

Tec. en Desarrollo de Software Base De Datos II Sede Rawson. Trabajo Práctico N° 1

Docente: Salerno Fabio

Alumnos:

• Alvarez Maria Emilia

- Delgado Marcela Alejandra
- Fava Maximiliano
- Loureiro Jessica Mariana
- Simon Camila

Introducción

En este informe se presenta el desarrollo de un software de administración de bases de datos PostgreSQL. El objetivo principal es integrar las consultas desarrolladas en el Laboratorio 3 en un software que permita monitorear y obtener información detallada de una base de datos en PostgreSQL.

Objetivos del Software

- Mostrar un menú con opciones para monitorear y obtener información de la base de datos.
- 2. Ejecutar las consultas correspondientes a cada opción del menú.
- 3. Mostrar los resultados de las consultas en formato de tabla.
- 4. Permitir la ejecución de consultas personalizadas por parte del usuario.

Funcionalidades Implementadas

- 1. Conexión a la base de datos: Establece la conexión con la base de datos PostgreSQL utilizando los datos de acceso proporcionados.
- 2. Menú de consultas: Presenta un menú con las siguientes opciones:
 - Obtener usuarios conectados, IP y consulta.
 - Obtener el tamaño de la base de datos del servidor.
 - Obtener el tamaño de una tabla.
 - Obtener detalles de las tablas: orden de los campos, nombre y tipo de dato.
 - Obtener el tamaño de las tablas de un esquema.
 - Listado detallado por base de datos.
 - Consulta personalizada.
- 3. Ejecución de consultas: Ejecuta la consulta SQL correspondiente a la opción seleccionada por el usuario y muestra los resultados en formato de tabla.
- 4. Consultas personalizadas: Permite al usuario ingresar una consulta personalizada que se ejecutará y mostrará los resultados.

Tecnologías Utilizadas

- Python 3.12.
- <u>psycopg2</u>: Permite la conexión y ejecución de consultas en bases de datos PostgreSQL.
- Visual Studio Code.
- Terminal de Ubuntu.

Estructura del Software

El software se compone de los siguientes archivos:

1. Conexion_bdd.py: Contiene la función **conexion_bdd()** que establece la conexión con la base de datos.

```
conexion_bdd.py
BBDD2 > TP > TP1 > ♥ conexion_bdd.py >
       import psycopg2
      from psycopg2 import sql
      def conexion_bdd():
           try:
  6
               connection = psycopg2.connect(
                   dbname='universidad', #Aca podemos modificar la base de datos a la que queremos acceder
                   user='postgres',
  8
                   password='admin',
  9
                   host='localhost',
port='5432'
 10
 11
 12
 13
               return connection
 14
           except Exception as e:
               print(f"Error al conectar a la base de datos: {e}")
 15
```

2. Diccionario_consultas.py: Define un diccionario llamado **consultas** que almacena las consultas SQL correspondientes a cada opción del menú.

3. Ejecutar_consulta.py: Contiene la función **ejecutar_consulta(connection, query)** que ejecuta una consulta SQL y muestra los resultados en formato de tabla.

```
ejecutar_consulta.py
BBDD2 > TP > TP1 > 🍖 ejecutar_consulta.py > ..
        from psycopg2 import sql
        def ejecutar_consulta(connection, query):
                 cursor = connection.cursor()
   6
                  cursor.execute(query)
   8
                  if cursor.description:
   9
                      # Imprimir encabezados de columna
                      column_names = [desc[0] for desc in cursor.description]
print("\n" + " | ".join(f"{name:<20}" for name in column_names))  # Formato de encabezados
print("-" * (len(column_names) * 22))  # Línea divisoria</pre>
  10
 13
  14
                      # Imprimir resultados
                      results = cursor.fetchall()
  15
 16
                       for row in results:
 17
                           print(" | ".join(f"{str(value):<20}" for value in row)) # Formato de filas</pre>
 18
                  else:
                      print("Consulta ejecutada correctamente, pero no se devolvieron resultados.")
  19
 20
                  cursor.close()
 21
 22
                  print(f"Error al ejecutar la consulta: {e}")
  23
```

4. main.py: Es el archivo principal que ejecuta el programa, presenta el menú de consultas y maneja la interacción con el usuario.

Menú del Software

Opción 1: Obtener los Usuarios Conectados.

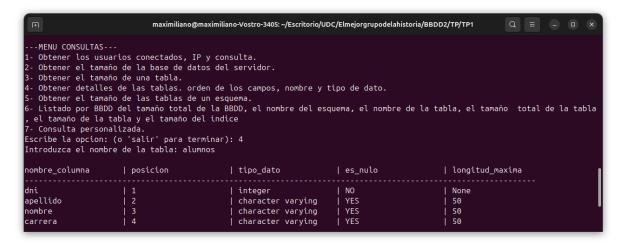


Opción 2 Obtener el tamaño de la base de datos del servidor:



Opción 3 Obtener el tamaño de una tabla:

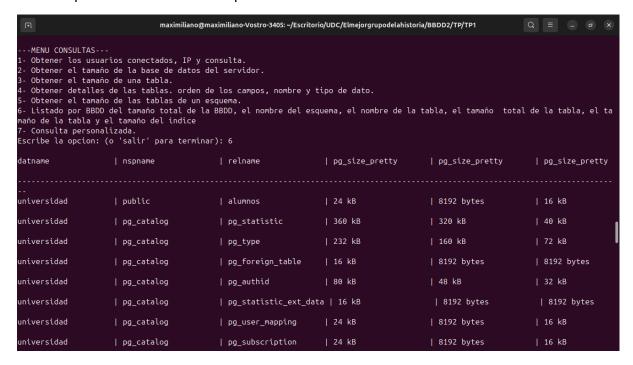
Opción 4 Obtener detalles de una tabla:



Opción 5 Obtener el tamaño de las tablas de un esquema:



Opción 6 Obtener Listado por BBDD:



Opción 7 Consulta personalizada:

