



API e Web Scraping

Robson Silva da Silva

Resumo

- Requisição de dados via API no portal transparência do Governo Federal;
- Web Scraping no portal do FBI dos EUA;
- Web Scraping no portal de vagas do Nubank.

 swagger

default (/v2/api-docs) ▾

Explorar

API REST do Portal da Transparência do Governo Federal

API de serviços do Portal da Transparência do Governo Federal

Criado por Diretoria de Tecnologia da Informação - DTI
Veja mais em <http://www.cgu.gov.br>
[Contate o desenvolvedor](#)
[Decreto nº 8.777, de 11 de maio de 2016](#)

Acordos de Leniência

Mostrar/Esconder | Listar operações | Expandir operações

Benefício de Prestação Continuada (BPC)

Mostrar/Esconder | Listar operações | Expandir operações

Bolsa Família

Mostrar/Esconder | Listar operações | Expandir operações

GET

/api-de-dados/bolsa-familia-disponivel-por-cpf-ou-nis

Consulta as parcelas disponibilizadas pelo Bolsa Família pelo CPF/NIS

GET

/api-de-dados/bolsa-familia-por-municipio

Consulta as parcelas do Bolsa Família por Município

Notas de Implementação

Filtros mínimos: Página (padrão = 1); Ano/Mês (YYYYMM); Código IBGE (<https://cidades.ibge.gov.br/brasil>);

Classe de resposta (Status 200)

OK

Modelo

Example Value

[

Coletando Dados para Requisições

```
In [3]: 1 #Coletando todas as capitais e colocando numa lista.
2 capitais = ['Rio Branco', 'Maceió', 'Macapá', 'Manaus', 'Salvador', 'Fortaleza',
3             'Brasília', 'Vitória', 'Goiânia', 'São Luís', 'Cuiabá', 'Campo Grande',
4             'Belo Horizonte', 'Belém', 'João Pessoa', 'Curitiba', 'Recife', 'Teresina',
5             'Rio de Janeiro', 'Natal', 'Porto Alegre', 'Porto Velho', 'Boa Vista',
6             'Florianópolis', 'São Paulo', 'Aracaju', 'Palmas']

In [4]: 1 # Coletei na internet os códigos das capitais para a requisição e coloquei em um dicionário.
2 capitaiscodeIBGE = {'Rio Branco': 1200401, 'Maceió': 2704302, 'Macapá': 1600303,
3                     'Manaus': 1302603, 'Salvador': 2927408, 'Fortaleza': 2304400,
4                     'Brasília': 5300108, 'Vitória': 3205309, 'Goiânia': 5208707,
5                     'São Luís': 2111300, 'Cuiabá': 5103403, 'Campo Grande': 5002704,
6                     'Belo Horizonte': 3106200, 'Belém': 1501402, 'João Pessoa': 2507507,
7                     'Curitiba': 4106902, 'Recife': 2611606, 'Teresina': 2211001,
8                     'Rio de Janeiro': 3304557, 'Natal': 2408102, 'Porto Alegre': 4314902,
9                     'Porto Velho': 1100205, 'Boa Vista': 1400100, 'Florianópolis': 4205407,
10                    'São Paulo': 3550308, 'Aracaju': 2800308, 'Palmas': 1721000}

In [5]: 1 #lista com os padrão de requisição de mês e ano.
2 month_2018 = [201801, 201802, 201803, 201804, 201805, 201806, 201807, 201808, 201809,
3              201810, 201811, 201812]
4 month_2019 = [201901, 201902, 201903, 201904, 201905, 201906, 201907, 201908, 201909,
5              201910, 201911, 201912]
```

Realizando as requisições via Requests

```
1 #interando por capital e ano. Realizei a contagem para listas month_2018_2019
2 index = ['codigoIBGE', 'nomeIBGEsemAcento', 'pais', 'uf', 'id', 'descricao', 'descricaoDetalhada']
3 dataFramelist = []
4 for i in range(27):
5     cod = capitaiscodeIBGE[capitais[i]]
6     for j in range(12):
7         mon = month_2019[j]
8         url = f'http://www.transparencia.gov.br/api-de-dados/\
9         bolsa-familia-por-municipio?mesAno={mon}&codigoIbge={cod}&pagina=1'
10        time.sleep(5)
11        try:
12            df = pd.DataFrame(requests.get(url).json()[0])
13            df.drop(index=index, inplace=True)
14            df.reset_index(inplace=True)
15            dataFramelist.append(df)
16            print(f'Capital: {capitais[i]} / Mês: {mon} / Restante: {27-i}')
17        except:
18            print(f'Mês {mon} não disponível')
19            continue
```

Foram 648 requests no portal transparência.

Manipulando os DataFrames

```
1 bf2019 = pd.concat(dataFramelist,axis=0)
```

```
1 bf2018 = pd.concat(dataFramelist,axis=0)
```

```
1 bf = pd.merge(bf2018,bf2019, how='outer')
```

```
1 # Dropando colunas que não me interessam.
```

```
2 bf.drop(columns=['Unnamed: 0','index','id','tipo'], inplace=True)
```

```
1 #mudando a ordem das colunas
```

```
2 bf = bf[['municipio', 'dataReferencia', 'valor', 'quantidadeBeneficiados']]
```

Editando o DataFrame Final

```
1 # calculando um valor de benefício médio.
2 bf['Valor por Benef'] = bf['valor']/bf['quantidadeBeneficiados']
```

```
1 # Coletando a população de cada capital.
2 pop = {'São Paulo': 12252023, 'Rio De Janeiro': 6718903, 'Brasília': 3015268, 'Salvador': 2872347,
3        'Fortaleza': 2669342, 'Belo Horizonte': 2512070, 'Manaus': 2182763, 'Curitiba': 1933105,
4        'Recife': 1645727, 'Goiânia': 1516113, 'Belém': 1492745, 'Porto Alegre': 1483771,
5        'São Luís': 1101884, 'Maceió': 1018948, 'Campo Grande': 895982, 'Natal': 884122,
6        'Teresina': 864845, 'João Pessoa': 809015, 'Aracaju': 657013, 'Cuiabá': 612547,
7        'Porto Velho': 529544, 'Macapá': 503327, 'Florianópolis': 500973, 'Rio Branco': 407319,
8        'Boa Vista': 399213, 'Vitória': 362097, 'Palmas': 299127}
```

```
1 # Usando os dados acima para adicionar no Data Frame.
2 bf['Pop'] = bf['municipio'].str.title().apply(lambda x: pop[x] )
```

```
1 bf['Valor per Habit'] = bf['valor']/bf['Pop']
```

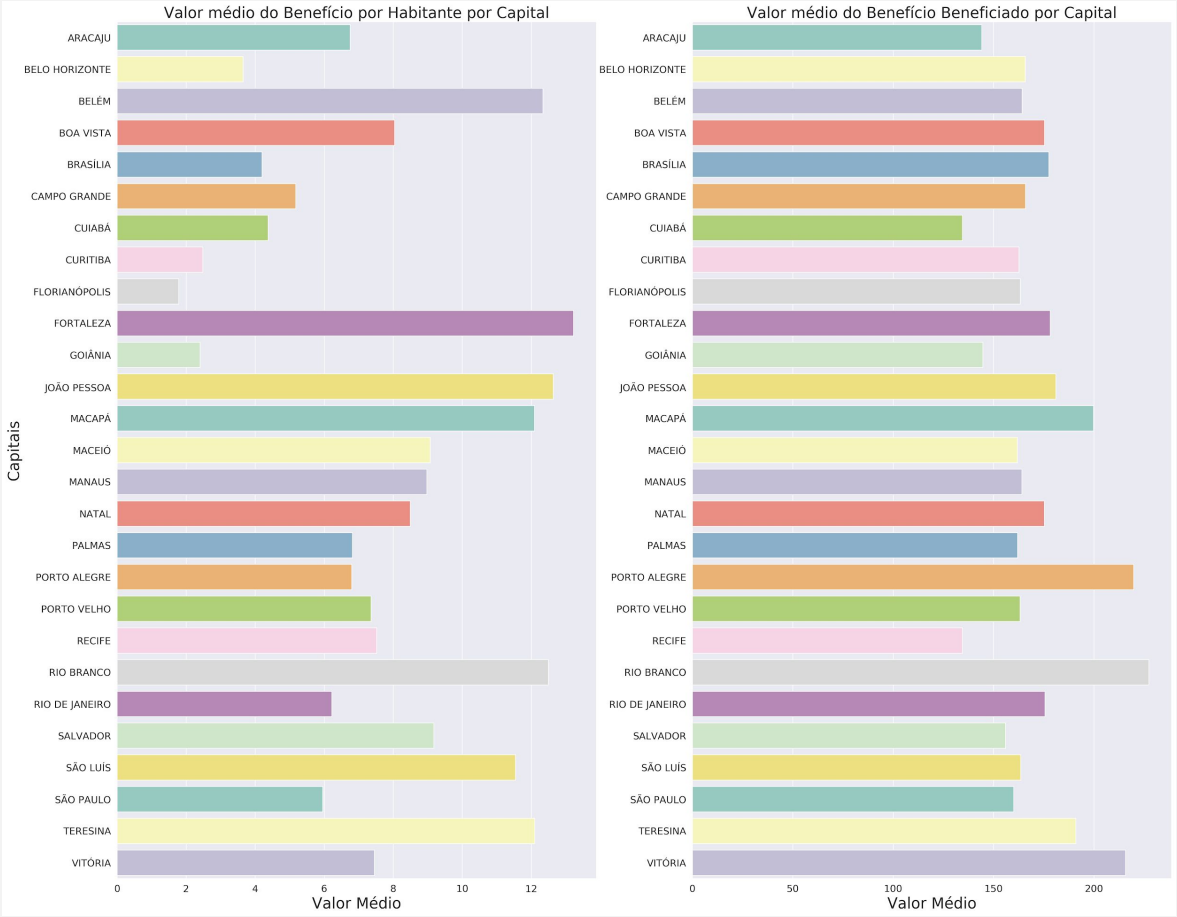
```
1 #Coletando a região de cada capital.
2 region = {'São Paulo': 'Sudeste', 'Rio De Janeiro': 'Sudeste', 'Brasília': 'Centro Oeste', 'Salvador': 'Nordeste',
3           'Fortaleza': 'Nordeste', 'Belo Horizonte': 'Sudeste', 'Manaus': 'Norte', 'Curitiba': 'Sul',
4           'Recife': 'Nordeste', 'Goiânia': 'Centro Oeste', 'Belém': 'Norte', 'Porto Alegre': 'Sul',
5           'São Luís': 'Nordeste', 'Maceió': 'Nordeste', 'Campo Grande': 'Centro Oeste', 'Natal': 'Nordeste',
6           'Teresina': 'Nordeste', 'João Pessoa': 'Nordeste', 'Aracaju': 'Nordeste', 'Cuiabá': 'Centro Oeste',
7           'Porto Velho': 'Norte', 'Macapá': 'Norte', 'Florianópolis': 'Sul', 'Rio Branco': 'Norte',
8           'Boa Vista': 'Norte', 'Vitória': 'Sudeste', 'Palmas': 'Norte'}
```

```
1 # Usando os dados acima para construir a coluna região.
2 bf['Regiao'] = bf['municipio'].str.title().apply(lambda x: region[x] )
```

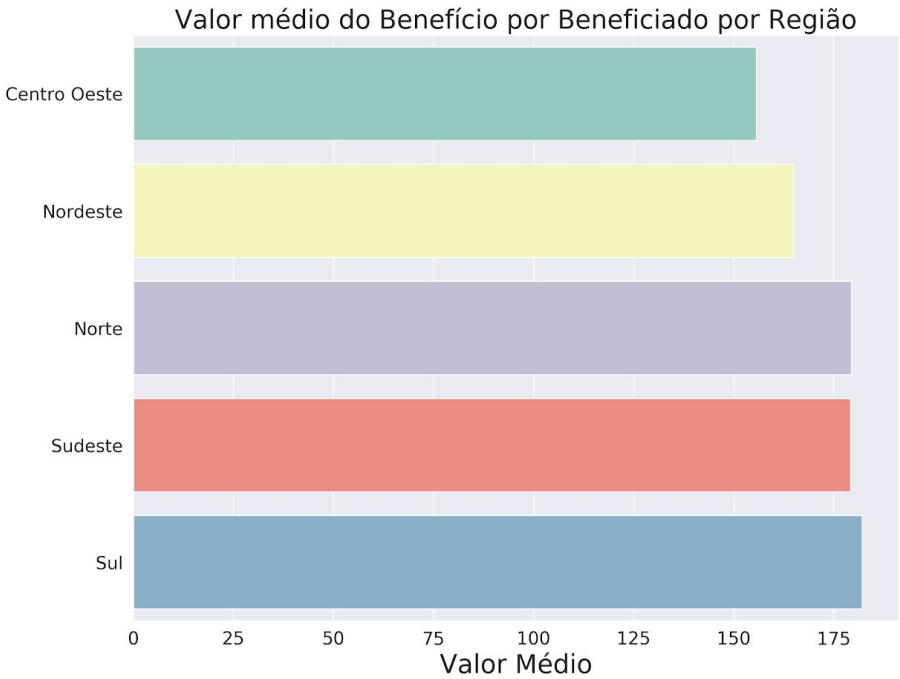
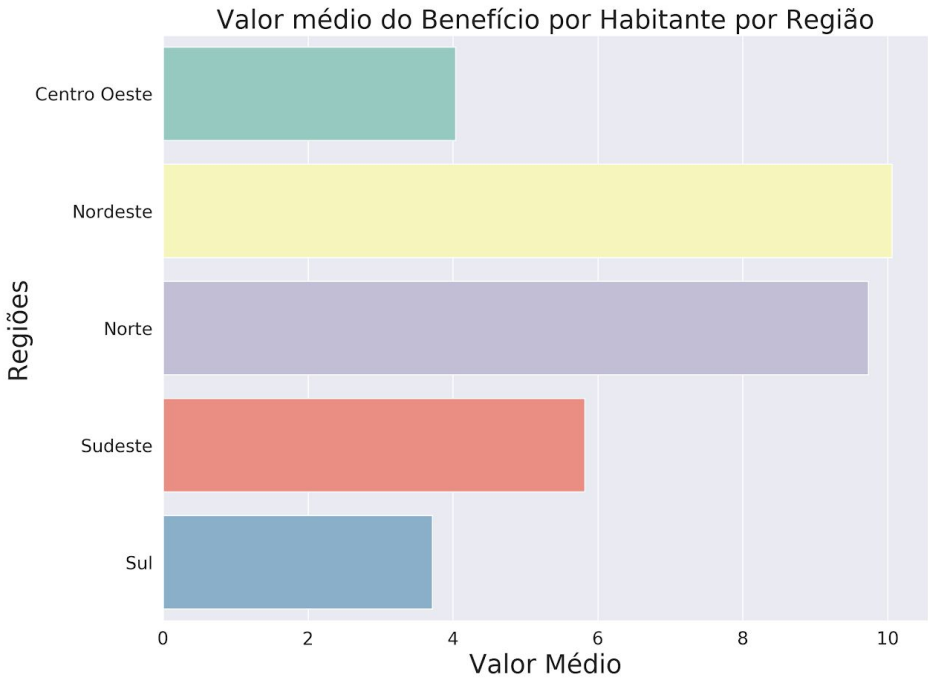

DataFrame Final

	municipio	dataReferencia	valor	quantidadeBeneficiados	Valor por Benef	Pop	Valor per Habit	Regiao	% pop.benef
0	RIO BRANCO	01/01/2018	4877303.0	23418	208.271543	407319	11.974160	Norte	5.749302
1	RIO BRANCO	01/02/2018	4907869.0	23588	208.066347	407319	12.049202	Norte	5.791038
2	RIO BRANCO	01/03/2018	4922561.0	23747	207.291911	407319	12.085272	Norte	5.830074
3	RIO BRANCO	01/04/2018	4717363.0	22638	208.382498	407319	11.581495	Norte	5.557806
4	RIO BRANCO	01/05/2018	4781556.0	22921	208.610270	407319	11.739094	Norte	5.627285
...
615	PALMAS	01/07/2019	2134158.0	12333	173.044515	299127	7.134622	Norte	4.122998
616	PALMAS	01/08/2019	2135159.0	12281	173.858725	299127	7.137968	Norte	4.105614
617	PALMAS	01/09/2019	2078646.0	11812	175.977481	299127	6.949042	Norte	3.948824
618	PALMAS	01/10/2019	2073937.0	11752	176.475238	299127	6.933299	Norte	3.928766
619	PALMAS	01/11/2019	2037836.0	11462	177.790612	299127	6.812611	Norte	3.831817

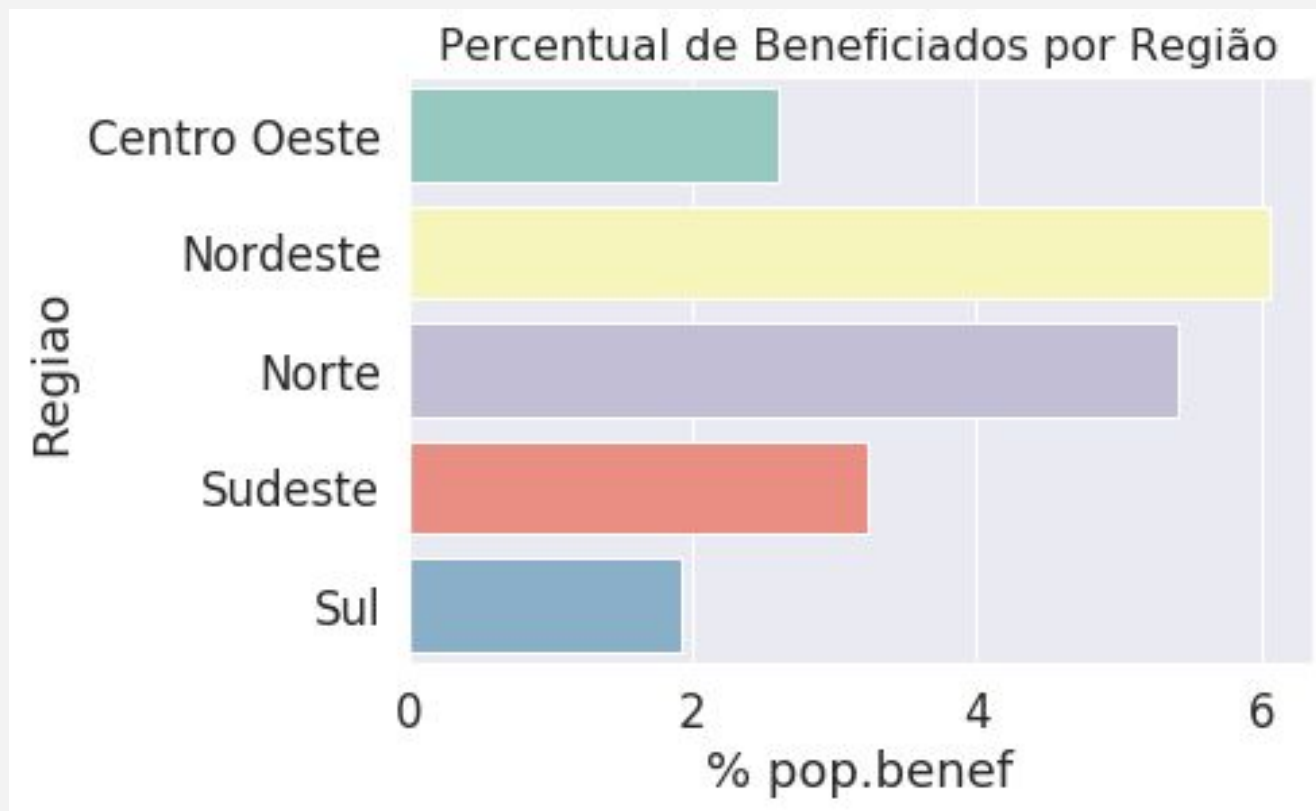
Análises e Conclusões



Análises e Conclusões



Análises e Conclusões



Análises e Conclusões

- Custo do Bolsa Família Nas capitais girou em torno de 4 Bilhões anuais. O projeto prevê o gasto total de de 29 Bilhões de reais;
- STF aprovou em 2018 um orçamento de 708 milhões.

Portal FBI.gov (Web Scraping)

MOST WANTED

Ten Most Wanted | Fugitives | **Terrorism** | Kidnappings/Missing Persons | Seeking Info | Parental Kidnappings | Bank Robbers | ECAP | VICAP

Most Wanted Terrorists | Seeking Information - Terrorism | Domestic Terrorism

Terrorism

Filter by Categories

 and Search for


Search for...

Filter by Year


Filter

Sort by: Newest


Results: 47 Items




Seeking Information -



Seeking Information -



Seeking Information -



Most Wanted Terrorists

Coletando os Dados

```
1 #URL escolhida.  
2 url= 'https://www.fbi.gov/wanted/terrorism'
```

```
1 #Fazendo a requisição.  
2 soup = BeautifulSoup(requests.get(url).content)
```

```
1 #Query classe terroristas.  
2 query = soup.body.find_all('li',{'class':'portal-type-person castle-grid-block-item'})
```

```
1 #Criando os links para procura.  
2 links = [item.p.a['href'] for item in query]
```

```
1 classes = ['wanted-person-remarks', 'wanted-person-details']  
2 d = {}  
3 for i in range(len(links)):  
4     e = {}  
5     s2p = BeautifulSoup(requests.get(links[i]).content)  
6     e['Name'] = str(s2p.h1.text.title())  
7     get = [item.text for item in s2p.body.find_all('td')]  
8     for k in range(0, len(get), 2):  
9         try:  
10             e[get[k]] = get[k+1]  
11         except:  
12             e[get[k]] = np.nan  
13     for j in range(2):  
14         s = str(s2p.body.find_all('div',{'class':classes[j]}))  
15         try:  
16             e[lst[j]] = re.findall(r'.*',s)[4]  
17         except:  
18             e[lst[j]] = np.nan  
19     else:  
20         d[i] = e
```

Dataset Pronto

Name	Date(s) of Birth Used	Place of Birth	Height	Build	Complexion	Sex	wanted-person-remarks	wanted-person-details	Hair	Eyes	Weight	Citizenship	Languages	Scars and Marks	Race	Occupation	Nationality	NCIC
Shaykh Aminullah	1961, 1967, 1973	Konar Province, Afghanistan	5'10"	Thin, with a large, round stomach	Light	Male	Aminullah wears thick glasses and a curly, che...	Shaykh Aminullah is wanted for questioning in ...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Faker Ben Abdelazziz Boussora	March 22, 1964	Tunisia	5'7"	NaN	Olive	Male	Boussora has predominately protruding ears and...	Faker Ben Abdelazziz Boussora is wanted for qu...	Black	Dark	160 to 170 pounds	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Abdullah Al-Rimi	1974	Ta'iz, Yemen	Unknown	Unknown	Olive	Male	Al-Rimi may be residing in Yemen.	Abdullah Al-Rimi is wanted for questioning in ...	Black	Black	Unknown	Yemeni	Arabic	None known	NaN	NaN	NaN	NaN
Ibrahim Salih Mohammed	October 16, 1966	Tarut, Saudi Arabia	5'4"	Unknown	Olive	Male	<p>Al-Yacoub is an alleged member of ...	Not informed	Black	Brown	150 pounds	Saudi Arabian	Arabic	None known	NaN	NaN	NaN	NaN

Coletando Coordenadas dos locais de nascimento

```
1 def coordinates(x):
2     time.sleep(5)
3     try:
4         x = str(x).replace(' ', '+')
5         print(x)
6         url = f'https://nominatim.openstreetmap.org/search?q={x}&format=geojson'
7         d = requests.get(url).json()
8         try:
9             return (d['features'][0]['geometry']['coordinates'][1], d['features'][0]['geometry']['coordinates']
10         except:
11             print(f'Localização <{x}> não obtida!')
12             return np.nan
13     except:
14         pass
```

```
1 terrorists['Coordinates'] = terrorists['Place of Birth'].apply(lambda x : coordinates(x))
```

```
Konar+Province,+Afghanistan
Tunisia
Ta'iz,+Yemen
Tarut,+Saudi+Arabia
Al+Ihsa,+Saudi+Arabia
Localização <Al+Ihsa,+Saudi+Arabia> não obtida!
Lebanon
Bloomington,+Indiana
```

Plotando o Mapa Com Gmaps

```
1 #https://console.developers.google.com/  
2 gmaps.configure(api_key='AIzaSyC8dPoiQ0l3VzZguoYgyQxR[REDACTED]')
```

```
1 marker_layer = gmaps.marker_layer(c, info_box_content=n)  
2 fig = gmaps.figure()  
3 fig.add_layer(marker_layer)  
4 fig
```



Página de Carreiras do Nubank

```
1 # Procurando jobs at Nubank
```

```
1 jobs_name = [item.a.text for item in nu.body.find_all('div',{'class':'opening'})]
2 jobs_location = [item.span.text for item in nu.body.find_all('div',{'class':'opening'})]
```

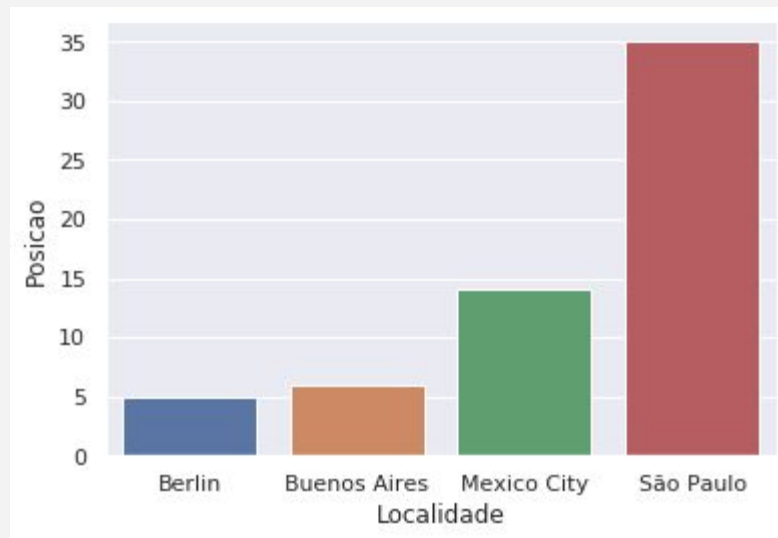
```
1 n = len(jobs_name)
2 d = {}
3 for i in range(n):
4     e = {}
5     e['Posicao'] = jobs_name[i]
6     e['Localidade'] = jobs_location[i]
7     d[i] = e
```

```
1 jobsAtnu = pd.DataFrame(d).T
```

```
1 jobsAtnu['Posicao'].apply(lambda x : re.findall(r'.*[dD]ata.*',x)).sum()
```

```
['Data Scientist',
 'Data Scientist (Econometrics/Statistics)',
 'Senior Data Scientist',
 'Senior Data Scientist',
 'Data Engineer',
 'Sr. Product Manager, Technical (Platforms - Data)']
```

Página de Carreiras do Nubank



FELIZ NATAL!

