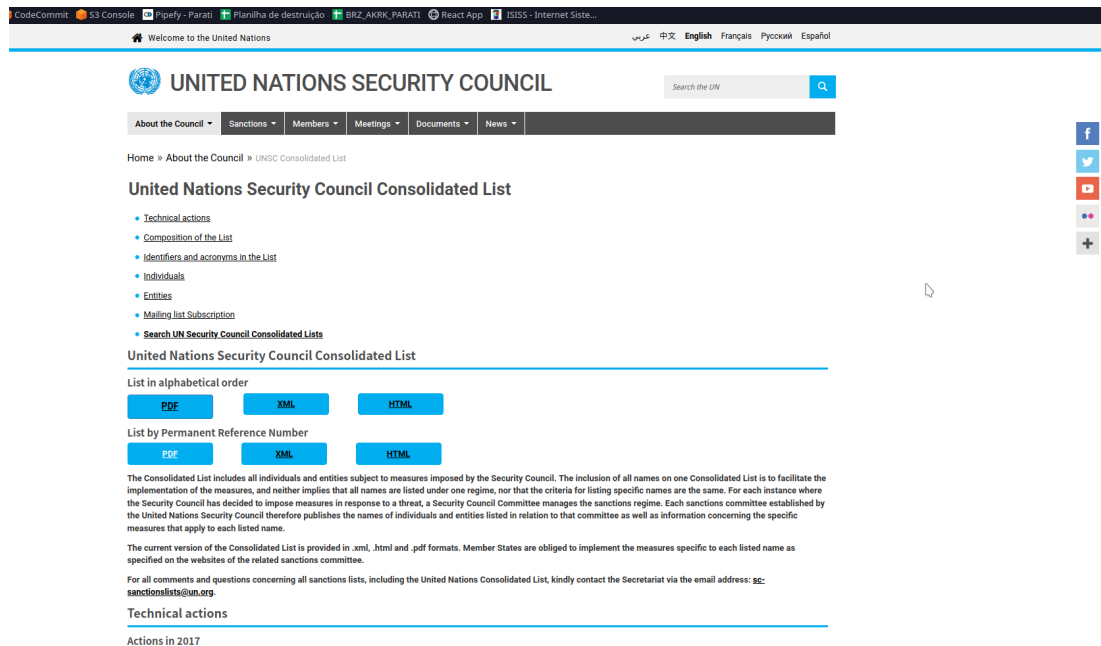


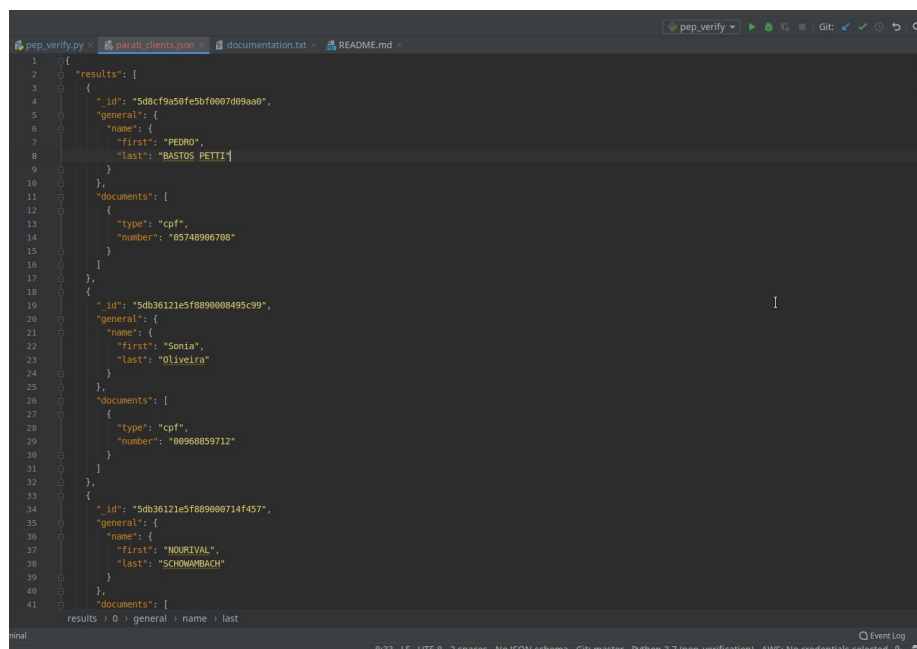
Análise de pessoas sujeitas a medidas impostas pelo Conselho de Segurança das Nações Unidas na base de dados da Parati, fevereiro 2020

Acessa-se a página do Conselho de Segurança da ONU “United Nations Security Council Consolidated List” (<https://www.un.org/securitycouncil/content/un-sc-consolidated-list>).



Download arquivo .XML mais recente disponibilizado contendo a lista de pessoas e entidades sujeitas a medidas impostas pelo Conselho de Segurança das Nações Unidas.

Cria-se arquivo .JSON com a lista de clientes da Parati a partir do banco de dados desta empresa.



Utiliza-se o script `fop_verify.py` escrito na linguagem de programação Python para carregar os dados do XML como uma planilha contendo os nomes dos indivíduos sujeitos a medidas impostas pelo Conselho de Segurança das Nações Unidas e transformar os dados dos clientes da Parati em uma planilha similar.

```

1  import pandas as pd, unicodedata
2  from locale import setlocale, LC_ALL
3  import xmltodict
4
5  setlocale(LC_ALL, 'pt_BR.UTF-8')
6
7  with open("consolidated.xml") as UN_list:
8      xml = UN_list.read()
9      fop_dict = xmltodict.parse(xml)
10
11  fop_list = pd.DataFrame.from_dict(fop_dict['CONSOLIDATED_LIST']['INDIVIDUALS']['INDIVIDUAL'], orient='columns', dtype=str)[['DATAID', 'FIRST_NAME', 'SECOND_NAME', 'THIRD_NAME', 'FOURTH_NAME']]
12  fop_list.to_csv("fop_list")
13
14  for name in ['FIRST_NAME', 'SECOND_NAME', 'THIRD_NAME', 'FOURTH_NAME']:
15      fop_list[name] = fop_list[name].apply(lambda value: str(unicodedata.normalize('NFKD', str(value)).encode('ASCII', 'ignore').decode('UTF-8')).upper())
16  fop_list['Name'] = fop_list['FIRST_NAME'] + " " + fop_list['SECOND_NAME'] + " " + fop_list['THIRD_NAME'] + " " + fop_list['FOURTH_NAME']
17  fop_list['Name'] = fop_list['Name'].apply(lambda value: value.replace(" NAN", "").replace(" NA", "").replace(" NONE", ""))
18  fop_list = fop_list[['DATAID', 'Name']]
19  print(fop_list)
20
21  client_list = pd.read_json("parati_clients.json", orient='records')
22
23  x = [{"i": len(client_list['results'])}]
24  for i in range(len(x)):
25      if 'documents' in client_list['results'][i]:
26          x[i] = (client_list['results'][i]['general']['name']['first']) + " " + client_list['results'][i]['general']['name']['last']
27  client_list = pd.DataFrame({'Name': x})
28  client_list['Name'] = client_list['Name'].apply(lambda value: str(unicodedata.normalize('NFKD', value).encode('ASCII', 'ignore').decode('UTF-8')).upper())
29
30  comparison = client_list[client_list['Name'].isin(fop_list['Name'])]
31  print(comparison)
32
33  comparison.to_csv("Analise_FOP_base_clientes Parati.csv", encoding='utf-8', sep=";", index=False)

```

O script procura por nomes que estejam em ambas as planilhas e os separa em uma nova planilha.

```

1  import pandas as pd, unicodedata
2  from locale import setlocale, LC_ALL
3  import xmltodict
4
5  setlocale(LC_ALL, 'pt_BR.UTF-8')
6
7  with open("consolidated.xml") as UN_list:
8      xml = UN_list.read()
9      fop_dict = xmltodict.parse(xml)
10
11  fop_list = pd.DataFrame.from_dict(fop_dict['CONSOLIDATED_LIST']['INDIVIDUALS']['INDIVIDUAL'], orient='columns', dtype=str)[['DATAID', 'FIRST_NAME', 'SECOND_NAME', 'THIRD_NAME', 'FOURTH_NAME']]
12  fop_list.to_csv("fop_list")
13
14  for name in ['FIRST_NAME', 'SECOND_NAME', 'THIRD_NAME', 'FOURTH_NAME']:
15      fop_list[name] = fop_list[name].apply(lambda value: str(unicodedata.normalize('NFKD', str(value)).encode('ASCII', 'ignore').decode('UTF-8')).upper())
16  fop_list['Name'] = fop_list['FIRST_NAME'] + " " + fop_list['SECOND_NAME'] + " " + fop_list['THIRD_NAME'] + " " + fop_list['FOURTH_NAME']
17  fop_list['Name'] = fop_list['Name'].apply(lambda value: value.replace(" NAN", "").replace(" NA", "").replace(" NONE", ""))
18  fop_list = fop_list[['DATAID', 'Name']]
19  print(fop_list)
20
21  client_list = pd.read_json("parati_clients.json", orient='records')
22
23  x = [{"i": len(client_list['results'])}]
24  for i in range(len(x)):
25      if 'documents' in client_list['results'][i]:
26          x[i] = (client_list['results'][i]['general']['name']['first']) + " " + client_list['results'][i]['general']['name']['last']
27  client_list = pd.DataFrame({'Name': x})
28  client_list['Name'] = client_list['Name'].apply(lambda value: str(unicodedata.normalize('NFKD', value).encode('ASCII', 'ignore').decode('UTF-8')).upper())
29
30  comparison = client_list[client_list['Name'].isin(fop_list['Name'])]
31  print(comparison)
32
33  comparison.to_csv("Analise_FOP_base_clientes Parati.csv", encoding='utf-8', sep=";", index=False)

```

Run: fop_verify.py

```

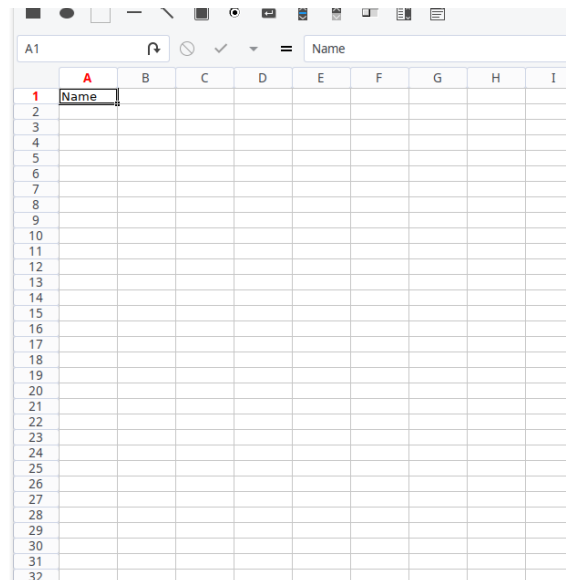
704 6908056 MUHAMMAD HANZA ZUBAIDI
705 6908704 AHMAD DUMAR IMHAMAD AL-FITOURI
706 6908707 ABD AL-RAHMAN AL-NILAD
707 2975510 FAIZULLAH KHAN HODRZAI
708 2959427 SAID JAN ABD AL-SALAM

```

[789 rows x 2 columns]
Empty DataFrame
Columns: [Name]
Index: []

Process finished with exit code 0

O resultado da verificação é exportado como uma planilha contendo os nomes de todos os indivíduos na lista consolidada das Nações Unidas encontrados no banco de dados da Parati em formato .CSV



	A	B	C	D	E	F	G	H	I
1	Name								
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									

A verificação é feita sob demanda, com base na listagem mais recente disponibilizada pela ONU, para cada nova proposta de cadastro de cliente no banco de dados. E para cada nova versão da lista consolidada o processo valida toda a lista de clientes contra a lista mais recente das Nações Unidas, dessa forma caso um cliente já cadastrado venha a sofrer as medidas impostas pelo Conselho de Segurança das Nações Unidas o programa o listará.