

Proyecto de seguimiento casos de covid Colombia

Jairo Antonio Caro Vanegas
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda - Bogotá, Colombia
Jairo.car01@usa.edu.co

Resumen

Con este proyecto se busca darle seguimiento al covid-19 en Colombia por medio del web scraping tomaremos datos de una página web en este caso Wikipedia y los almacenaremos en una base de datos MySQL y serán graficados en Python.

Palabras clave:

Python 1, MySQL 2, Covid 3, Colombia 4.

1. Marco teórico

El Web scraping es un método de obtención de datos de una página web en formato html con el fin de poderlos usar en python.

2. Resultados

Como primera parte del proyecto se busco obtener los datos de la página wikipedia para eso utilizamos el siguiente código.

```
url='https://es.wikipedia.org/wiki/  
Pandemia_de_enfermedad_por_coronavirus_de_2020_en-Colombia'  
req=requests.get(url)  
soup=BeautifulSoup(req.content,'html.parser')  
datos=soup.find_all('th')  
total=list()  
for i in datos:  
    total.append(i.text)  
Casos=str(total[89])  
Recuperados=str(total[50])  
Hospitalizados=str(total[56])  
Unidades=str(total[59])  
Muertes=str(total[62])  
print(Casos,Recuperados,Hospitalizados,Unidades,Muertes)
```

En este código lo que estamos haciendo es llamar a la página y obtener todos los archivos "th" para poder guardarlos en una lista luego de guardarlos los buscamos en la lista y se los asignamos a cada variable.

En la segunda parte creamos la conexión a la base de datos donde se almacenaran a diario los datos actualizados de casos, muertes, hospitalizados y en UCIS

```
connection = pymysql.connect(host='localhost',user='root',  
,password='1234',db='seguimineto',charset='utf8mb4', cursorclass=pymysql.cursors.DictCursor)  
try:  
    with connection.cursor() as cursor:  
        sql = "select fecha from total where fecha="+fecha+";"  
        cursor.execute(sql)  
        result = cursor.fetchone()  
    if result == None:  
        with connection.cursor() as cursor:  
            sql = "insert into total values("+fecha+", "+Casos+", "+Recuperados+",  
            , "+Hospitalizados+", "+Unidades+", "+Muertes+");"
```

```

        cursor.execute(sql)
        connection.commit()

finally:
    connection.close()

```

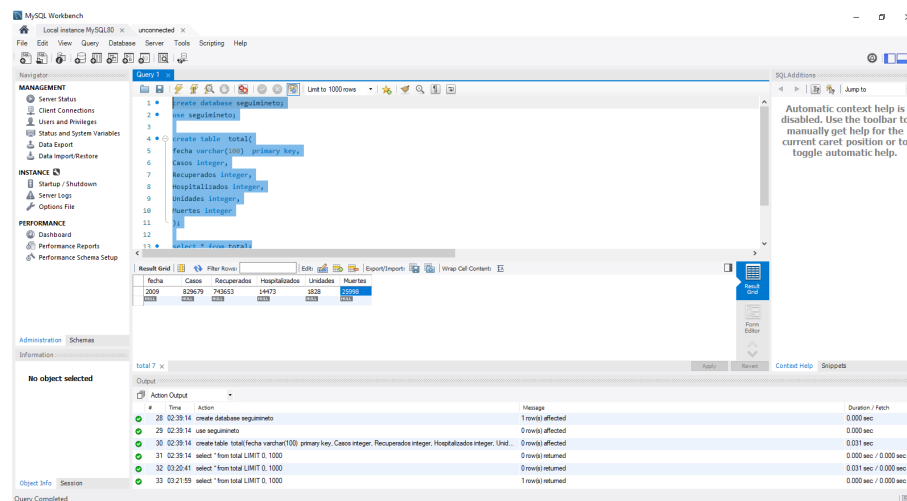
La base de datos que tenemos configurada contara con el siguiente codigo

```

create database seguimineto;
use seguimineto;
create table total(
fecha varchar(100) primary key,
Casos integer,
Recuperados integer,
Hospitalizados integer,
Unidades integer,
Muertes integer
);
select * from total;

```

Aquí podemos validar que los archivos se están guardando correctamente en la base de datos La ultima parte de la codificación



(a)

Figura 1: Base de datos covid

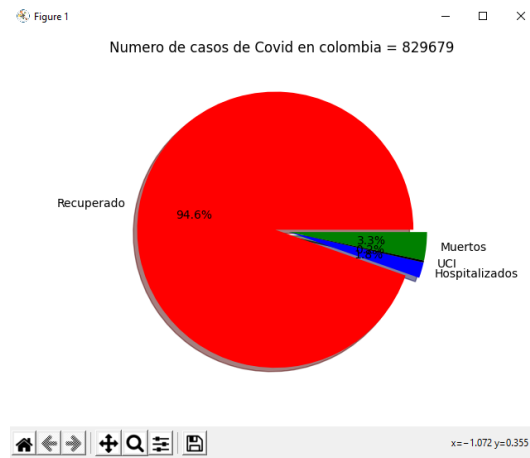
se crearon tres graficas con los datos obtenidos dos de barras y una de pie.

```

estado=('Recuperado','Hospitalizados','UCI','Muertos')
slices=(Recuperados,Hospitalizados,Unidades,Muertes)
colores=('red','blue','black','green')
Valores=(0.1,0,0,0)
pyplot.pie(slices,color=colores,labels=estado,autopct='%1.1f%%',explode=Valores,shadow=True)
pyplot.axis('equal')
pyplot.title('Numero de casos de Covid en colombia = '+Casos)
pyplot.show()
slices1=(Recuperados,Hospitalizados,Unidades,Muertes)
pyplot.title('Numero de casos de Covid en colombia = '+Casos)
pyplot.bar(estado,slices1,color=colores)
pyplot.axis('equal')
pyplot.show()
pyplot.title('Numero de casos de Covid en colombia = '+Casos)
pyplot.barh(range(4),slices1,color=colores)
pyplot.yticks(range(4),estado, rotation=60)

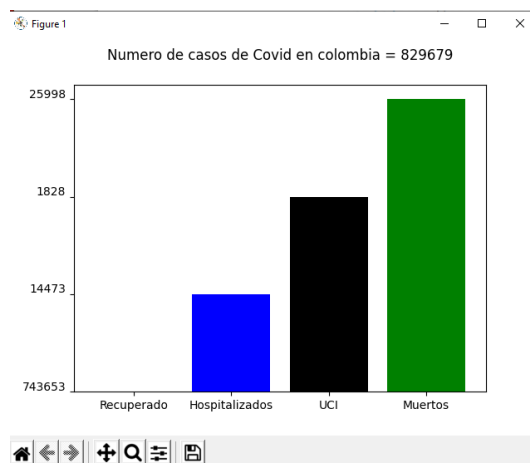
```

```
pyplot.axis('equal')
pyplot.show()
```



(a)

Figura 2: Diagrama de pie



(a)

Figura 3: Diagrama de barras

Para la segunda parte del proyecto usamos tres nuevas paginas web para poder extraer los datos de las localidades y de los departamentos en colombia

```
url_1='https://canaltrece.com.co/noticias/cuantos-casos-coronavirus-covid-19-bogota-barrios-1'
url_2='https://colombia.as.com/colombia/2020/10/29/actualidad/1603970788_671558.html'
url_3='https://es.wikipedia.org/wiki/Pandemia_de_enfermedad_por_coronavirus_de_2020_en_Colomb'
page_1=requests.get(url_1)
page_2=requests.get(url_2)
page_3=requests.get(url_3)
soup_1=BeautifulSoup(page_1.content,'html.parser')
soup_2=BeautifulSoup(page_2.content,'html.parser')
soup_3=BeautifulSoup(page_3.content,'html.parser')
contagios_bogota=soup_1.find_all('li')
contagios_localidad=soup_1.find_all('strong')
contagios_colombia=soup_2.find_all('span')
```

```

datos=soup_3.find_all('th')
Total_cbta=list()
Total_cbta_1=list()
Total_ccol=list()
total=list()
#Web scraping por localidades Bogotá
for i in contagios_bogota:
    Total_cbta.append(i.text)
for i in Total_cbta:
    if(recorrer<19):
        for e in i:
            if(e==':') :
                espacio=1
            if((espacio==1) and (e!=(".")) and (e!=':') and (e!=" ") and (e!=(' \xa0'))):
                longitud=longitud+1
                #print(e)
                localidades.insert(0,e)
            #print('\n'+str(longitud)+'\n')
            aux.insert(0,longitud)
            espacio = 0
            longitud=0
            recorrer+=1
recorrer=18
vaus=86
otraux=86
while(recorrer>=0):
    otraux=otraux-aux[recorrer]
    if(aux[recorrer]==5):
        auxsuma=10000
    if(aux[recorrer]==4):
        auxsuma=1000
    while(vaus>otraux):
        valor=valor+auxsuma*float(localidades[vaus])
        vaus-=1
        auxsuma=auxsuma/10
    recorrer-=1
    Cpl.insert(0,valor)
    valor=0
print(Cpl)
for i in contagios_localidad:
    Total_cbta_1.append(i.text)
for o in Total_cbta_1:
    if((cont>=5) and (cont<24)):
        localidades_nom.insert(0,Total_cbta_1[cont])
    cont+=1
print(localidades_nom)
print('\n')
#Web scraping por Departamentos Colombia

for i in contagios_colombia:
    Total_ccol.append(i.text)
print("Contagios hoy "+str(Total_ccol[82]))
print("Contagios Totales "+str(Total_ccol[83]))
print("Muertes hoy "+str(Total_ccol[84]))
print("Total Muertes "+Total_ccol[86])
print("Recuperados hoy "+str(Total_ccol[85]))
print("Total Recuperados "+Total_ccol[87])
print("Activos "+Total_ccol[88])

```

```

print("Muestras procesadas "+Total_ccol[89])
print("\n")
for i in Total_ccol:
    if((conteo>89)and(conteo<156)):
        if((conteo%2)==(0)):
            departamentos.insert(0,Total_ccol[conteo])
            conteoaux+=1
        else:
            auxdato=1000
            auxdato=float(Total_ccol[conteo])*1000
            cpd.insert(0,auxdato)

    conteo+=1
cpd[0]=cpd[0]/1000
print(departamentos)
print(cpd)

```

En el programa se entregan las coordenadas de cada de una de las capitales de los departamentos para que después de creado el radio dependiente de los contagios este quede centrado, una vez creado el mapa este se guardara en la ruta C:

Users
 jacar
 OneDrive
 Documentos
 python
 se nales
 Colombia.html”

```

vconteo=0
vaux=0
radio=[]
colores_1=''
poscircuitos=[[4.643493, -74.097520],[6.217,-75.567],[3.42158,-76.5205],[10.9878,-74.7889],[
4.4259,-74.1243],[7.11392,-73.1198],
[ 8.817,-74.717],[8.75,-75.883],[10.45,-73.2510],
[4.15,-73.633],[1.2136,-77.2811],[7.9,-72.57],[2.92504,-75.2897],
[4.433,-75.217],[11.1450,-74.1206],[4.813,-75.694],
[9.3,-75.491],[5.533,-73.367],[2.433,-76.617],[5.067,-75.517],
[1.612,-75.6],[11.533,-72.911],[4.532,-75.652],[5.35,-72.452],
[1.159,-76.647],[5.683,-76.655],[7.083,-70.757],[-4.208,-69.943],
[12.537,-81.7313],[2.567,-72.633],[3.867,-67.917]
,[1.255, -70.235],[4.433,-69.84]]

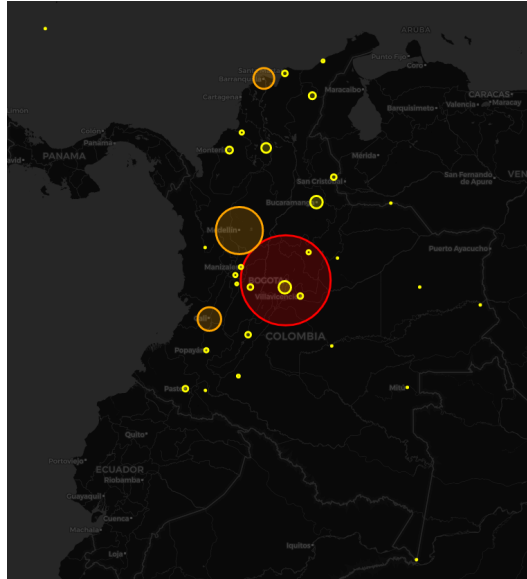
for i in cpd:
    vaux=i/2
    radio.insert(0,vaux)
print(radio)
#Mapa de Colombia
Colombia=folium.Map(location=[4.643493, -74.097520],zoom_start=5)
folium.raster_layers.TileLayer('Open Street Map').add_to(Colombia)
folium.raster_layers.TileLayer('Stamen Terrain').add_to(Colombia)
folium.raster_layers.TileLayer('Stamen Toner').add_to(Colombia)
folium.raster_layers.TileLayer('Stamen Watercolor').add_to(Colombia)
folium.raster_layers.TileLayer('CartoDB Positron').add_to(Colombia)
folium.raster_layers.TileLayer('CartoDB Dark_Matter').add_to(Colombia)
folium.LayerControl(position='topright',collapsed=False).add_to(Colombia)
#Circuitos
while(vconteo<33):
    if((radio[vconteo]<150000)and(radio[vconteo]>30000)):
        colores_1='orange'
    elif(radio[vconteo]<30000):

```

```

        colores_1='yellow'
    else:
        colores_1='red'
    folium.Circle(radius=radio[vconteo], location=poscirculos[vconteo], color=colores_1, fill
    vconteo+=1
Colombia.save("C:\\Users\\jacar\\OneDrive\\Documentos\\python\\señales\\Colombia.html")

```



(a)

Figura 4: Mapa de calor casos covid departamentos Colombia

Para la creación del mapa de Bogotá se hizo lo mismo que en el mapa de Colombia se le entregaron la coordenadas de cada una de las localidades de Bogotá, se creo el radio dependiendo de los casos confirmados y se guardo el mapa en la ruta establecida en el programa.

```

Bogota=folium.Map(location=[4.643493, -74.097520], zoom_start=11)
folium.raster_layers.TileLayer('Open Street Map').add_to(Bogota)
folium.raster_layers.TileLayer('Stamen Terrain').add_to(Bogota)
folium.raster_layers.TileLayer('Stamen Toner').add_to(Bogota)
folium.raster_layers.TileLayer('Stamen Watercolor').add_to(Bogota)
folium.raster_layers.TileLayer('CartoDB Positron').add_to(Bogota)
folium.raster_layers.TileLayer('CartoDB Dark Matter').add_to(Bogota)
folium.LayerControl(position='topright', collapsed=False).add_to(Bogota)
radio_2=[]
v2conteo=0
colores_2=''
poscirculos_1=[[4.56619,-74.16198],[4.565,-74.116],
                [4.594,-74.074],[4.61341,-74.10623],[4.58940,-74.10472],
                [4.603,-74.091],
                [4.645,-74.094],[4.66405,-74.07501],[4.72859,-74.08219],
                [4.707,-74.107],[4.67472,-74.13223],[4.627,-74.157],
                [4.631,-74.195],[4.57228,-74.13507],[4.51465,-74.09739],
                [4.55840,-74.08884],[4.610,-74.070],[4.657,-74.047],
                [4.71199,-74.04306]]

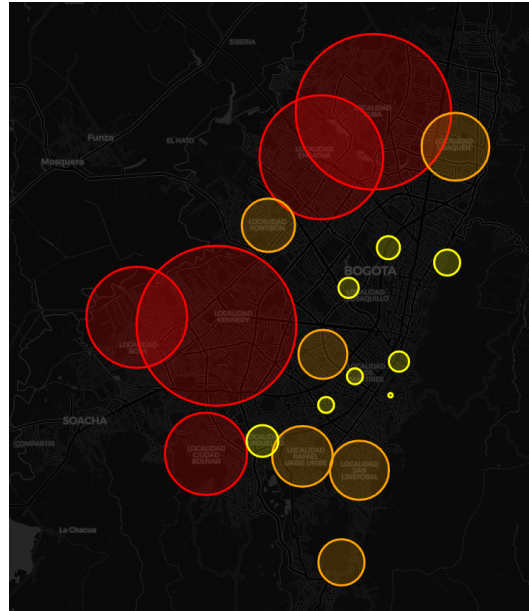
v2aux=0
for i in Cpl:
    v2aux=i/10
    radio_2.insert(19,v2aux)
print(radio_2[0])

```

```

print(poscirculos_1[0])
while(v2conteo<19):
    if((radio_2[v2conteo]<2000)and(radio_2[v2conteo]>1000)):
        colores_2='orange'
    elif(radio_2[v2conteo]<1000):
        colores_2='yellow'
    else:
        colores_2='red'
    folium.Circle(radius=radio_2[v2conteo], location=poscirculos_1[v2conteo], color=colores_2)
    v2conteo+=1
Bogota.save("C:\\Users\\jacar\\OneDrive\\Documentos\\python\\señales\\Bogota.html")

```



(a)

Figura 5: Mapa de calor casos covid localidades Bogotá

Para finalizar se guardaron los datos en una base de datos local.

```

try:
    cnx = mysql.connector.connect(user='root', password='1234', database='segimiento', host='127.0.0.1')
    cursor = cnx.cursor()
    print('Conectado')
    for i in range(len(Cpl)):
        sentencia = ("insert into localidades (fecha,nombre,casos) values (%s,%s,%s);")
        nombre= str(localidades_nom[i])
        casos= str(Cpl[i])
        datos =(fecha,nombre,casos)
        cursor.execute(sentencia,datos)
        cnx.commit()

except mysql.connector.Error as err:
    if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong with your user name or password")
    elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exist")
    else:
        print(err)
else:
    print('')

```

```

cnx.close()

try:
    cnx = mysql.connector.connect(user='root', password='1234', database='segimiento', host='127.0.0.1')
    cursor = cnx.cursor()
    print('Conectado')
    for i in range(len(cpd)):
        sentencia = ("insert into depa(fecha,nombre,casos) values (%s,%s,%s);")
        nombre= str(departamentos[i])
        casos= str(cpd[i])
        datos =(fecha,nombre,casos)
        cursor.execute(sentencia,datos)
    cnx.commit()

except mysql.connector.Error as err:
    if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong with your user name or password")
    elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exist")
    else:
        print(err)
else:
    cnx.close()

```

(a)

3. Conclusiones

4. Referencias