

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

SQL: 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

The screenshot displays the pgAdmin web interface. At the top, there is a dark blue header with the 'pgAdmin' logo and a navigation menu containing 'Archivo', 'Objeto', 'Herramientas', and 'Ayuda'. On the right side of the header, the user's email 'dvlches@gmail.com (gitlab-catedras)' is visible. Below the header is a secondary navigation bar with 'Navegador' and a search icon, followed by a breadcrumb trail: 'Tablero', 'Propiedades', 'SQL', 'Estadísticas', 'Dependencias', and 'Dependientes'. The main content area is titled 'Bienvenido' and features the pgAdmin logo and the text 'Management Tools for PostgreSQL'. A prominent orange notification box states: 'Actualmente esta ejecutando la versión 6.12 de pgAdmin 4, sin embargo la versión actual es 6.13. Por favor clicar aquí para mas información.' Below this, the text reads 'Rico en características | Maximiza PostgreSQL | Código Abierto' and provides a brief description of the tool. The 'Enlaces Rápidos' section contains two buttons: 'Agregar un Nuevo Servidor' and 'Configurar pgAdmin'. The 'Empezando' section at the bottom lists four links: 'Documentación PostgreSQL', 'Página web de pgAdmin', 'Planeta PostgreSQL', and 'Soporte de la Comunidad'.

ACCESO A BBDD – POSTGRESQL

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

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

- Luego

```
poetry add psycopg2
```


ACCESO A BBDD – POSTGRESQL

```
#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()
```

ACCESO A BBDD – POSTGRESQL

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

ACCESO A BBDD – POSTGRESQL

Consultas

#	Autor	Título	Descripción	Estado
<input type="text" value="Ingrese el id"/>	<input type="text" value="Ingrese el autor"/>	<input type="text" value="Ingrese el título"/>	<input type="text" value="Ingrese la descripción"/>	
<input type="text" value="Ingrese el estado"/>	<input type="button" value="Enviar"/>			
Volver al inicio				

ACCESO A BBDD – POSTGRESQL

```
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()
```

ACCESO A BBDD – POSTGRESQL

```
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: _____

- PgAdmin de la cátedra:

- _____

- _____

FIN