



Projeto: Relacionando Covid e clima

Isabela Caroline de Sousa 218071
Lucas Antevere Santana 201775
Matheus Bulhões Barbosa 222157



Tema

- Covid - 19
- Condições climáticas x contaminação
- Analisar a sazonalidade da Doença



Análise

Relacionar dados locais como:

- Temperatura
- Umidade
- Pressão
- Contaminação diária
- Número de Óbitos



Metodologia

- Seleção de países c/ diferentes condições climáticas
- Escolha do período de tempo para análise
- Casos x Condições Climáticas no Tempo
- Minimizar influências externas (comportamento social e ações governamentais)
- Busca por padrões na evolução da doença

Modelo Conceitual



Nota: o objetivo do projeto é verificar se a relação "influencia" de fato existe.



Seleção de países

- Para a realização dessa etapa utilizamos o modelo de grafos
- Nosso objetivo: agrupar os países por semelhanças climáticas e, assim, selecionar quais os países cujos dados relacionados à covid-19 seriam usados no decorrer do nosso projeto
- A característica visual que os grafos possuem facilitou a escolha dos países depois de agrupados



Seleção dos países

Nessa etapa, inicialmente, fizemos um programa em python para unir duas APIs, uma vez que não conseguimos encontrar uma API que contivesse todos os dados necessários. Com a tabela que criamos, usando o NEO4J, verificamos os países com características climáticas próximas (temperatura, pressão e umidade) e tentamos separá-los em grupos de acordo com similaridade.

Open Weather Map API

JSON

Example of API response

```
{
  "message": "Count: 24",
  "cod": "200",
  "city_id": 4290968,
  "calctime": 0.00297316,
  "cnt": 24,
  "list": [
    {
      "dt": 1578384000,
      "main": {
        "temp": 275.45,
        "feels_like": 271.7,
        "pressure": 1034,
        "humidity": 74,
        "temp_min": 274.26,
        "temp_max": 276.48
      },
      "wind": {
        "speed": 2.16,
        "deg": 87
      },
      "clouds": {
        "all": 99
      },
      "weather": [
        {
          "id": 501,
          "main": "Rain",
          "description": "moderate rain",
          "icon": "10n"
        }
      ],
      "rain": {
        "in": 0.9
      }
    },
    ...
  ]
}
```

Código em Python

```
import io
import requests
import json
url = "https://api.ipgeolocationapi.com/countries"
data = requests.get(url)
js = data.json()
countries = []

for i in js:
    country = [i, str(js[i]['geo']['latitude'])[0:6], str(js[i]['geo']['longitude'])[0:6]]
    while (len(country[1]) < 6):
        country[1] = country[1]+'0'
    while (len(country[2]) < 6):
        country[2] = country[2]+'0'
    countries.append(country)

url2 = "http://api.openweathermap.org/data/2.5/weather?lat=000000&lon=111111&appid=9de243494c0b295cca9337e1e96b00e2"
for i in countries:
    old = url2[51:57]
    url2 = url2.replace(old, i[1], 1)
    old = url2[62:68]
    url2 = url2.replace(old, i[2], 1)
    data = requests.get(url2)
    js = data.json()
    print(i[0]+' ', '+str(js['main']['temp'])+', '+str(js['main']['pressure'])+', '+str(js['main']['humidity']))
```

Seleção dos países - Análise no Neo4j

1. Importamos a tabela criada anteriormente e criamos os nós
2. Conectamos os países com características similares em termos de temperatura, pressão e umidade
3. Consideramos que os países que possuem os 3 tipos de relações de similaridade possuem uma similaridade climática
4. Deletamos as outras relações de similaridade para que os grafos não fiquem poluídos visualmente
5. Determinamos os países com os quais um determinado país é semelhante
6. Visualizamos graficamente os grupos de países similares entre si



Queries no Neo4j

- Para temperatura, pressão e umidade, conectamos os países que têm características próximas

```
MATCH (c1:Country)
MATCH (c2:Country)
WHERE ABS(toFloat(c1.temp) - toFloat(c2.temp)) < 4 AND c1 <> c2
CREATE (c1)-[:similar_temperature]->(c2)
```

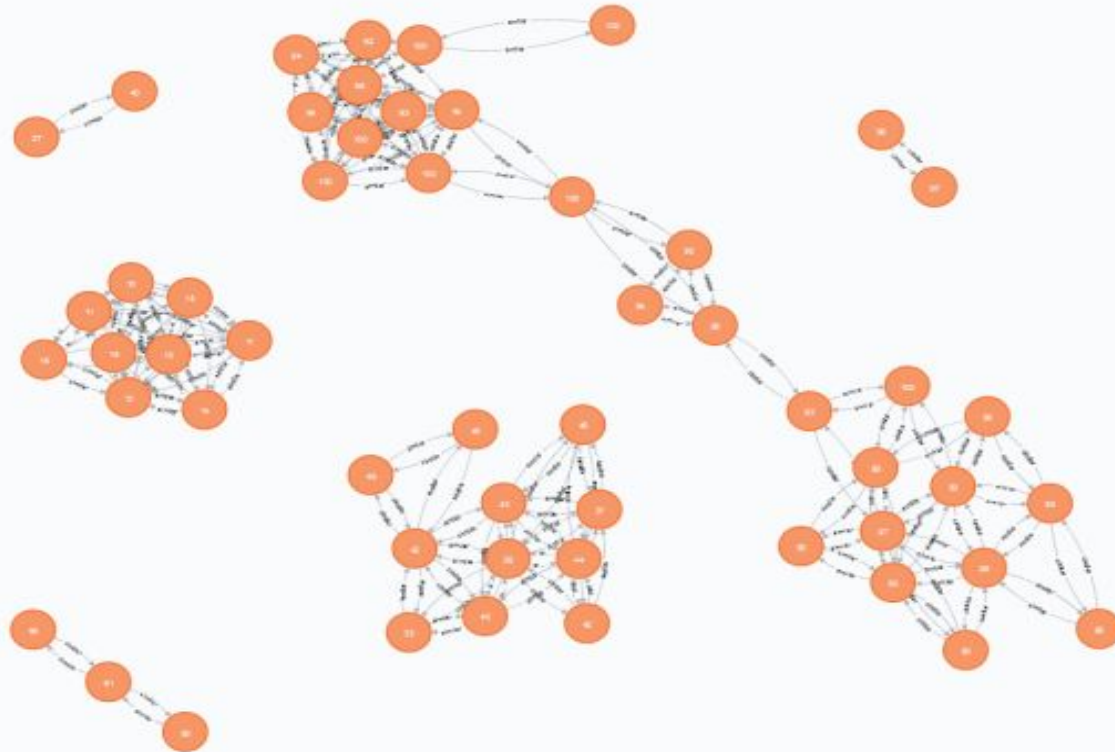
```
MATCH (c1:Country)
MATCH (c2:Country)
WHERE ABS(toFloat(c1.pressure) - toFloat(c2.pressure)) < 6 AND c1 <> c2
CREATE (c1)-[:similar_pressure]->(c2)
```

```
MATCH (c1:Country)
MATCH (c2:Country)
WHERE ABS(toFloat(c1.humidity) - toFloat(c2.humidity)) < 10 AND c1 <> c2
CREATE (c1)-[:similar_humidity]->(c2)
```

- Consideramos que os países que têm os 3 tipos de similaridades possuem uma similaridade climática, no geral

```
MATCH p=(c1)-[:similar_humidity]->(c2)<-[:similar_pressure]-(c1)-[:similar_temperature]->(c2)
CREATE (c1)-[:similar]->(c2)
```

Grafo contendo grupos de Países





Seleção de países - Resultado

Para os cinco grupos expressivos que foram encontrados, como mostrado anteriormente, escolhemos os seguintes países:

- Malásia, Brunei
- Holanda, Coreia do Sul
- Egito, Emirados Árabes
- Uruguai, Brasil, Costa do Marfim
- Senegal, Sudão

Escolha do período de tempo para análise

Para abranger um período de tempo suficiente para a realização da análise, escolhemos os cinco períodos listados a seguir:

- 10/03 à 20/03
- 10/05 à 20/05
- 10/07 à 20/07
- 10/09 à 20/09
- 10/11 à 20/11



Casos X Condições climáticas

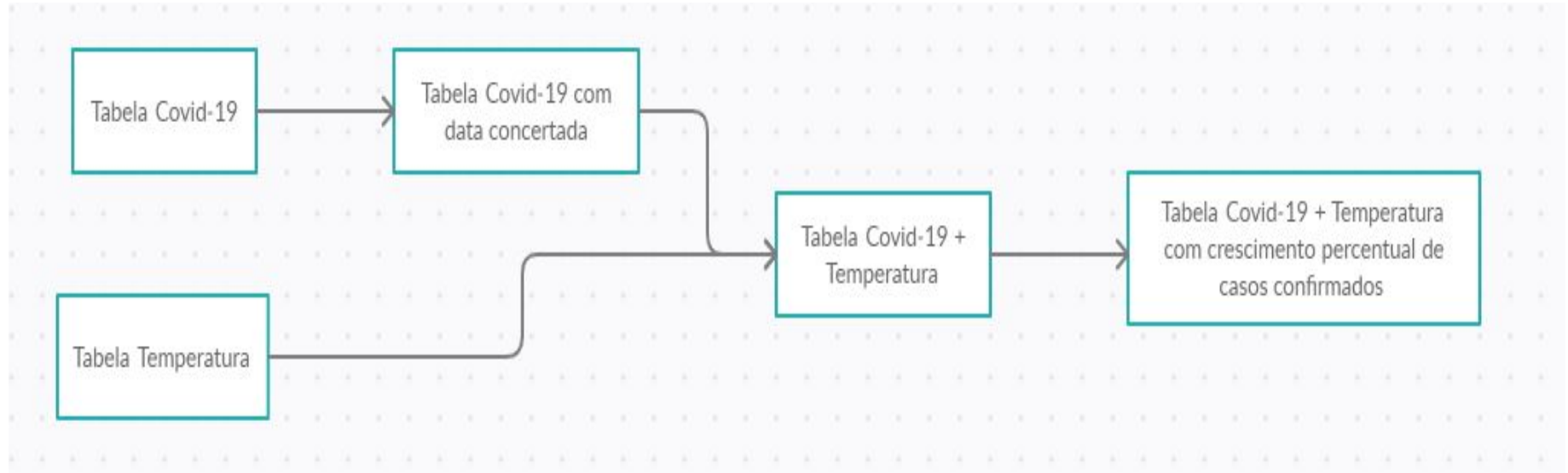
- Para realização dessa etapa utilizamos o modelo relacional e a linguagem SQL
- Inicialmente realizamos uma análise preliminar correspondente a etapa 3 do projeto
- Em seguida estendemos essa análise para abranger todos os países que fazem parte do nosso grupo de interesse

Análise preliminar



1. Importamos as tabelas Covid-19 e Temperatura
2. Seleccionamos um intervalo de datas a partir da tabela Covid-19
3. Para que houvesse compatibilidade entre os formatos de datas das duas tabelas, alteramos a coluna “Data” da tabela Covid-19 para um formato padrão de data
4. Fizemos a junção das duas tabelas a partir da data
5. Calculamos o crescimento percentual de casos confirmados

Representação da Análise



Crescimento Percentual



```
DROP VIEW IF EXISTS Covid_Percent;

CREATE VIEW Covid_Percent AS
SELECT
  C.Confirmed as Cases,
  COALESCE((C.Confirmed - CL.Confirmed)/ NULLIF(CAST(CL.Confirmed AS FLOAT), 0), 0) + 0.0 AS CasesVariation,
  C.Deaths,
  COALESCE((C.Deaths - CL.Deaths)/ NULLIF(CAST(CL.Deaths AS FLOAT), 0), 0) + 0.0 AS DeathsVariation,
  C.Recovered,
  T.Maximo,
  T.Minimo,
  T.Media,
  T.Data
FROM Covid_Data AS C, Covid_Data AS CL, Temperatura AS T
WHERE CL.Data = DATEADD(day, -1, C.Data) AND C.Data = T.Data
ORDER BY T.Data;

SELECT * FROM Covid_Percent
```



Análise complementar

Para essa análise repetimos o que foi implementado na análise preliminar, porém com um volume maior de dados, considerando que a análise foi estendida para 11 países e 5 intervalos de tempo para cada um deles.

COVID - 19 API

```
1 {  
2   {  
3     "Country": "Brazil",  
4     "CountryCode": "BR",  
5     "Province": "",  
6     "City": "",  
7     "CityCode": "",  
8     "Lat": "-14.24",  
9     "Lon": "-51.93",  
10    "Confirmed": 0,  
11    "Deaths": 0,  
12    "Recovered": 0,  
13    "Active": 0,  
14    "Date": "2020-01-22T00:00:00Z"  
15  },  
16  {  
17    "Country": "Brazil",  
18    "CountryCode": "BR",  
19    "Province": "",  
20    "City": "",  
21    "CityCode": "",  
22    "Lat": "-14.24",  
23    "Lon": "-51.93",  
24    "Confirmed": 0,  
25    "Deaths": 0,  
26    "Recovered": 0,  
27    "Active": 0,  
28    "Date": "2020-01-23T00:00:00Z"  
29  },  
30 }
```

Seleção dos dados para os países escolhidos

```
países = ['Malaysia', 'Brunei', 'Netherlands', 'South Korea', 'Egypt', 'United Arab Emirates', 'Uruguay', 'Brazil', 'Ivory Coast', 'Senegal', 'Sudan']
url = "https://api.ipgeolocationapi.com/countries"
data = requests.get(url)
js = data.json()
countries = []

for i in js:
    use = False
    for j in js[i]['unofficial_names']:
        if(j in países):
            use = True
    if js[i]['name'] in países:
        use = True
    if use:
        country = [i, js[i]['name'], str(js[i]['geo']['latitude'])[0:6], str(js[i]['geo']['longitude'])[0:6]]
        while (len(country[1]) < 6):
            country[1] = country[1]+'0'
        while (len(country[2]) < 6):
            country[2] = country[2]+'0'
        countries.append(country)

url2 = "https://api.covid19api.com/country/XX"
print('Codigo, Nome, Confirmados, Mortes, Recuperados, Data')
for i in countries:
    old = url2[34:37]
    url2 = url2.replace(old, '/' + i[0], 1)
    data = requests.get(url2)
    js = data.json()
    for j in js:
        if (str(j['Province']) == ''):
            print(j['CountryCode'] + ',' + j['Country'] + ',' + str(j['Confirmed']) + ',' + str(j['Deaths']) + ',' + str(j['Recovered']) + ',' + str(j['Date']))
```

Desta forma criamos as tabelas covid e weather.

Join entre Covid e Weather

```
DROP TABLE IF EXISTS Covid;

CREATE TABLE Covid(
  code VARCHAR(2),
  name VARCHAR(100),
  cases INT,
  deaths INT,
  recoveries INT,
  day DATE NOT NULL
)

AS SELECT codigo, nome, confirmados, mortes, recuperados, data
FROM CSVREAD('https://raw.githubusercontent.com/MatheusBulhoes/MC536-Trabalho/main/final/covid.csv')
```

```
SELECT * FROM Covid;
```

```
DROP TABLE IF EXISTS Weather;

CREATE TABLE Weather(
  code VARCHAR(2),
  name VARCHAR(100),
  averageTemp INT,
  averagePressure INT,
  averageHumidity INT,
  day DATETIME NOT NULL
)

AS SELECT codigo, nome, tempmedia, pressao, umidade, data
FROM CSVREAD('https://raw.githubusercontent.com/MatheusBulhoes/MC536-Trabalho/main/final/weather.csv')
```

```
SELECT * FROM Weather;
```

```
SELECT Covid.code, Covid.name, Covid.cases, Covid.deaths, Weather.averageTemp, Covid.day
FROM Covid
JOIN Weather ON Covid.day = Weather.day AND Covid.code = Weather.code;
```

Crescimento Percentual

```
DROP TABLE IF EXISTS Content;

CREATE TABLE Content(
  code VARCHAR(2),
  name VARCHAR(100),
  cases INT,
  deaths INT,
  averagetemp INT,
  day DATE NOT NULL
)
AS SELECT code, name, cases, deaths, averagetemp, day
FROM CSVREAD('https://raw.githubusercontent.com/MatheusBulhoes/MC536-Trabalho/main/final/content.csv')
```

```
DROP VIEW IF EXISTS Covid_Percent;

CREATE VIEW Covid_Percent AS
SELECT
  C.code,
  C.name,
  C.cases as Cases,
  COALESCE((C.cases - CL.cases)/ NULLIF(CAST(CL.cases AS FLOAT), 0), 0) + 0.0 AS CasesVariation,
  C.averagetemp,
  C.day
FROM Content AS C, Content AS CL
WHERE CL.day = DATEADD(day, -1, C.day) AND C.code = CL.code
ORDER BY C.code, C.day;

SELECT * FROM Covid_Percent
```


Plotando os gráficos

```
# Load content
contentURL = "https://raw.githubusercontent.com/MatheusBulhoes/MC536-Trabalho/main/final/content.csv"

df = pd.read_csv(contentURL)

#Organize content
dataPerCountry = {}
for row in df.iteruples():

    cases = row[3]
    deaths = row[4]
    temp = row[5]

    day = row[6]
    dayString = day[:4] + "-" + day[4:6] + "-" + day[6:8]

    dayObj = date.fromisoformat(dayString)

    if row[2] in dataPerCountry:
        dataPerCountry[row[2]].append((cases, deaths, temp, dayObj))
    else:
        dataPerCountry[row[2]] = [(cases, deaths, temp, dayObj)]
```

```
# curva dos casos
fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('Dias')
ax1.set_ylabel('Casos', color=color)
ax1.plot(dayX, casesY, color=color)
ax1.tick_params(axis='y', labelcolor=color)

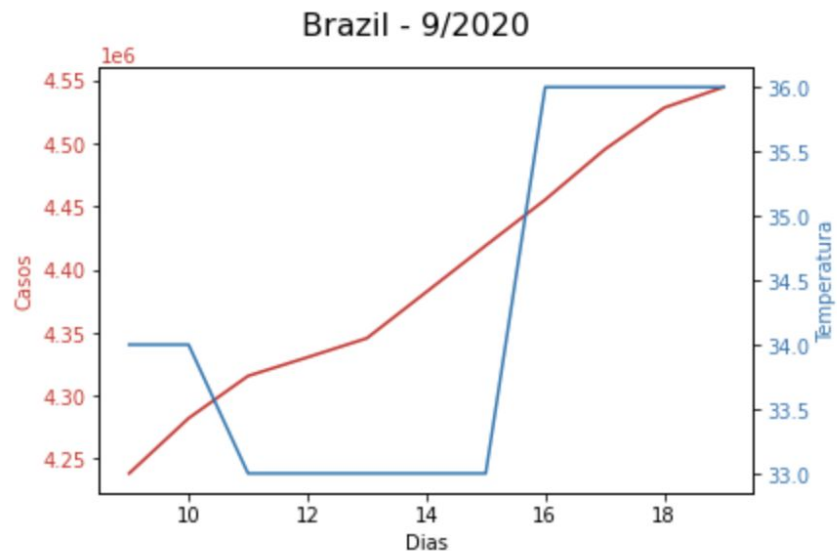
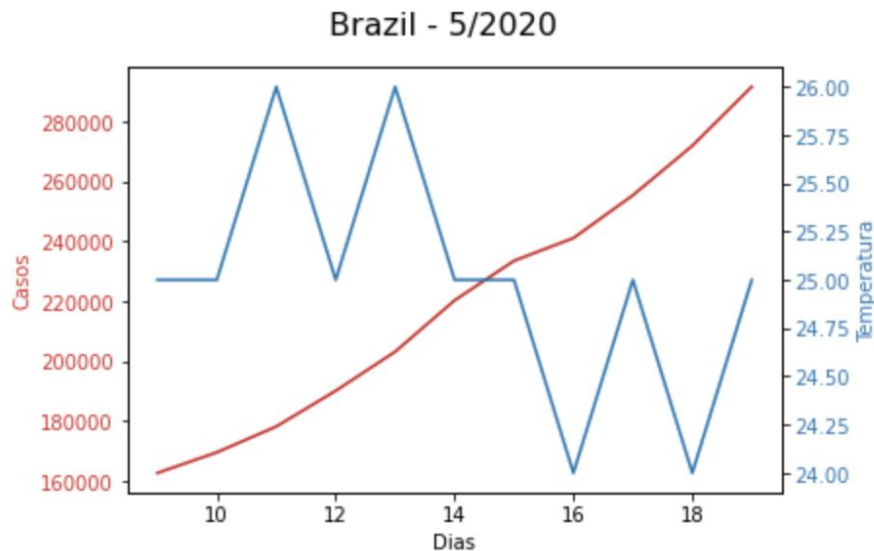
# curva da temperatura
ax2 = ax1.twinx()

color = 'tab:blue'
ax2.set_ylabel('Temperatura', color=color)
ax2.plot(dayX, tempY, color=color)
ax2.tick_params(axis='y', labelcolor=color)

fig.suptitle(countryDataKey + " - " + str(item[3].month) + "/" + str(item[3].year), fontsize=16)
plt.show()
```

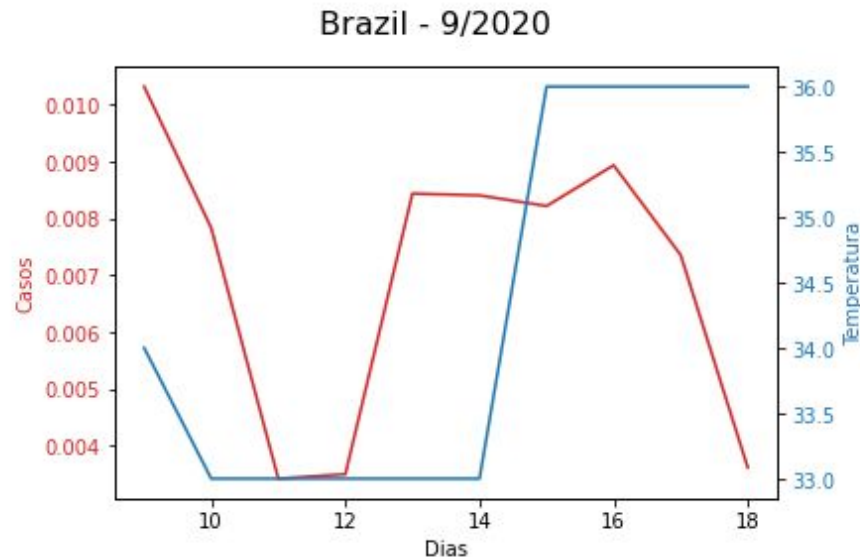
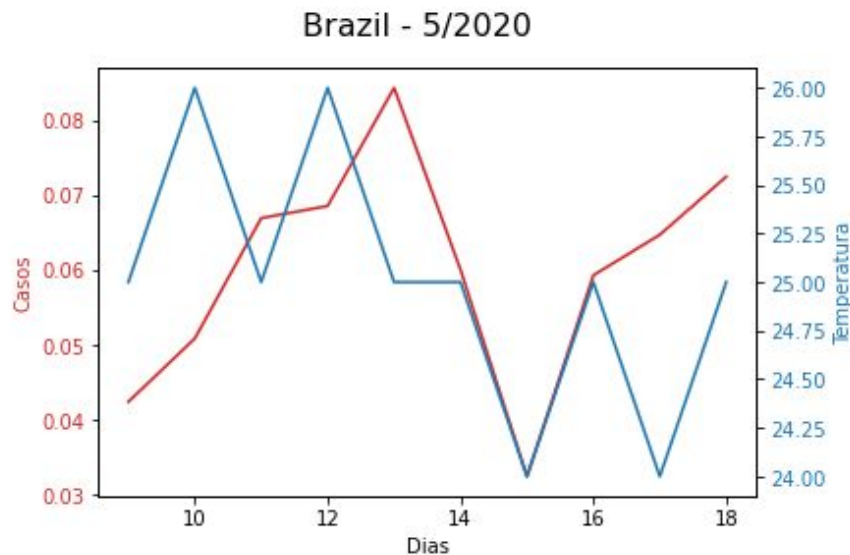
A partir de ambas as tabelas geradas nos passos anteriores, podemos gerar vários gráficos relacionando o crescimento da doença e a temperatura com as datas

Casos X Condições climáticas: Resultados



Em geral, o número de casos está em constante crescimento.

Casos X Condições climáticas: Resultados



Enquanto o número percentual de casos tende a cair ao longo do tempo.



Casos X Condições climáticas: Resultados

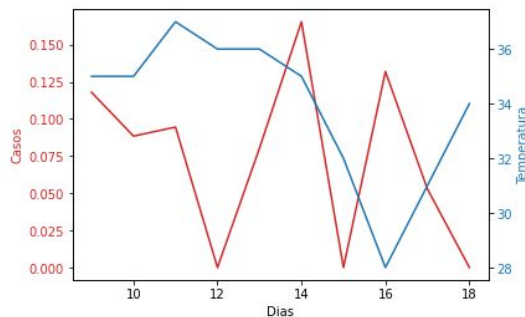
Iremos destacar alguns resultados interessantes obtidos entre os seguintes grupos já determinados:

- Malásia, Brunei
- Holanda, Coreia do Sul
- Egito, Emirados Árabes
- Uruguai, Brasil, Costa do Marfim
- Senegal, Sudão

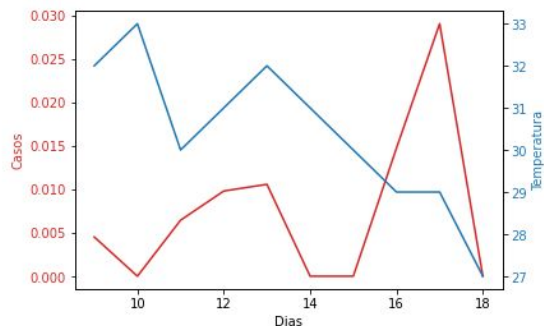
Obs: Na API que utilizamos os dados de Brunei não foram atualizados, por isso, um novo grupo foi formado, Senegal, Sudão e Malásia.

Análise de um dos grupos de países

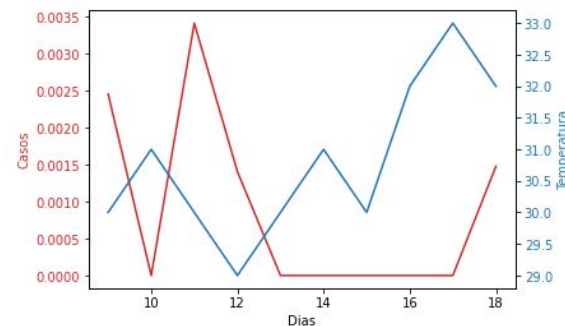
Sudan - 5/2020



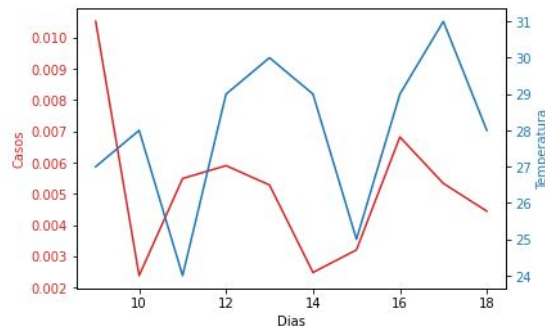
Sudan - 7/2020



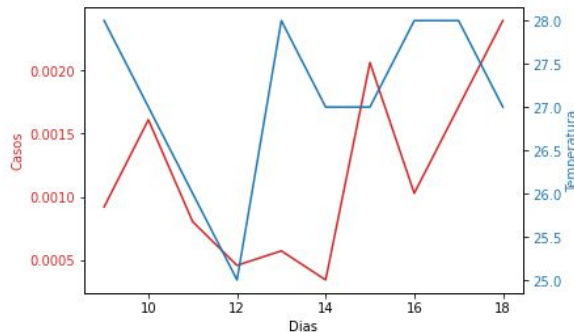
Sudan - 9/2020



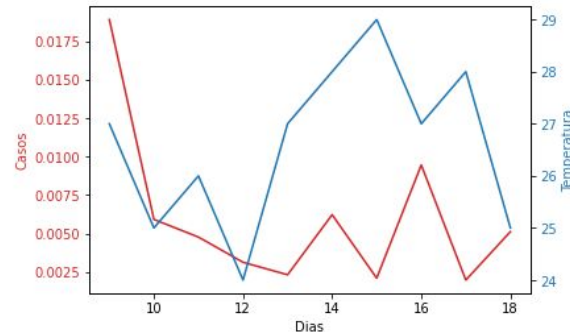
Malaysia - 5/2020



Malaysia - 7/2020



Malaysia - 9/2020



Análise geral dos outros grupos



- Holanda, Coreia do Sul:

Holanda: Decrescente, segunda onda em Setembro

Coreia: Números baixos e crescentes a partir de Maio

- Egito, Emirados Árabes

Egito: Decrescente, constante no fim do ano

Emirados: Decrescente, pico em Setembro

- Uruguai, Brasil, Costa do Marfim

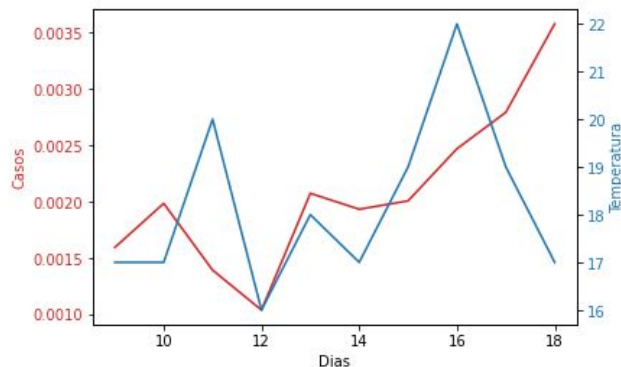
Uruguai: Constante ao longo do ano, leve crescimento em Novembro

Costa do Marfim: Decrescente ao longo do ano todo

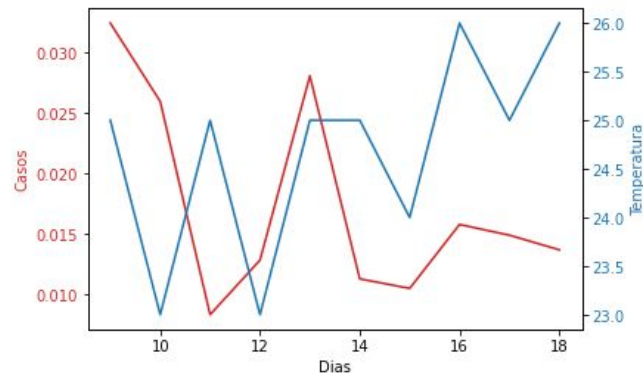
Brasil: Decrescente ao longo do ano todo (+1M/mês)

Caso específico: Mesma temperatura

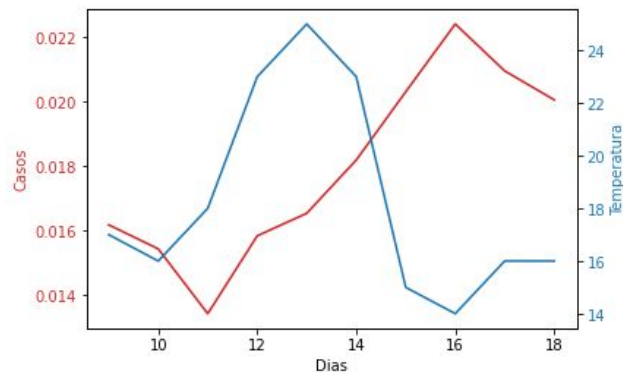
Netherlands - 7/2020



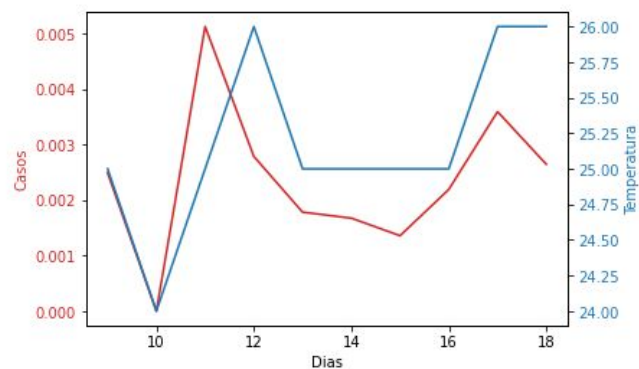
Côte d'Ivoire - 7/2020



Netherlands - 9/2020



Côte d'Ivoire - 9/2020





Conclusão

Observamos que a influência da temperatura no aumento de casos confirmados é mínima e irrelevante com relação à influência que o comportamento social exerce.