MonitoraCovid

Bruno Vieira 11201811306
Felipe Martins 21012014
Gabriel Migliorini 11201721812

1. Introdução

O sistema MonitoraCovid foi desenvolvido com o proprósito de ser uma ferramenta flexível e automatizada para acompanhar o avanço da pandemia.

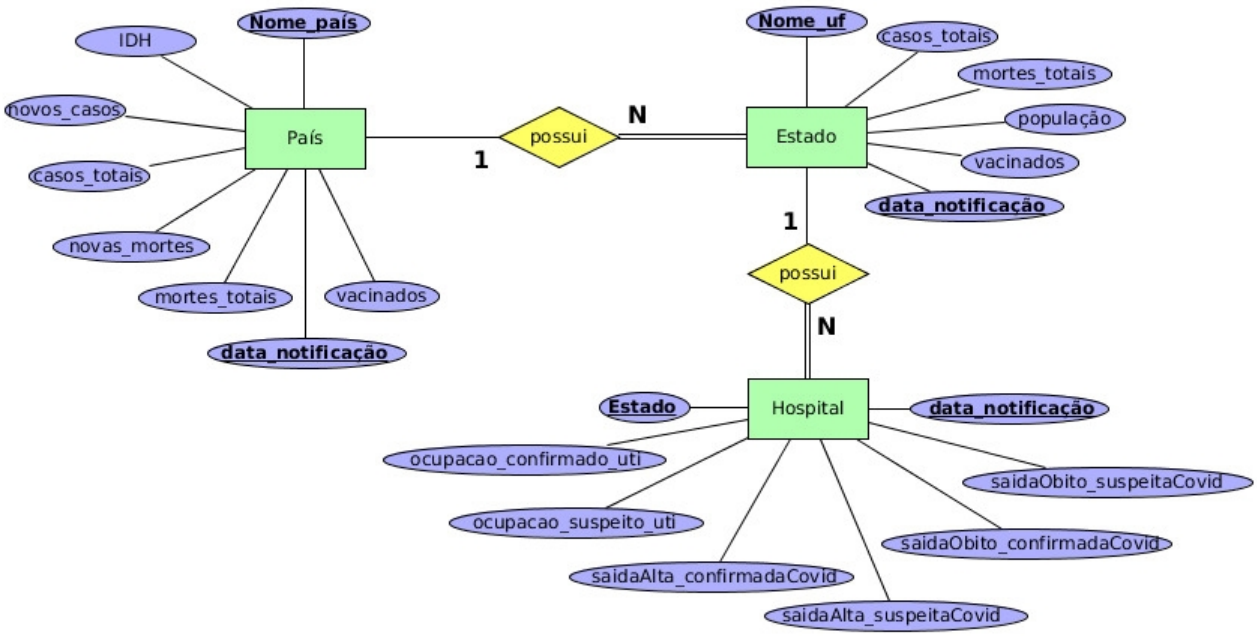
Para isso, o sistema utiliza bots que capturam automaticamente as bases nacionais e internacionais disponíveis na internet. Posteriormente, são geradas tabelas de acordo com os modelos aqui apresentados. Por fim, são disponibilizadas funções de consulta pré-estabelecidas assim como a disponibilidade de realização de consultas próprias.

A implementação foi realizada utilizando PostgresSQL e Python. Bibliotecas auxiliares tais como: psycopg , tkinter, pandas, dask e matplotlib foram utilizadas para conexão com o banco de dados, criação de interfaces, manipulação de consultas e plotagem.

No mais, é importante ressaltar que foram utilizadas bases já existentes. Assim, as restrições de modelagem são impostas pelas mesmas – isso ficará evidente na transição de modelo ER para Lógico.

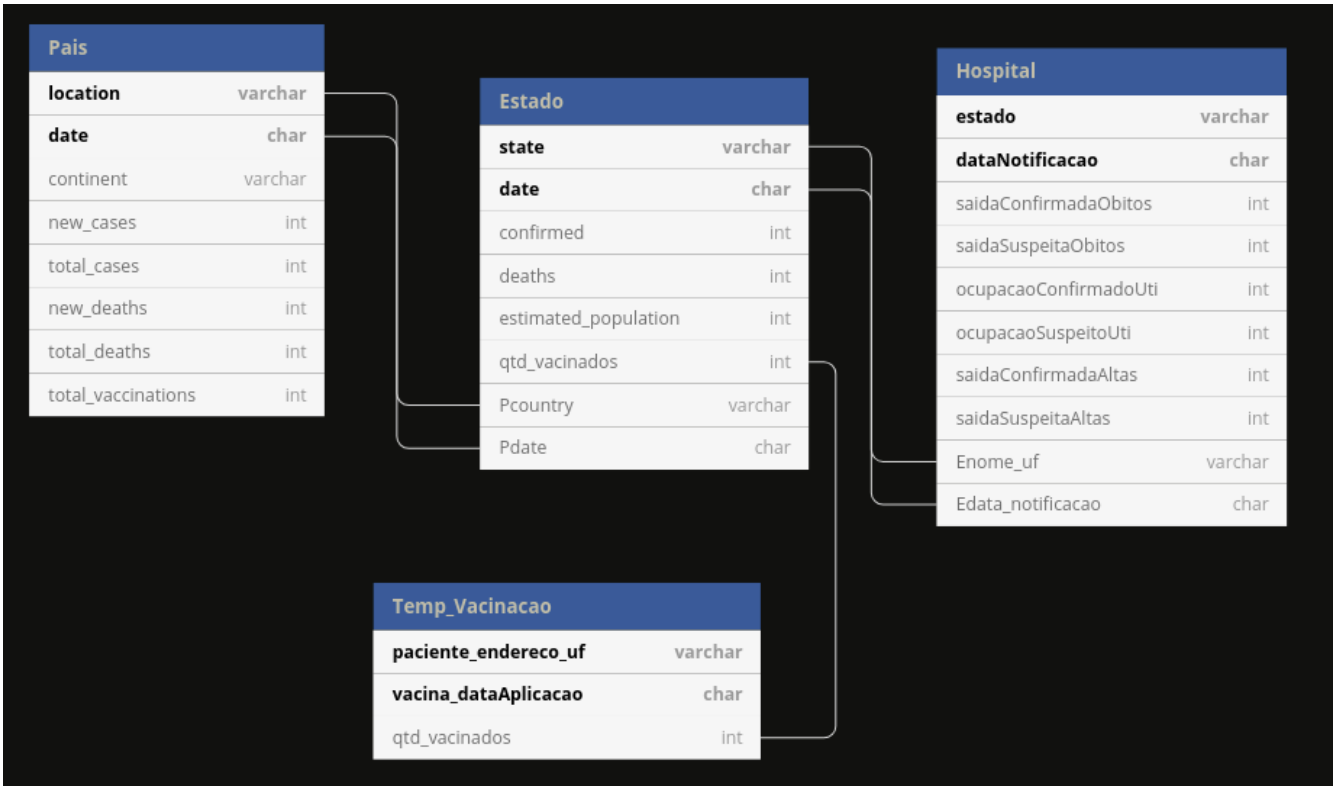
2. Modelo Entidade-Relacionamento

Nosso modelo é constituído por 3 entidades: País, Estado e Hospital. Dentre as escolhas que fizemos está a simetria de atributos entre País e Hospital. Com exceção do IDH, os demais atributos estão presentes (ou podem ser gerados) na entidade País: isso se reflete nas funcionalidades mostradas posteriormente.



3. Modelo Lógico

A partir do modelo ER estabelecemos os tipos de cada atributo assim como as ligações entre as entidades. Porém, como dito anteriormente, aqui surge a primeira restrição imposta à modelagem. Isso se dá uma vez que a base que caracteriza a entidade Estado não possui informação acerca de vacinas. Em função disso, o modelo lógico inclui uma tabela temporária que informa a origem do atributo qtd_vacinados.



4. Modelo Físico

A criação do modelo físico é feita em duas etapas. Primeiro são carregadas as bases completas disponíveis na internet. Posteriormente, são criadas tabelas a partir do modelo lógico apresentado – mostrado no código abaixo.

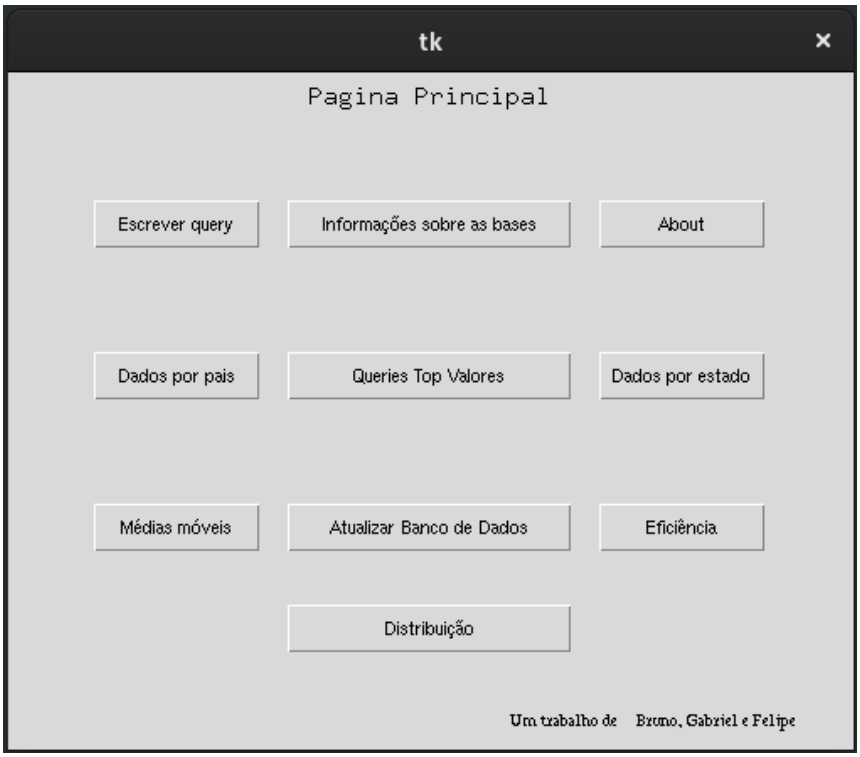
```
CREATE TABLE IF NOT EXISTS  Pais            (location varchar(255), date char(10), continent varchar (255), new_cases integer,
                                             total_cases integer, new_deaths integer, total_deaths integer, total_vaccinations integer,
                                             populacao integer, primary key(location, date));

CREATE TABLE IF NOT EXISTS  Estado          (state varchar(2), date char(10), confirmed integer, deaths integer,
                                             qtd_vacinados integer, Pcountry varchar(255), Pdate char(10),
                                             populacao integer, primary key(state, date));

CREATE TABLE IF NOT EXISTS  temp_vacinacao  (paciente_endereco_uf varchar(2), vacina_dataAplicacao char(10),
                                             qtd_vacinados integer, primary key(paciente_endereco_uf,vacina_dataAplicacao));

CREATE TABLE IF NOT EXISTS  Hospital        (estado varchar(255), dataNotificacao char(10), saidaConfirmadaObitos integer,
                                             saidaSuspeitaObitos integer, ocupacacaoConfirmadoUti integer, ocupacaoSuspeitoUti integer,
                                             saidaConfirmadaAltas integer, saidaSuspeitaAltas integer, Enome_uf varchar(2),
                                             Edata_notificacao varchar(10), primary key(estado,dataNotificacao));
```

5. Principais funcionalidades

Como dito anteriormente, nosso modelo é simétrico. Assim, para toda função em escala Global haverá um equivalente em escala Nacional.

Antes de listar as funcionalidades que envolvem o uso de consultas pré-determinadas vamos listar as funcionalidades 'passivas': **Escrever query, Atualizar Banco de Dados, Informações sobre as Bases** e **Baixar Base Completa**. Com exceção da última funcionalidade todas demais podem ser visualizadas na interface abaixo. Essas funções são importantes pois permitem que o usuário realize uma consulta própria (e salve) a partir do modelo lógico apresentado nas informações – tornando o programa flexível. Assim como também permite que as bases se mantenham atualizadas – tornando o programa automatizado.

Já em relação as funções que envolvem o uso de SQL temos: **Dados por país, Dados por Estado, Médias móveis, Eficiência e Distribuição**. As duas primeiras retornam todos dados presentes na respectiva tabela. As demais, que possuem opções **'Global'** e **'Nacional'**, serão apresentadas na sequência.

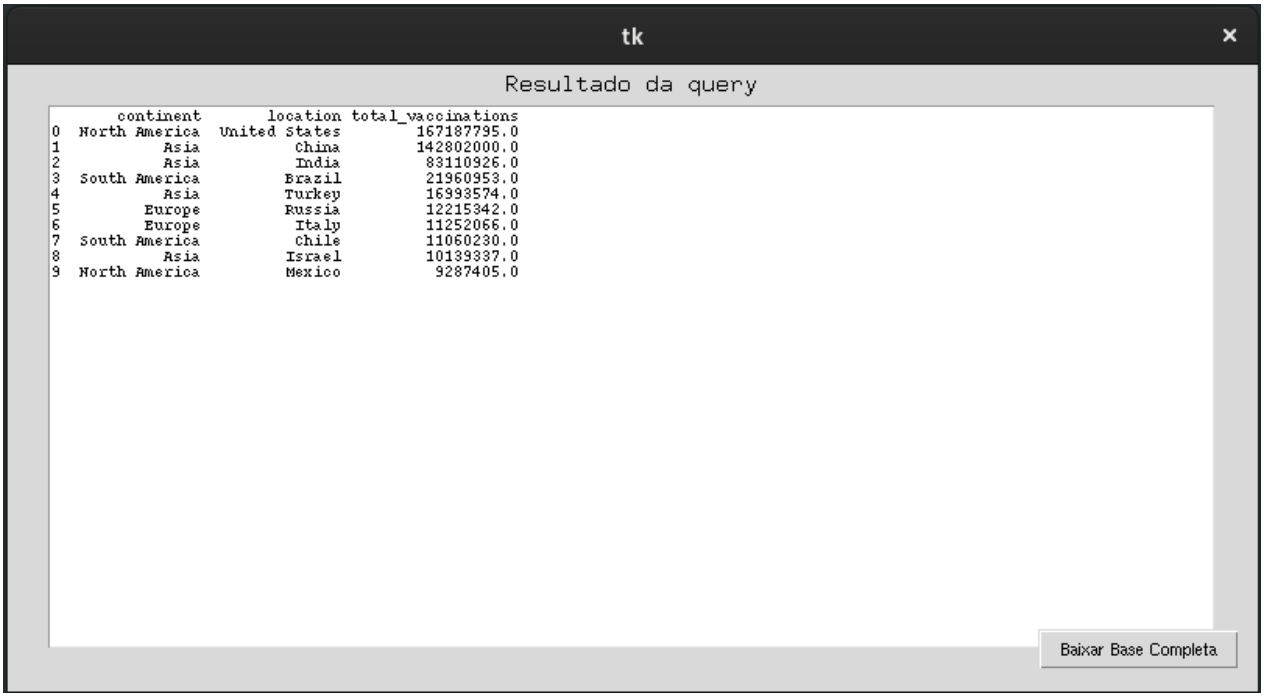


Top10

A função Top10 retorna uma lista de 10 países em relação a um critério e ordenamento específico. Dentre os critérios possíveis estão: contaminações, morte, taxa de morte e vacinação. O ordenamento pode ser realizado de modo decrescente (ex: top10 menos mortes) ou crescentes. Vale ressaltar que os valores são selecionados a partir da data mais atual disponível.

O código abaixo apresenta a generalização da função para que todas combinações possam ser geradas a partir de um mesmo código. Em seguida apresentamos o resultado da consulta (opção Global) juntamente com a opção de baixá-la.

```
SELECT state, ''' + criterio + '''
FROM (SELECT A.state, A.confirmed, A.deaths,
      cast(A.deaths as double precision) / cast(A.confirmed as double precision) as death_rate

      FROM estado as A, (SELECT state, max(date) as date FROM estado GROUP BY state) as B
      WHERE A.state = B.state and A.date = B.date) as A
      ORDER BY ''' + criterio + ''' ''' + ordenamento + ''' limit 10;
```



	continent	location	total_vaccinations
0	North America	United States	167187795.0
1	Asia	China	142802000.0
2	Asia	India	83110926.0
3	South America	Brazil	21960953.0
4	Asia	Turkey	16993574.0
5	Europe	Russia	12215342.0
6	Europe	Italy	11252066.0
7	South America	Chile	11060230.0
8	Asia	Israel	10139337.0
9	North America	Mexico	9287405.0

Eficiencia

Aqui estabelecemos um indicador comparativo de eficiência. O indicador é calculado do seguinte modo: (1 – taxa_morte). Contudo, reconhecemos a métrica como redundante. Para eliminar a redundância tentamos utilizar um outro estimador: (1 – log(mortes_totais) / casos_por_milhão). Embora essa nova forma tenha corrigido algumas distorções acabou criando outras. A imagem abaixo mostra a versão inicial do estimador aplicado aos estados brasileiros.

```sql
SELECT A.date, A.state, 1 - (CAST(A.deaths AS FLOAT) / CAST(A.confirmed AS FLOAT)) as eficiency
FROM estado as A, (SELECT state, max(date) as data FROM estado GROUP BY state) as B
WHERE A.state = B.state and A.date = B.data
ORDER BY 2 desc limit 10;
```

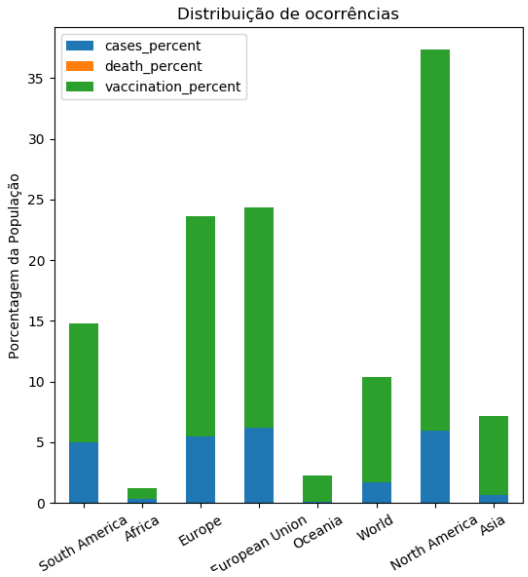
Distribuição

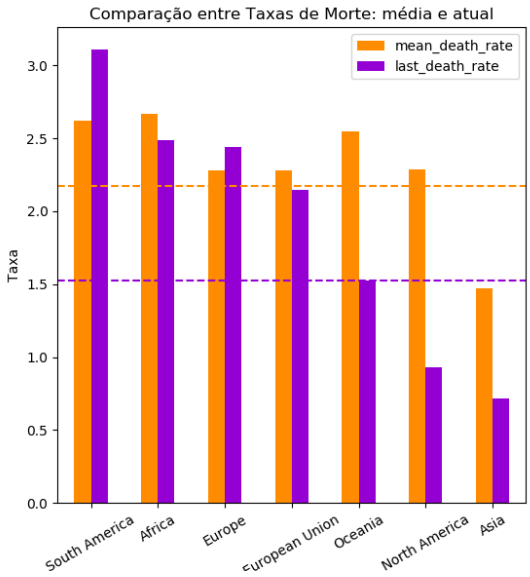
Nessa funcionalidade, apresentamos um comparativo geral entre continentes. Para isso criamos variáveis para porcentagem de casos, mortes de vacinação. O intuito aqui é comparar **como** foram orientadas as políticas para o controle da doença. Por exemplo, um país que possui uma porcentagem alta de casos em relação à população não realizou uma quarentena eficaz. Ao passo que uma alta porcentagem de mortes indica um problema na estrutura médica e logística. Por fim, há a porcentagem de vacinados.

Utilizando as três informações em conjunto compreendemos como os governos lidaram com a doença. Países que apresentam um alta porcentagem de mortes e baixa porcentagem de vacinação realizam políticas *errôneas* voltadas à ‘imunidade de rebanho’. Vale ressaltar que parte dessa funcionalidade está no comparativo de taxas de mortes (média e atuais). Isso porque a porcentagem de mortes em uma localidade não apresenta significância em um gráfico – ainda que milhares de pessoas tenham morrido.

Assim, também comparamos a taxa média de mortes e a última taxa reportada. Também inserimos a taxa média dentre as localidades observadas. A função abaixo mostra a consulta necessária para gerar as variáveis descritas. Em seguida temos os gráficos comparativos gerados por esses dados.

```sql
SELECT *, death_percent + vaccination_percent + cases_percent as total_percent
FROM (SELECT A.continent, A.location,
         CAST(total_cases AS FLOAT) * 100  / CAST(A.populacao AS FLOAT)  as cases_percent,
         CAST(A.new_deaths AS FLOAT) * 100  / CAST(A.populacao AS FLOAT)  as death_percent,
         CAST(A.total_vaccinations AS FLOAT) * 100  / CAST(A.populacao AS FLOAT)  as vaccination_percent,
         CAST(total_deaths as FLOAT) * 100 / CAST(total_cases as FLOAT) as mean_death_rate,
         CAST(new_deaths as FLOAT) * 100 / CAST(new_cases as FLOAT) as last_death_rate

     FROM pais as A, (SELECT location, max(date) as date FROM pais GROUP BY location) as B
     WHERE A.location = B.location and A.date = B.date and A.new_cases > 0 and total_cases > 0) as consolidado
ORDER BY last_death_rate desc;
```

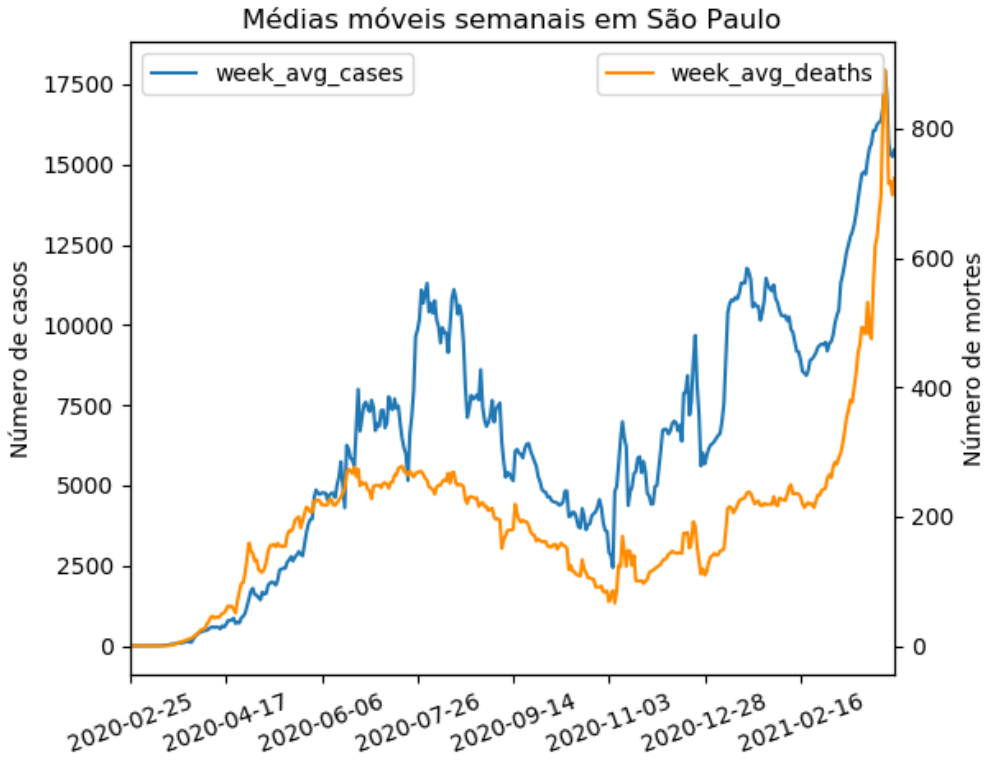




Média Móvel

Por fim, há a funcionalidade de médias móveis. A função abaixo realiza o cálculo da média em uma janela de 7 dias: tanto para mortes quanto para casos. Em seguida utilizamos o resultado para filtrar localidades e plotar um exemplo. A opção Global conta com um gráfico das médias móveis do Brasil enquanto a opção Nacional possui um gráfico em relação à cidade de São Paulo (mostrado abaixo).

```
SELECT date, state, week_avg_deaths,
        avg(CAST(confirmed AS FLOAT)) OVER(
        PARTITION BY state
        ORDER BY date
        ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)
        AS week_avg_cases
FROM    (SELECT *,
        avg(CAST(deaths AS FLOAT)) OVER(
        PARTITION BY state
        ORDER BY date
        ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)
        AS week_avg_deaths
        FROM estado) AS aux;
```



6. Conclusão

O MonitoraCovid é um projeto conciso para acompanhamento da COVID19. O projeto em si foi importante não somente de consolidação do que aprendemos ao longo do quadrimestre mas também para o entendimento das restrições impostas pelos dados já existentes. Além disso, também houve aprendizado na utilização prática de ferramentas que conectam o banco com a interface em si.

Reconhecemos que existem melhorias a serem feitas, dentre elas, a criação de indicadores para hospitais - pouco explorada nesta versão. Por fim, consideramos um projeto bem sucedido em função do tempo disponível e do conhecimento que tínhamos ao iniciar o projeto.

Link para vídeo e código:
https://drive.google.com/drive/folders/1ak4GChgrIhhP-W6iLWV7p2C7GhX-8-e-?usp=sharing