



**Universidad Nacional  
Autónoma de México**  
Facultad de Ingeniería  
División de Ingeniería Mecánica e  
Industrial



**Laboratorio de Cómputo de Ingeniería Mecatrónica  
(1472)**

*Profesor: Miguel Serrano Reyes  
Semestre 2021-1*

**Práctica No. 3**

Nombre de la práctica

Base de Datos Mysql

Grupo: 2

**Nombre del Estudiante:**  
**Celaya González David Aleajandro**

```

1#!/usr/bin/env python3
2# -*- coding: utf-8 -*-
3import requests
4import pandas as pd
5from bs4 import BeautifulSoup
6import mysql.connector as mysql
7
8
9req = requests.get("https://es.wikipedia.org/wiki/M%C3%A9xico")
10soup = BeautifulSoup(req.content,"html.parser")
11tablas = soup.find("table",{"class":"wikitable"})
12df = pd.read_html(str(tablas))[0]
13
14
15df = df.drop(columns=1)
16df = df.drop(columns=3)
17df = df.drop(columns=5)
18df=df.drop([0,1])
19df.columns=["Estado","Capital","Poblacion"]
20
21lista = []
22slista = []
23
24for i in range(2,34):
25    slista.append(df.Estado[i])
26    slista.append(df.Capital[i])
27    slista.append(df.Poblacion[i].replace(u'\xa0', u'))
28    lista.append(slista)
29slista=[]
--

```

Como primera parte es importante llamar a las librerías que se necesitaran, en este caso se utiliza pandas y bs4 para poder hacer Scraping de la pagina web de donde extraeremos los datos. Mysql será otra librería la cual tendrá que ver con la base de datos.

En esta primera parte se realizó una filtración del código html de la página, se examina para poder eliminar las columnas que no queremos de la tabla original y con ello realizamos una tabla a nuestra conveniencia que únicamente obtendrá “Estados”, “Capital” y “Población”. Esta información la almacenaremos en una sublista dentro de una lista para poder tener un manejo más práctico en la base de datos.

```

In [133]: runfile('/home/david/Escritorio/Scraping.py', wdir='/home/david/Escritorio')
<mysql.connector.connection.MySQLConnection object at 0x7f0b9cedf358>
[('information_schema',), ('Estados_CGDA',), ('mysql',), ('performance_schema',), ('sys',)]

```

```

31
32 db = mysql.connect(host="localhost",user="root",passwd="ToroRoso")
33 print(db)
34 cursor=db.cursor()
35
36
37 cursor.execute("SHOW DATABASES")
38
39 databases=cursor.fetchall()
40 print(databases)
41

```

En esta siguiente parte del código accedemos a nuestro localhost con el usuario “root” para después pedir que se nos muestre todas las bases de datos relacionadas con ese usuario o a las que ese usuario tenga acceso.

```

41
42 #Creamos la base de datos
43 #cursor.execute("CREATE DATABASE Estados_CGDA")
44
45 db = mysql.connect(host="localhost",user="root",passwd="ToroRoso",database="Estados_CGDA")
46 cursor=db.cursor()
47
48 #cursor.execute("CREATE TABLE ESTADOS4(ID INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, ESTADO VARCHAR(255),CAPITAL VARCHAR(255), POBLACION VARCHAR(255))")
49
50 #MUESTRA TODAS LAS TABLAS
51
52 cursor.execute("SHOW TABLES")
53 tablas=cursor.fetchall()
54 for tabla in tablas:
55     print(tabla)
56
57 cursor.execute("DESC ESTADOS4")
58 print(cursor.fetchall())
59

```

En esta parte creamos nuestra base de datos llamada “Estados\_CGDA” para despues crear un objeto db pero accediendo a la misma. Lo siguiente fue crear una tabla llamada “ESTADOS4” con los ‘ID’ correspondientes y los diferentes datos que esta llevaria.

Continuando en el codigo una vez hecha la tabla, pedimos que nos muestre todas las tablas que ya existen esto con la finalidad de ver que se haya creado.

Una vez que nosotros ya confirmamos que nuestra tabla ha sido creada, podemos pedir una descripción de la misma.

```

In [134]: runfile('/home/david/Escritorio/Scraping.py', wdir='/home/david/Escritorio')
mysql.connector.connection.MySQLConnection object at 0x7f0b9a05d128>
[('information_schema',), ('Estados_CGDA',), ('mysql',), ('performance_schema',), ('sys',)]
('ESTADOS',)
('ESTADOS2',)
('ESTADOS3',)
('ESTADOS4',)
('ID', 'int(11)', 'NO', 'PRI', None, 'auto_increment'), ('ESTADO', 'varchar(255)', 'YES', '',
None, ''), ('CAPITAL', 'varchar(255)', 'YES', '', None, ''), ('POBLACION', 'varchar(255)', 'YES',
'', None, '')]

```

```

60 #Inserta información
61
62
63 query="INSERT INTO ESTADOS4 (Estado,Capital,Poblacion) VALUES (%s,%s,%s)"
64 i=0 #Nos dara iteraciones, no se te olvide
65 for slista in lista:
66     values=[(slista[0],slista[1],slista[2])]
67     cursor.executemany(query,values)
68     db.commit()
69     i+=1
70
71 print("AGREGUE",i,"DATOS NUEVOS")
72
73 #Consultando tabla
74
75 query="SELECT * FROM ESTADOS4"
76 cursor.execute(query)
77 registros=cursor.fetchall()
78 for registro in registros:
79     print(registro)

```

Es momento de insertar los datos extraidos con Scraping, es en esta parte donde las sublistas nos ayudaran a tener un manejo de información que nos resulte mas comodo y facil. También es donde aparece un concepto clave en las bases de datos y es “query” o consulta. Sus multiples funciones nos ayudaran a navegar e iterar entre las tablas que existan o creamos nosotros, en este caso en especifico agregamos los datos y despues consultamos la tabla para confirmar que los datos se hayan escrito con éxito.

```

[('ID', 'int(11)', 'NO', 'PRI', None, 'auto_increment'), ('ESTADO', 'varchar(255)', 'YES', '', None, ''), ('CAPITAL', 'varchar(255)', 'YES', '', None, ''), ('POBLACION', 'varchar(255)', 'YES', '', None, '')]
AGREGUE 32 DATOS NUEVOS
(1, 'Entidad federativa', 'Capital', 'Población')
(2, 'Aguascalientes', 'Aguascalientes', '1184996')
(3, 'Baja California', 'Mexicali', '3155870')
(4, 'Baja California Sur', 'La Paz', '637026')
(5, 'Campeche', 'San Francisco de Campeche', '822441')
(6, 'Chiapas', 'Tuxtla Gutiérrez', '4796580')
(7, 'Chihuahua', 'Chihuahua', '247487')
(8, 'Ciudad de México', 'No aplica', '1479')
(9, 'Coahuila', 'Saltillo', '151571')
(10, 'Colima', 'Colima', '650555')
(11, 'Durango', 'Victoria de Durango', '1632934')
(12, 'Guanajuato', 'Guanajuato', '5486372')
(13, 'Guerrero', 'Chilpancingo de los Bravo', '3388768')
(14, 'Hidalgo', 'Pachuca de Soto', '2665018')
(15, 'Jalisco', 'Guadalajara', '7350682')
(16, 'México', 'Toluca de Lerdo', '22333')
(17, 'Michoacán', 'Morelia', '58667')
(18, 'Morelos', 'Cuernavaca', '1777227')
(19, 'Nayarit', 'Tepic', '1084979')
(20, 'Nuevo León', 'Monterrey', '4653458')
(21, 'Oaxaca', 'Oaxaca de Juárez', '3801962')
(22, 'Puebla', 'Heroica Puebla de Zaragoza', '34251')
(23, 'Querétaro', 'Santiago de Querétaro', '11658')
(24, 'Quintana Roo', 'Chetumal', '1325578')
(25, 'San Luis Potosí', 'San Luis Potosí', '2585518')
(26, 'Sinaloa', 'Cullacán Rosales', '2767761')
(27, 'Sonora', 'Hermosillo', '2662480')
(28, 'Tabasco', 'Villahermosa', '24747')
/28 'Tampulco', 'Ciudad Uxtepec', '18440')

```

```

81 #Muestra los estados con población mayor a 2 millones de habitantes
82
83 query="SELECT Estado,Capital FROM ESTADOS4 WHERE Poblacion>2000000"
84 cursor.execute(query)
85 registros=cursor.fetchall()
86 for registro in registros:
87     print(registro)
88

```

Por ultimo vuelvo a utilizar query para poder filtrar aquellos estados de México que su población sea mayor a dos millones de habitantes.