## **PROYECTO DE SOFTWARE**

Cursada 2024

## 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 tuve que instalar:

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

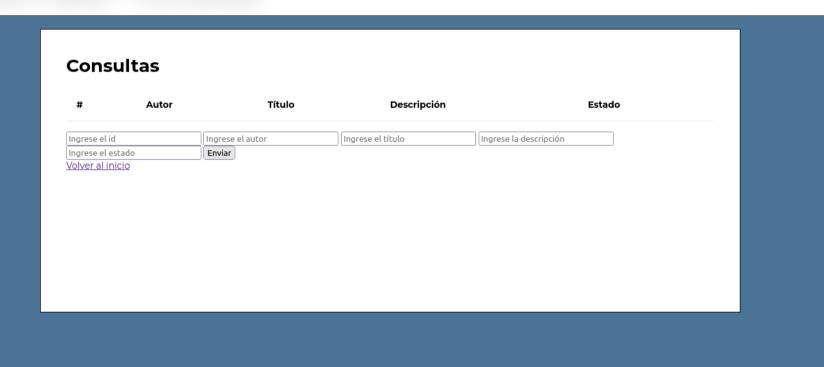
Luego

poetry add psycopg2

```
#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']}',
```

```
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"])
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:
)	
)	



Speaker notes