreSQL



	nombre character varying (20)	carnet integer
1	Alan	202100023
2	Andres	201400524
3	Carlos	201400524
4	Roberto	201556789

Figura 3: Base de datos visualizada desde PostgreSQL

### 2. Octave

```
octave:11> N=pq_exec_params(conn, 'select * from r202100023;')
N =
scalar structure containing the fields:

data =
{
  [1,1] = Alan
  [2,1] = Andres
  [3,1] = Carlos
  [4,1] = Roberto
  [1,2] = 202100023
  [2,2] = 201400524
  [3,2] = 201400524
  [4,2] = 201556789
}

columns =
{
  [1,1] = nombre
  [1,2] = carnet
}

types =
1x2 struct array containing the fields:

name
is_array
is_composite
is_enum
elements
```

Figura 4: Base de datos obtenida desde Octave

## D. Repositorio privado

Repositorio creado en Github

## V. CONCLUSIONES

- \* Se instaló el software de PostgreSQL debido a su fiabilidad, conformidad con los estándares, extensibilidad, soporte para datos no estructurados, y su naturaleza de código abierto.
- \* Como se puede observar en el código desarrollado para crear la tabla de base de datos en el software

resulta ser un código simple y se crea la tabla sin ningún problema.

- \* Haciendo uso del software de Octave se logró conectar con la base de datos que se creó en PostgreSQL y desde su terminal se logra agregar datos a la tabla y se puede llegar a visualizar.
- \* Si se visualiza la tabla de los datos desde Octave podremos observar una tabla presentada en forma de lista, con una combinación de números al inicio

que indican la fila y columna, pero si se observa desde el programa de pgAdmin nos presenta una tabla de la forma en que se esperaría visualizarla.

- \* Al lograr hacer que Octave se conecte con la base de datos esto nos brinda una ventaja ya que en casos como el mío, la computadora en algunas ocasiones presenta error al querer ejecutar el programa de pgAdmin, por lo tanto, al poder manipular la tabla desde Octave esto nos da la oportunidad de poder trabajar sin problema en la base de datos.

- 
- [1] <https://www.postgresql.org/community/>
  - [2] <https://www.postgresql.org/docs/>
  - [3] [https://wiki.octave.org/Database\\_package](https://wiki.octave.org/Database_package)
  - [4] [https://wiki.postgresql.org/wiki/Main\\_Page](https://wiki.postgresql.org/wiki/Main_Page)