

```
In [1]: 1 import requests
        2 from bs4 import BeautifulSoup
        3 import re
```

```
In [2]: 1 from neomodel import (config, StructuredNode, StringProperty, IntegerProperty,
        2     UniqueIdProperty, RelationshipTo, RelationshipFrom)
        3
        4 config.DATABASE_URL = 'bolt://neo4j:Angel2019@localhost:7687'
        5
        6 class Asambleista(StructuredNode):
        7     nameCan = StringProperty(unique_index=True)
        8     partido = StringProperty(unique_index=True)
        9 class Noticia(StructuredNode):
       10     idNoti = UniqueIdProperty()
       11     date = StringProperty(unique_index=False)
       12     mensaje = StringProperty(unique_index=False)
       13     link = StringProperty(unique_index=False)
       14     asambleista = RelationshipTo('Asambleista', 'NOTICIA_DE')
       15
```

In [15]:

```

1  #Datos url google, seccion noticias
2
3  start_url = ['https://www.google.com/search?q=Juan+Cristobal+Lloret+Valdivie
4              'https://www.google.com/search?q=Juan+Cristobal+Lloret+Valdivie
5              ]
6  #print(nurl)
7  asamb = Asambleista(nameCan='Juan Cristobal Lloret Valdivieso', partido='UNE
8
9  c=1
10 print("Contenido de pagina google candidato asambleista Juan Lloret")
11 for ur in start_url:
12     url = ur
13     respuesta = requests.get(
14         url,
15         headers = {
16             'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
17         }
18     )
19     contenido_web = BeautifulSoup(respuesta.text, 'lxml')
20     noticias = contenido_web.find('div', attrs={'class': 'eqAnXb'})
21     lin = noticias.findChildren('div', attrs={'class': 'dbsr'})
22
23     for enlace in lin:
24         enla=str(enlace.a.get('href'))
25
26         men = ""
27         sms1 = enlace.findChildren('div', attrs={'class': 'yJHHTd'})
28         for message1 in sms1:
29             sms2 = message1.findChildren('div', attrs={'class': 'Y3v8qd'})
30             men = str(sms2)
31
32         #Para limpiar Las etiquetas
33         cleantext = BeautifulSoup(men, "lxml").text
34         mens1 = cleantext.strip("[").strip("]")
35
36         #print("Fecha")
37         fecha1=str(enlace.findChildren('span', attrs={'class': 'WG9SHc'}))
38         fecclear = BeautifulSoup(fecha1, "lxml").text
39         #Variable de fecha
40         fecha2 = fecclear.strip("[").strip("]")
41
42
43         noticia = Noticia(idNoti=c,date=fecha2, mensaje=mens1, link=enla).s
44         noticia.asambleista.connect(asamb)
45         print("Datos guardados")
46         c=c+1
47

```

Contenido de pagina google candidato asambleista Juan Lloret

Datos guardados

Datos guardados

Datos guardados

Datos guardados

Datos guardados

Datos guardados

Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados

In [16]:

```

1  # Datos candidato presidente ecosia.
2  print("Contenido de pagina ecosia de candidato Andres Arauz")
3  cont=0
4  for i in range(5):
5      print('Item')
6      print(cont)
7      url = 'https://www.ecosia.org/news?q=Andres%20Arauz&p='+str(cont)
8      respuesta = requests.get(
9          url,
10         headers = {
11             'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
12         }
13     )
14     myData = []
15     contenido_web = BeautifulSoup(respuesta.text, 'lxml')
16     noticias = contenido_web.find('div', attrs={'class':'layout'})
17     sections = noticias.findChildren('div', attrs={'class':'result__body'})
18     cont=cont+1
19     for section in sections:
20         #Codigo para grafo
21         codigo = "unesecosia"+str(c)
22         #print('Link:')
23         link = section.a.get('href')
24         clink = BeautifulSoup(str(link), "lxml").text
25
26         #Link limpio en variable
27         clink = str(clink)
28
29         hora = section.findChildren('time', attrs={'class':'news-result__dat
30         chora = BeautifulSoup(str(hora), "lxml").text
31         #Hora limpia en variable
32         schora = (chora.strip("[").strip("]").rstrip().lstrip())
33
34         sms = section.findChildren('div', attrs={'class':'result__content-in
35         csms = BeautifulSoup(str(sms), "lxml").text
36
37         #Mensaje limpio en variable
38         scsms =(csms.strip("[").strip("]").strip('')).rstrip().lstrip()
39
40         noticia = Noticia(idNoti=c,date=schora, mensaje=scsms, link=clink).
41         noticia.asambleista.connect(asamb)
42         print("Datos guardados")
43         c=c+1

```

Contenido de pagina ecosia de candidato Andres Arauz

Item

0

Datos guardados

Datos guardados

Datos guardados

Datos guardados

Datos guardados

Datos guardados

Datos guardados

Datos guardados

```
Datos guardados
Item
1
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Item
2
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Datos guardados
Item
3
Datos guardados
Datos guardados
Datos guardados
Item
4
Datos guardados
Datos guardados
Datos guardados
```

```

In [17]: 1 cont=1
2 print("Contenido de pagina ask.")
3 for i in range(3):
4     print('Page')
5     print(cont)
6     url = 'https://www.ask.com/web?o=0&l=dir&qo=pagination&q=andres%20arauz%'
7     respuesta = requests.get(
8         url,
9         headers = {
10             'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
11         }
12     )
13     contenido_web = BeautifulSoup(respuesta.text, 'lxml')
14     noticias = contenido_web.find('div', attrs={'PartialSearchResults-body'})
15     seccion = noticias.findChildren('div', attrs={'class': 'PartialSearchRes
16
17     for item in seccion:
18         print("Item")
19
20         tit = str(item.find('div').a)
21         titulo = BeautifulSoup(tit, "lxml").text
22         print("Pagina: "+titulo)
23
24         link = str(item.find('div').a.get("href"))
25         print("Link: "+link)
26
27         sms = str(item.find('p', attrs = {'class': 'PartialSearchResults-item
28         sms1 = BeautifulSoup(sms, "lxml").text
29         print("Mensaje: "+sms1)
30
31         noticia = Noticia(idNoti=c,date=titulo, mensaje=sms1, link=link).sa
32         noticia.asambleista.connect(asamb)
33         print("Datos guardados")
34         c=c+1
35     cont=cont+1

```

[gory/noticias/](#))

Mensaje: Andrés Arauz hace un llamado a la comunidad internacional a mantene
rse

vigilante del proceso electoral en Ecuador. Home; /; Noticias ...

Datos guardados

Item

Pagina: Andrés Arauz, el economista que quiere emular a Rafael Correa

Link: <https://www.primicias.ec/noticias/politica/andres-arauz-perfiles-candi-datos-elecciones/> (<https://www.primicias.ec/noticias/politica/andres-arauz-p-erfiles-candidatos-elecciones/>)

Mensaje: Ofrecemos disculpas a nuestros lectores. Noticias relacionadas. Can
didatos que

cantan, bailan, se pelean, hacen donaciones y ofertas. Política.

Datos guardados

Item

Pagina: Andrés Arauz no participará en debate presidencial ... - El Telégraf
o

Link: <https://www.eltelegrafo.com.ec/noticias/politica/3/andres-arauz-debate-presidencial> (<https://www.eltelegrafo.com.ec/noticias/politica/3/andres-ara-uz-debate-presidencial>)

In [18]:

```

1  #Datos de noticias google, candidato presidente Andres Arauz
2  print("Datos de noticias google, candidato presidente Andres Arauz")
3  item=0
4  con=1
5  cont=0
6  for i in range(19):
7      url = url = 'https://www.google.com/search?q=andres+arauz&tbm=nws&ei=PY
8      respuesta = requests.get(
9          url,
10         headers = {
11             'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
12         }
13     )
14     contenido_web = BeautifulSoup(respuesta.text, 'lxml')
15     noticias = contenido_web.find('div', attrs={'class': 'eqAnXb'})
16     lin = noticias.findChildren('div', attrs={'class': 'dbsr'})
17     cont=cont+10
18     item=item
19     print("Enlace: ")
20     print(url)
21     for enlace in lin:
22         item=item+1
23         enla=str(enlace.a.get('href'))
24         print("Enlace")
25         print(enla)
26         #print('Mensaje')
27         men = ""
28         sms1 = enlace.findChildren('div', attrs={'class': 'yJHHTd'})
29         for message1 in sms1:
30             sms2 = message1.findChildren('div', attrs={'class': 'Y3v8qd'})
31             men = str(sms2)
32
33         #Para limpiar las etiquetas
34         print("Mensaje")
35         cleantext = BeautifulSoup(men, "lxml").text
36         mens1 = cleantext.strip("(").strip(")")
37         #print(mens1)
38
39         print("Fecha")
40         fecha1=str(enlace.findChildren('span', attrs={'class': 'WG9SHc'}))
41         fecclear = BeautifulSoup(fecha1, "lxml").text
42         #Variable de fecha
43         fecha2 = fecclear.strip("(").strip(")")
44         print(fecha2)
45
46         noticia = Noticia(idNoti=c,date=fecha2, mensaje=mens1, link=enla).s
47         noticia.asambleista.connect(asamb)
48         print("Datos guardados")
49
50         print(c)
51         c = c + 1
52
53     con=con+1

```

Enlace

<https://www.larepublica.ec/blog/2021/01/02/arauz-dice-pulmon-izquierdo-compl>

ico-coronavirus-esta-bien/ (<https://www.larepublica.ec/blog/2021/01/02/arauz-dice-pulmon-izquierdo-complico-coronavirus-esta-bien/>)

Mensaje

Fecha

Hace 1 semana

Datos guardados

80

Enlace

<https://www.elcomercio.com/actualidad/resumen-primer-dia-debate-candidatos.html> (<https://www.elcomercio.com/actualidad/resumen-primer-dia-debate-candidatos.html>)

Mensaje

Fecha

hace 2 días

Datos guardados

81

Enlace