# **PROYECTO DE SOFTWARE**

Cursada 2022

## 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

**SUL**: lenguaje de consulta

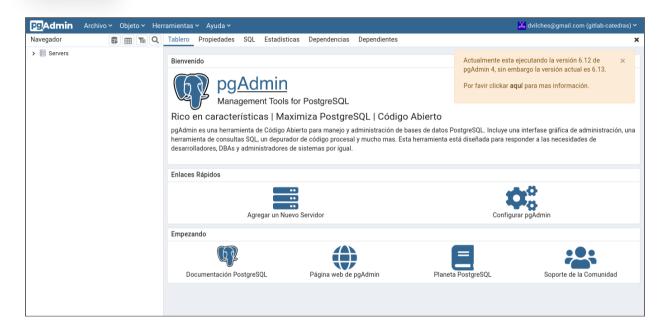
### **PGADMIN**

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



- Sitio oficial de \_\_\_\_\_
- pgAdmin de la cátedra:

#### **PGADMIN**



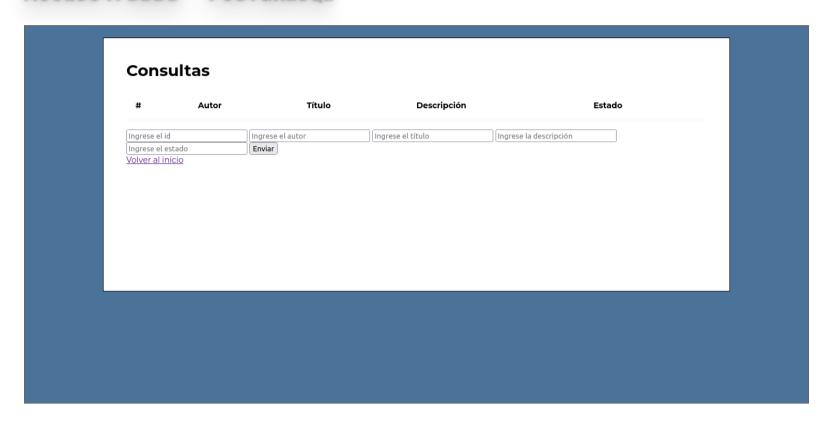
- Vamos a acceder a través de psycopg2.
- En Ubuntu 20.04 tuve que instalar:

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

Luego

poetry add psycopg2

- 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="provecto db")
        cur = conn.cursor()
```

```
from flask import render template, request, Blueprint
from src.core.issue import Issue
issue blueprint = Blueprint("issues", name ,
url prefix="/issues")
def issues index():
    issue = Issue()
    issues = issue.getAll()
    return render template ("issues/index.html",
issues=issues)
def issues add():
    data issue = {"id": request.form.get("id"),}
             "user": request.form.get("user"),
             "title": request.form.get("title"),
request.form.get("description"),
             "status": request.form.get("status"),
    issue = Issue()
    issue.insert(data issue)
    issues = issue.ge\overline{t}All()
    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:	
•	PgAdmin de la cátedra:	
•		
•		

# FIN