Proyecto de Software

Tema

• Acceso a Bases de Datos.

Accediendo a Bases de Datos

Lenguaje SQL (Structured Query Language)

- Sentencias insert, update, select, etc....
- Ejemplos:
 - select * from tabla where condición
 - insert into tabla (campos) values (valores)
 - update tabla set campo1='valor1' where condición

Importante

MySQL, PostgreSQL: motores de base de datos

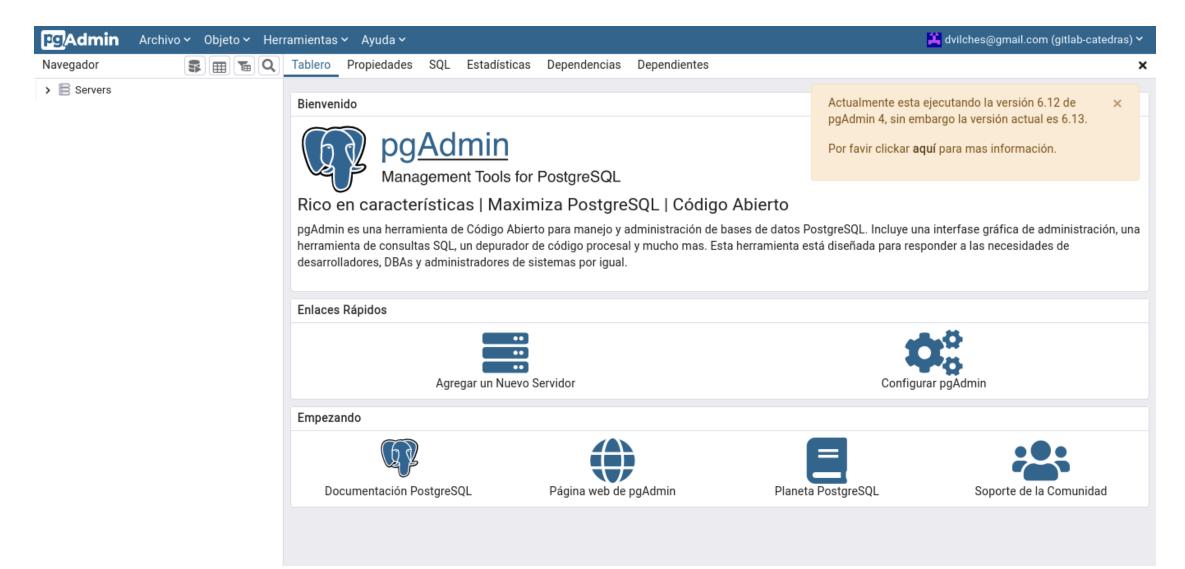
SQL: lenguaje de consulta

pgAdmin

- Interfaz de Administración de la Base de Datos PostgreSQL
- Podemos exportar e importar a varios formatos



pgAdmin



- Vamos a acceder a través de psycopg2.
- En Ubuntu tuve que instalar:

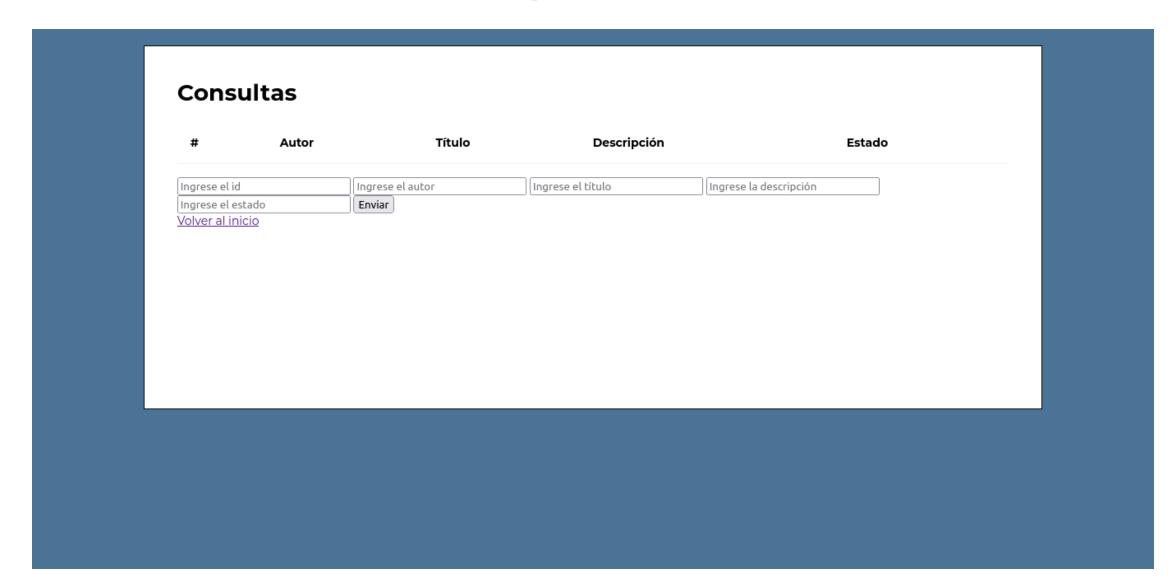
```
sudo apt-get install libpq-dev
```

Luego

```
poetry add psycopg2-binary
```

```
#src/core/issue.py
import psycopg2
conn = psycopg2.connect(
        host="localhost",
        database="proyecto_db",
        user="proyecto_db",
        password="proyecto_db")
cur = conn.cursor()
cur.execute('select * from issues')
issues = cur.fetchall()
cur.close()
conn.close()
```

- Algún cambio en el controlador o en la vista?
- Que piensan?
- Probamos...



```
import psycopg2, psycopg2.extras
class Issue:
    def getAll(self):
        conn = psycopg2.connect(
            host="localhost",
            database="proyecto_db",
            user="proyecto_db",
           password="proyecto_db")
        cur = conn.cursor(cursor_factory=psycopg2.extras.RealDictCursor)
        cur.execute('select * from issues')
        issues = cur.fetchall()
        print(issues)
        cur.close()
        conn.close()
        return issues
    def insert (self, issue):
        conn = psycopg2.connect(
               host="localhost",
                database="proyecto_db",
               user="proyecto_db",
                password="proyecto_db")
        cur = conn.cursor()
        cur.execute(f"insert into issues (\"user\", title, description, status) values ('{issue['user']}', '{issue['title']}', '{issue['description']}', '{issue['status']}')")
        conn.commit()
        cur.close()
        conn.close()
```

```
from flask import render_template, request, Blueprint
from src.core.issue import Issue
issue_blueprint = Blueprint("issues", __name__, url_prefix="/issues")
@issue blueprint.route("/")
def issues index():
    issue = Issue()
    issues = issue.getAll()
    return render template("issues/index.html", issues=issues)
@issue_blueprint.route("/add", methods=["POST"])
def issues add():
    data issue = {"id": request.form.get("id"),
             "user": request.form.get("user"),
             "title": request.form.get("title"),
             "description": request.form.get("description"),
             "status": request.form.get("status"),
    issue = Issue()
    issue.insert(data_issue)
    issues = issue.getAll()
    return render_template("issues/index.html", issues=issues)
```

Ahora bien ...

- ¿Qué pasa si queremos migrar de motor de BDD?
- ¿Qué pasa si queremos tener múltiples BDD conectadas?
- ¿Qué pasa cuando queremos incorporar otro modelo?

Enlaces

- PostgreSQL: https://www.postgresql.org/
- PgAdmin de la cátedra: https://sql.proyecto2025.linti.unlp.edu.ar
- https://www.digitalocean.com/community/tutorials/how-to-use-a-postgresql-database-in-a-flask-application
- https://badcodernocookie.com/pg_config-executable-not-found/

Fin