Qual o nível de risco de cada cidade?

Foi utilizado o código abaixo para gerar a quantidade de casos para cada município, através de um gráfico.

Na primeira parte, foi realizada a filtragem das 5 cidades escolhidas para a analise no banco de dados.

```
1
2   ```
    Run Cell | Run Below | Debug Cell
3   #%% Filtrando pelas 5 cidades selecionadas
4   query_df_city = pd.DataFrame()
5   query = """
6       SELECT c.nome
7       FROM cidade c
8   """
9
10  query_df_city = get_data_from_db(query, query_df_city, ['nome'])
11
```

Posteriormente, foi realizada a filtragem dos dados por cidade, além da formatação da primeira coluna do DataFrame criado.

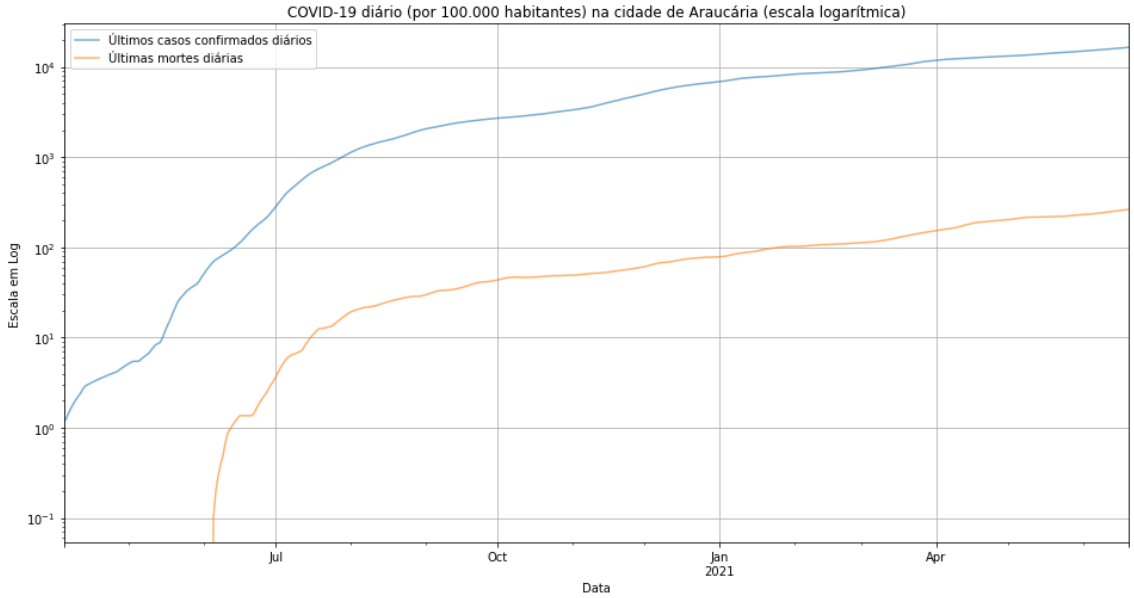
```
    Run Cell | Run Above | Debug Cell
12  #%% Filtrando pela cidade e formatando a primeira coluna para a Data facilitando o plot
13  def filter_and_format_by_city(city_name):
14      df_city_filter = query_df[query_df['nome'] == city_name]
15
16      df_city_filter.data = pd.to_datetime(df_city_filter.data, format="%Y-%m-%d")
17      df_city_filter.set_index("data", inplace=True)
18      df_city_filter.index.name = "Data"
19      return df_city_filter
```

Por fim, foi feita uma função para a construção do gráfico, criando uma variável onde irão ser inseridos os dados de cada cidade.

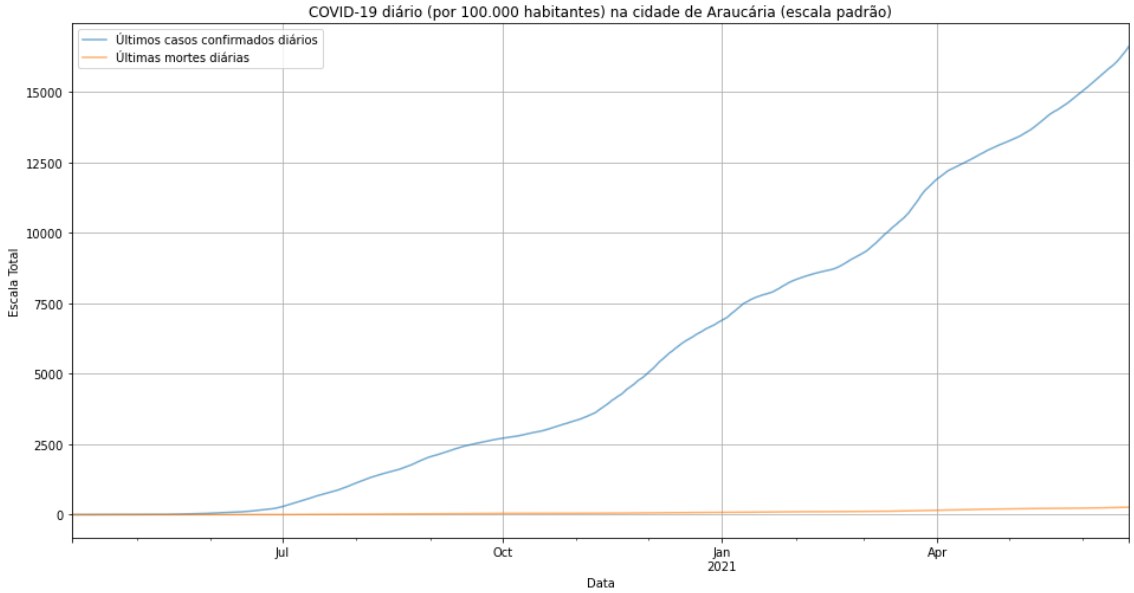
```
    Run Cell | Run Above | Debug Cell
21  #%% Construção do gráfico diário com o cálculo das ultimas mortes e confirmados dividido pela população a cada 100000 habitantes
22
23  def plot_df_by_last_case_and_deaths_by_100000_habitants(df, city_name):
24      ax = (
25          (100000 * df["ultimos_confirmados"] / df["populacao_estimada"])
26          .rolling(7, center=True)
27          .mean()
28          .plot(style="-", figsize=FS, logy=True, alpha=0.6)
29      )
30      ax = (
31          (100000 * df["ultimas_mortes"] / df["populacao_estimada"])
32          .rolling(7, center=True)
33          .mean()
34          .plot(style="-", ax=ax, logy=True, alpha=0.6)
35      )
36      ax.grid()
37      _ = ax.set(
38          title=f"COVID-19 diário (por 100.000 habitantes) na cidade de {city_name} (escala logarítmica)",
39          ylabel="Escala em Log",
40      )
41      _ = ax.legend(["Últimos casos confirmados diários", "Últimas mortes diárias"])
42      ax.autoscale(enable=True, axis="x", tight=True)
```

Araucária:

```
Run Cell | Run Above | Debug Cell
#%% Filtro, otimização e plot dos valores na cidade de Araucária

first_city = query_df_city.iat[0, 0]
df_first = filter_and_format_by_city(first_city)
plot_df_by_last_case_and_deaths_by_100000_habitants(df_first, first_city)

```

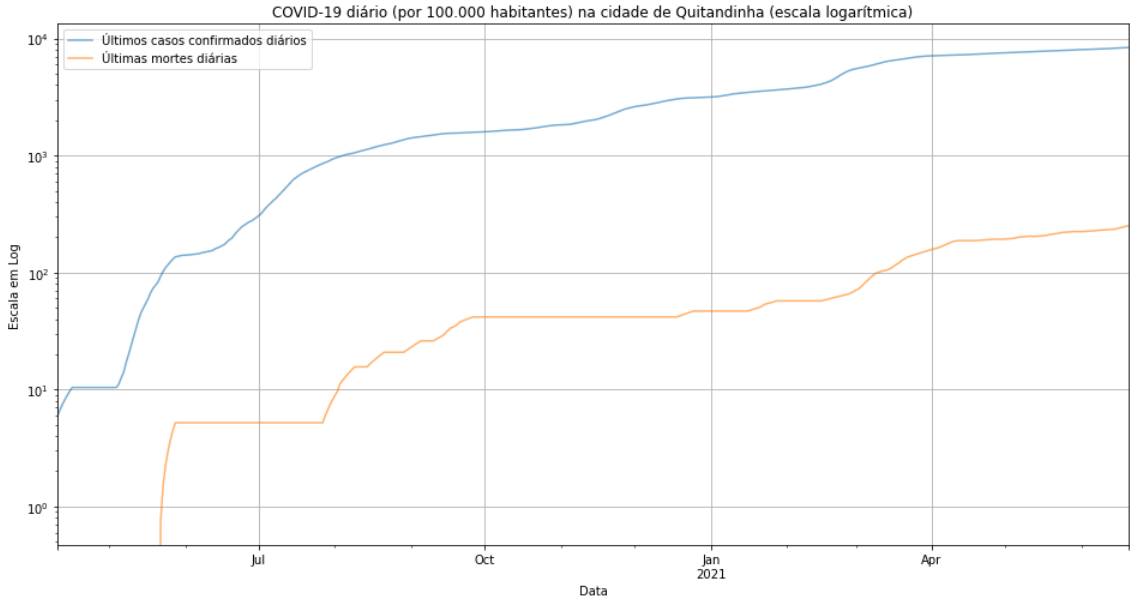


Como é possível ver, houve um crescimento exponencial, tanto do número de casos, como do número de mortes. A escala logarítmica do eixo y foi utilizada para facilitar a visualização dos dados, visto que a grande amplitude entre eles poderia causar uma distorção, bem como para demonstrar a brusca variação no crescimento.



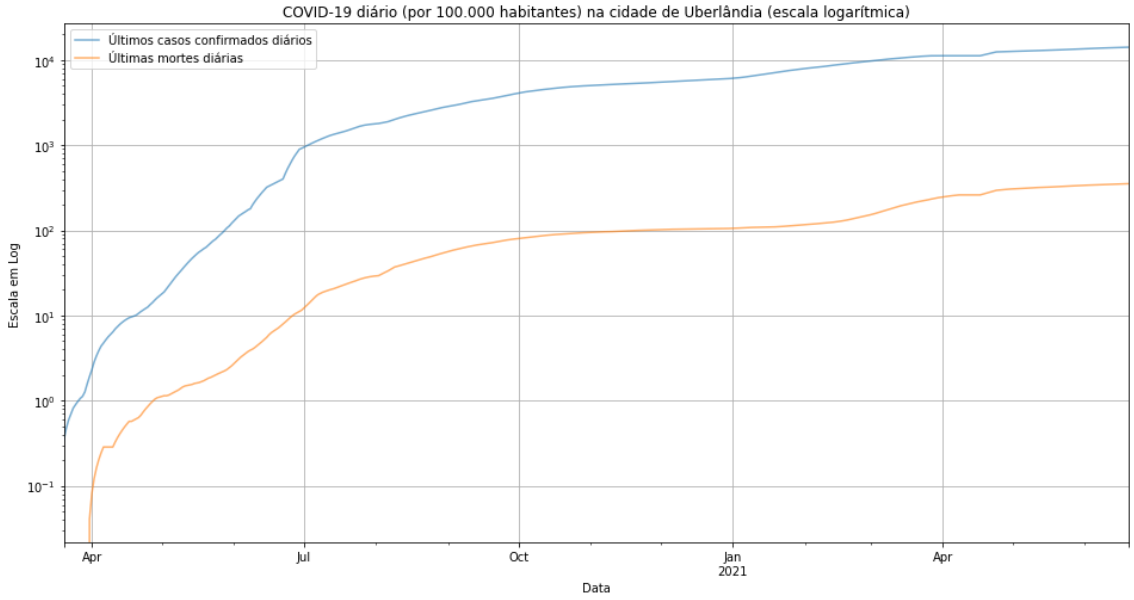
Quitandinha:

```
Run Cell | Run Above | Debug Cell
#%% Filtro, otimização e plot dos valores na cidade de Quitandinha

second_city = query_df_city.iat[1, 0]
df_second = filter_and_format_by_city(second_city)
plot_df_by_last_case_and_deaths_by_100000_habitants(df_second, second_city)
```



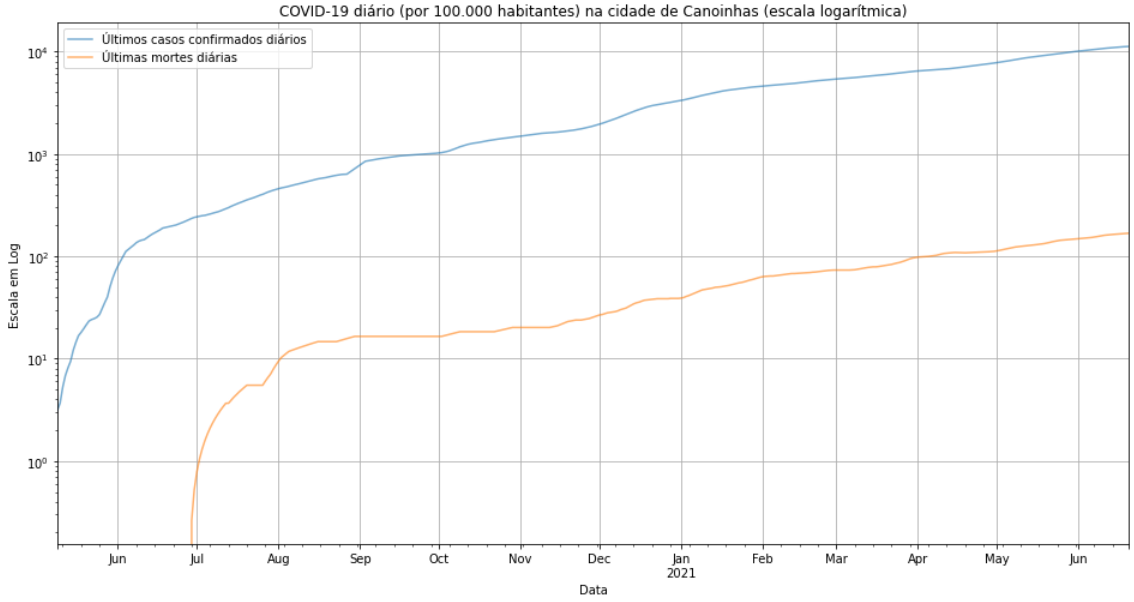
Uberlândia:

```
Run Cell | Run Above | Debug Cell
#%% Filtro, otimização e plot dos valores na cidade de Uberlândia

third_cirty = query_df_city.iat[2, 0]
df_third = filter_and_format_by_city(third_cirty)
plot_df_by_last_case_and_deaths_by_100000_habitants(df_third, third_cirty)
```



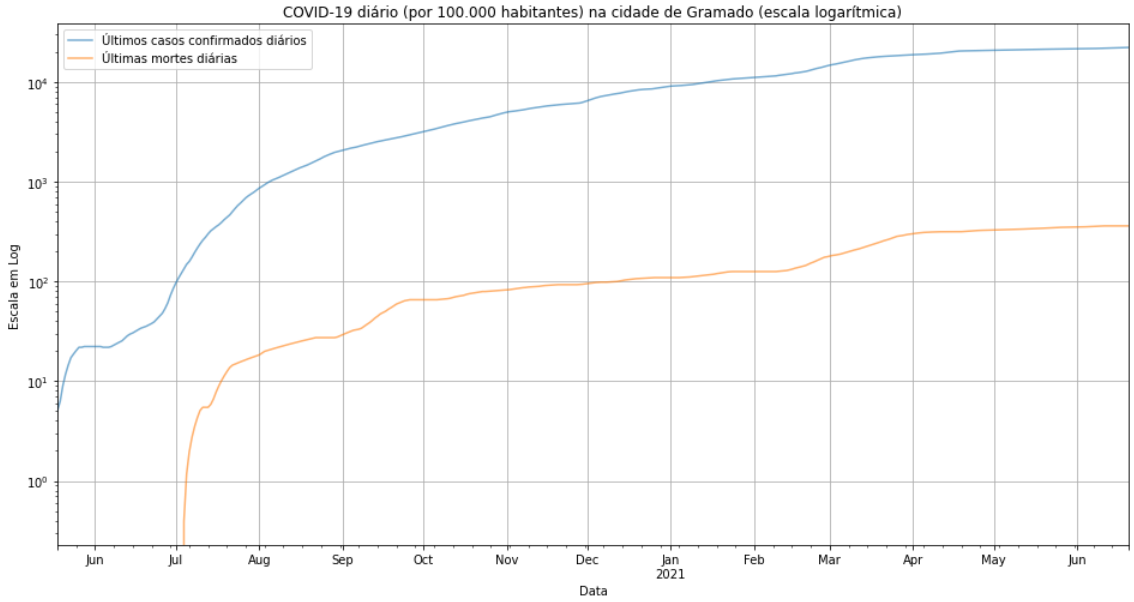
Canoinhas:

```
Run Cell | Run Above | Debug Cell
#%% Filtro, otimização e plot dos valores na cidade de Canoinhas

fourth_city = query_df_city.iat[3, 0]
df_fourth = filter_and_format_by_city(fourth_city)
plot_df_by_last_case_and_deaths_by_100000_habitants(df_fourth, fourth_city)
```



Gramado:

```
Run Cell | Run Above | Debug Cell
#%% Filtro, otimização e plot dos valores na cidade de Gramado

fifth_city = query_df_city.iat[4, 0]
df_fifth = filter_and_format_by_city(fifth_city)
plot_df_by_last_case_and_deaths_by_100000_habitants(df_fifth, fifth_city)
```



Quais medidas de prevenção deveriam ser aplicadas em cada cidade?

Como o nível de risco depende da taxa de crescimento de contaminação em cada cidade, as cidades que estão com taxas abaixo de 25% de crescimento necessitam de cuidados básicos, como distanciamento, evitar aglomerações, uso de álcool em gel e máscara em locais públicos.

Níveis entre 25% e 50% exigem implementações de quarentenas para pessoas de riscos, além da suspensão de atividades não essenciais com maior aglomeração e testagem.

Níveis entre 50% e 100% necessitam, além das medidas já adotadas pelos níveis mais baixos, proibições de eventos que causem grandes aglomerações, além de serviços de atendimento ao público que possam ser realizados de maneira remota e testagem.

Acima de 100% de crescimento, é recomendada a adoção de todas as medidas acima, bem como a implantação de quarentena obrigatória por, ao menos, 15 dias; triagem de pessoas que venham a entrar ou sair dos municípios e testagem.

Como foi visto, dentre as cidades analisadas, todas necessitam da adoção de medidas rígidas, pois o crescimento de casos está a meses descontrolado. Ainda que em alguns momentos a taxa tenha sido menor, o número de casos por 100000 habitantes demonstra um completo descontrole da infecção.

Dentre estas cidades, quais devem ter prioridade na alocação de recursos para o combate àCOVID-19?

A alocação de recursos deve priorizar cidades que possuam maior carência do sistema de saúde, ou que possuam uma taxa alta de transmissões, para que se possa haver o controle das transmissões o mais rápido possível.

Sendo assim, conforme o gráfico abaixo, pode-se notar que as cidades que deve possuir maior envio de recursos são Gramado e Uberlândia, pois além de possuírem uma população maior, também estão com os maiores números de mortes e casos de COVID-19 a cada 100000 habitantes.

```
Run Cell | Run Below | Debug Cell
#%% Otimização por Data de todos os valores
query_df_new = query_df
query_df_new.data = pd.to_datetime(query_df_new.data, format="%Y-%m-%d")
query_df_new.set_index("data", inplace=True)
query_df_new.index.name = "Data"
print(query_df)

Run Cell | Run Above | Debug Cell
#%% Cálculo das ultimas mortes por população com todos os valores da otimização a cada 100000 habitantes

covid_rate_all_cities = query_df_new[["nome", "ultimas_mortes", "populacao_estimada"]]
covid_rate_all_cities["covid_rate"] = 100000 * covid_rate_all_cities['ultimas_mortes'] / covid_rate_all_cities["populacao_estimada"]
covid_rate_all_cities.drop(["populacao_estimada", "ultimas_mortes"], axis=1, inplace=True)

covid_rate_all_cities = covid_rate_all_cities.pivot_table(index="Data", columns="nome", values="covid_rate")

Run Cell | Run Above | Debug Cell
#%% Plot agregado das cidades
ax = covid_rate_all_cities.rolling(7, center=True).mean().plot(figsize=FS, alpha=0.6)
ax.grid()
_ = ax.set(
    title="COVID-19 diário (por 100.000) nos Munícipios escolhidos",
    ylabel="Total de mortes COVID-19",
)
ax.autoscale(enable=True, axis="x", tight=True)
```