

# Ayudantía 3

## Web y SQL

---

Ayudantes:

Consuelo Inostroza, Amelia González, Tomás Jackson, Vicente San Martín  
Ricardo Oviedo, Daniela Torrent, Mateo Andrade

# Objetivos:

Esta ayudantía está enfocada en la Entrega 2. Hoy aprenderemos:

- Cómo trabajar y conectarse al **servidor**.
- **HTML**: Para crear la visualización de la página.
- **Python**: Para poder hacer la conexión a la base de datos y hacer consultas **SQL** a esta.
- Usaremos la librería **psycopg2** de python para poder trabajar con **postgres**.
- **IMPORTANTE**: ¿Cómo evitamos **inyecciones** en SQL?

# Uso del servidor

¿Como conectarse al servidor?

Conociendo nuestro número de grupo, en el terminal haremos lo siguiente:

```
ssh grupoXX@codd.ing.puc.cl
```

# Observaciones

- Nuestra contraseña por defecto será: grupoXX
- Para cambiar la contraseña:
  - Comando passwd
    - Pedirá la clave actual
    - Pedirá la clave nueva dos veces

# ¿Cómo utilizamos Postgresql en el server?

1. Luego de ingresar al servidor, hacemos el comando **psql**.
2. Ingresar contraseña (por defecto: grupoXX)
3. Cambiamos la contraseña:

**ALTER USER <grupoXX> ENCRYPTED PASSWORD 'newpassword';**

1. Ingresar a la base de datos correspondiente:

**\c grupoXXeN**

Para esta entrega N=2.

# ¿Cómo trabajamos en postgres?

- **Creación** de tablas:

```
CREATE TABLE users (User_id INT (serial) PRIMARY KEY, username VARCHAR (15) UNIQUE NOT NULL, ...);
```

- Poblamos tablas (**manualmente**):

```
INSERT INTO users (User_id, username, ...) VALUES (1, 'sdawda', ...);
```

- **Para poblarlas con un CSV:**

```
\COPY users (columns_name) FROM 'relative/path/to/file.csv' DELIMITER ',' CSV HEADER;
```

# Comandos útiles

- \l lista de base de datos.
- \c <db> Ingresar a la base de datos db.
- \dt lista de las tablas de esa base de datos
- \d <table\_name> descripción de la tabla <table\_name>
- \? todos los comandos de psql
- \q salir (CTRL + D también funciona)

# HTML

## Hyper Text Markup Language

- Es el lenguaje de marcado estándar utilizado para crear páginas web.
- Será la representación visual de la página.
- Se puede editar desde cualquier editor de texto.
- Cómo funciona:

Marca:

`<marca>Contenido</marca>`

Marca viene a ser el tipo de `marca` y contenido lo que está dentro de esta.



# Tipos de marca

- **html:** marca que describe el documento HTML (todas las otras marcas van dentro de él)
- **head:** Marca que contiene el encabezado (información sobre el documento) acá pueden agregar stylesheet.css
- **title:** Marca que contiene el título de la página.
- **body:** Marca que contiene el cuerpo del documento. Dentro de ella agregamos todos los elementos que queremos ver en nuestra visualización.
- **h1:** Marca que contiene un título.

# Tipos de marca

- **table:** Marca que define una tabla. las siguientes marcas van dentro de esta tabla:
  - **tr:** contiene la fila de una tabla.
  - **td:** contiene una celda de una tabla.
  - **th:** contiene el encabezado de la fila de una tabla.

# Python y psycopg2

Importamos la librería (nos sirve para trabajar con postgres):

```
import psycopg2
```

# Conexión con psycopg2

Objeto para conectar:

```
try:
    conn = psycopg2.connect(
        database="database",
        user="user",
        host="localhost",
        port=5432,
        password="pass")
except:
    print("No me pude conectar")
```

# Conexión con pyscopg2

- `conn.commit()`: Luego de haber realizado todos nuestros cambios en la base de datos, este código hará que nuestros cambios sean permanentes en ella.
- `conn.close()`: Cerramos la conexión a la base de datos.

## Inserción

```
query = "INSERT INTO Peliculas VALUES (" + \
        titulo + ", " + ano + \
        ", " + director + ")";
cur = conn.cursor()
try:
    cur.execute(query)
except psycopg2.Error as e:
    print(e.pgcode)
```

# Cursores (fetchone)

-Puntero que va recorriendo los resultados.

-Iremos iterando de fila en fila y las iremos imprimiendo una por una:

```
import psycopg2
```

```
try:
```

```
    conn = psycopg2.connect(database="dbname",  
                             user="dbuser", host="localhost",  
                             password="dbpass")
```

```
    cur = conn.cursor()
```

```
    cur.execute("SELECT * FROM R")
```

```
    row = cur.fetchone()
```

```
    while row:
```

```
        print(row)
```

```
        row = cur.fetchone()
```

```
except:
```

```
    print("Hubo algún problema")
```

## Otra forma (fetchall)

```
try:
    conn = psycopg2.connect(database="dbname",
                             user="dbuser", host="localhost",
                             password="dbpass")
    cur = conn.cursor()
    cur.execute("SELECT * FROM R")
    rows = cur.fetchall()
    for row in rows:
        print(row)
except:
    print("Hubo algún problema")
```



# Pasar parámetros (en orden)

Ingresar argumentos en orden y sin declararlos:

```
cur.execute("""  
    INSERT INTO users (id, fecha_primer_inicio_sesion, username)  
    VALUES (%s, %s, %s);  
""",  
    (10, datetime.date(2020, 10, 18), "dawda"))
```

# Pasar parámetros (con nombres)

Si queremos declarar y nombrar a los argumentos que ingresemos:

```
cur.execute("""  
    INSERT INTO users (id, fecha_primer_inicio_sesion, username)  
    VALUES (%(id)s, %(fecha_inicio)s, %(usuario)s);  
""",  
    {'id': 10, 'fecha_inicio': datetime.date(2020, 10, 18),  
    'usuario': "dawda"})
```

# OJO

Si no declaramos argumentos como en la anterior viñeta y si solo insertamos un argumento en el execute, entonces no nos olvidemos de cerrar bien el paréntesis y colocar una ‘,’ luego del argumento para insertar:

```
cur.execute("INSERT INTO some_table VALUES (%s)", ("adawd")) # INCORRECTO
```

```
cur.execute("INSERT INTO some_table VALUES (%s)", ("adawd",)) # CORRECTO
```

# Inyección SQL

## NUNCA HACER ESTO: (No concatenar)

```
SQL = "INSERT INTO usuarios (name) VALUES ('%s')"  
data = ("Jaime", )  
cur.execute(SQL % data)
```

## No usar comillas ni usar el operador %.

### Otro error:

```
#OTRO ERROR:  
query = "SELECT * FROM users WHERE name = " + nombre  
cursor.execute(query)
```

## Forma correcta

Para evitar inyecciones, cambiamos lo siguiente en el anterior código. No utilizamos las comillas en el parámetro y no usamos el operador %.

```
SQL = "INSERT INTO usuarios (name) VALUES (%s)"
```

```
data = ("Jaime", )
```

```
cur.execute(SQL, data)
```

# HTML y Python (archivo html)

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulario</title>
</head>
<body>
  <h2>Formulario de Consulta</h2>
  <form action="consulta.py" method="post">
    Nombre: <input type="text" name="nombre"><br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```

# HTML y Python (archivo .py) (Posible Forma)

```
import cgi
import psycopg2
conn = psycopg2.connect(dbname="grupoXXeN", user="grupoXX",
password="contraseña", host="localhost", port=5432)
cursor = conn.cursor()
form = cgi.FieldStorage()
nombre = form.getvalue('nombre')
cursor.execute("SELECT * FROM tabla WHERE nombre = %s",
(nombre,))
resultados = cursor.fetchall()
cursor.close()
conn.close()
```