

Aplicação de Consulta e Gerenciamento de Dados da COVID-19 no Brasil através do ALERTA-19

Projeto desenvolvido por: Éverson Filipe Campos da Silva Moura e Vinicius Durval de Moraes

Disciplina: Desenvolvimento Rápido em Python (DEV RÁPIDO EM PYTHON)

Professor: Sebastião Rogério

Instituição: UniFavip Wyden

1. Título do Projeto

Aplicação de Consulta e Gerenciamento de Dados da COVID-19 no Brasil - ALERTA-19

2. Introdução

Este projeto consiste no desenvolvimento de uma aplicação CRUD (Create, Read, Update, Delete) para acesso e visualização de dados da pandemia de COVID-19 no Brasil. Concebido no âmbito da disciplina "Desenvolvimento Rápido em Python (DEV RÁPIDO EM PYTHON)" da UniFavip Wyden, ministrada pelo Professor Sebastião, o trabalho reflete a ênfase do curso em metodologias ágeis de desenvolvimento, bancos de dados e interfaces gráficas.

O problema central que a aplicação se propõe a resolver é a dificuldade em consultar e gerenciar informações atualizadas e históricas sobre a COVID-19 no Brasil de forma acessível e interativa. A complexidade de lidar diretamente com grandes volumes de dados epidemiológicos brutos torna a compreensão e a exploração dessas informações um desafio para o público em geral. Neste cenário, a importância do Desenvolvimento Rápido de Aplicações (RAD) é manifesta, demonstrando como essa metodologia permite a construção ágil de uma solução funcional e robusta a partir de um dataset real, validando os princípios da disciplina. Este relatório documenta o processo de desenvolvimento e os resultados alcançados.

3. Justificativa

A motivação para a escolha deste projeto reside na relevância contínua dos dados de saúde pública, especialmente aqueles relacionados à pandemia de COVID-19. Há uma demanda real e persistente por ferramentas que simplifiquem o acesso e a interpretação de grandes volumes de dados epidemiológicos brutos, que frequentemente são complexos e dispersos.

O projeto surge como uma resposta a essa necessidade, oferecendo uma solução interativa que democratiza o acesso à informação da COVID-19 para estudantes, pesquisadores e o público em geral. Ao transformar dados brutos em uma interface acessível, a aplicação contribui significativamente para a compreensão da pandemia e suas implicações. Além disso, o desenvolvimento deste sistema oferece uma valiosa experiência prática, aprofundando a formação acadêmica dos envolvidos nos princípios e técnicas do "Desenvolvimento Rápido em Python".

4. Objetivos

4.1. Objetivo Geral

Construir uma aplicação CRUD em Python capaz de consultar, gerenciar e apresentar dados da COVID-19 no Brasil de forma intuitiva e eficiente.

4.2. Objetivos Específicos

- Implementar funcionalidades CRUD para manipulação de dados de casos de COVID-19.
- Integrar a aplicação a um banco de dados relacional (SQLite) para persistência e recuperação otimizada dos dados.
- Desenvolver uma interface gráfica de usuário (GUI) intuitiva utilizando bibliotecas Python, como Tkinter para a parte gráfica e Flask para a camada de serviço/API, garantindo a interação fluida com o backend.
- Permitir a filtragem e visualização de dados com base em critérios como estado, município e data, exibindo métricas chave como casos confirmados, óbitos e novos casos diários.
- Demonstrar a aplicação prática de metodologias de desenvolvimento rápido e ágil na construção de uma solução funcional.

5. Base de Dados

A base de dados central para este projeto é o dataset público "COVID-19 no Brasil" do Brasil.IO, especificamente o arquivo `caso_full.csv.gz`. Este dataset é acessível publicamente através da URL:

https://data.brasil.io/dataset/covid19/caso_full.csv.gz. O Brasil.IO é uma iniciativa notável que agrega informações sobre casos confirmados e óbitos por SARS-CoV-2 em níveis estadual e municipal no Brasil, compilando dados de secretarias de saúde.

O formato dos dados é CSV, compactado em GZ, e o arquivo possui um tamanho considerável de aproximadamente 88.15MB, indicando um vasto volume de informações. As colunas mais relevantes, essenciais para as funcionalidades de consulta e gerenciamento da aplicação, e seus prováveis significados, são inferidas como:

- `date`: Data do registro dos dados.

- **state**: Nome do estado brasileiro (ex: 'SP', 'RJ').
- **city**: Nome do município brasileiro.
- **confirmed_cases**: Número acumulado de casos confirmados até a data.
- **deaths**: Número acumulado de óbitos até a data.
- **new_cases**: Contagem diária de novos casos confirmados.
- **new_deaths**: Contagem diária de novos óbitos.
- **estimated_population**: População estimada da localidade para cálculo de taxas.
- **is_last**: Indicador booleano se é o último registro disponível para a data/localidade.
- **place_type**: Tipo de localidade (ex: 'city', 'state').
- **city_ibge_code**: Código IBGE do município, para georreferenciamento.
- **epidemiological_week**: Semana epidemiológica do registro.
- **last_available_date**: Última data em que os dados foram atualizados para aquele registro.
- **last_available_confirmed**: Último número de casos confirmados disponível para a localidade.
- **last_available_deaths**: Último número de óbitos disponível para a localidade.
- **new_recovered**: Contagem diária de novos pacientes recuperados.
- **accum_recovered**: Número acumulado de pacientes recuperados.
- **source**: Fonte específica do dado.

Os pré-processamentos realizados sobre este dataset incluem a descompactação do arquivo **.gz**, o carregamento eficiente dos dados para o banco de dados SQLite para otimização das operações, e o tratamento de valores ausentes (NaN). Adicionalmente, foram realizadas conversões de tipos de dados (por exemplo, de strings para datas e inteiros) para adequá-los ao modelo do banco de dados e garantir a correta manipulação e filtragem pela aplicação.

6. Tecnologias Utilizadas

O desenvolvimento da aplicação foi alicerçado nas seguintes tecnologias:

- **Linguagem de Programação**: Python 3.x, escolhida por sua sintaxe clara, vasta gama de bibliotecas e capacidade de desenvolvimento rápido.
- **Frameworks/Bibliotecas**:
 - **Flask**: Utilizado para a estrutura web da aplicação, servindo como uma API para gerenciar as rotas e a comunicação com o backend dos dados.
 - **Tkinter**: Empregada para a construção da interface gráfica do usuário (GUI) desktop. A integração entre Tkinter e Flask permite que a GUI interaja com a camada de serviço web, garantindo uma aplicação robusta e responsiva.
 - **SQLite3**: Selecionado como sistema de gerenciamento de banco de dados relacional para armazenamento local e persistente dos dados da COVID-19, facilitando as operações de CRUD.

- **Pandas:** Essencial para a manipulação e o pré-processamento eficiente do dataset CSV inicial, incluindo leitura, limpeza, e preparação dos dados antes do carregamento no SQLite.
- **Ferramentas de Desenvolvimento:**
 - **VSCode (Visual Studio Code):** Ambiente de desenvolvimento integrado (IDE) que proporcionou um ambiente produtivo com recursos de depuração e extensão.
 - **Git:** Sistema de controle de versão distribuído, fundamental para o gerenciamento das alterações no código-fonte.
 - **GitHub:** Plataforma de hospedagem de código-fonte que facilitou a colaboração e o versionamento do projeto.

7. Metodologia de Desenvolvimento

O projeto foi conduzido sob os princípios do Desenvolvimento Rápido de Aplicações (RAD) e metodologias ágeis, com um foco claro em ciclos de desenvolvimento curtos, prototipagem ágil e feedback contínuo. Adotou-se uma abordagem iterativa e incremental, permitindo a construção progressiva das funcionalidades CRUD e a adaptação a novos requisitos ou insights ao longo do processo, alinhado às diretrizes da disciplina.

As principais etapas de desenvolvimento incluíram:

1. **Análise de Requisitos e Coleta de Dados:** Compreensão das necessidades da aplicação para consulta e gerenciamento de dados da COVID-19 e aquisição do dataset do Brasil.IO.
2. **Design do Modelo de Dados e Schema do Banco:** Definição da estrutura do banco de dados SQLite e do modelo para armazenamento dos dados.
3. **Prototipagem Rápida da Interface e Funcionalidades CRUD:** Criação de protótipos iniciais da interface e implementação das funcionalidades essenciais, com foco primário na operação de *Read* (consulta).
4. **Desenvolvimento Iterativo:** Ciclos contínuos de codificação, testes e refinamento das funcionalidades, garantindo a entrega incremental de valor.
5. **Testes de Funcionalidade e Integração:** Verificação da correta operação das funcionalidades e da integração entre os componentes da aplicação (GUI, Flask, SQLite).

Como um projeto individual, a organização do trabalho priorizou a gestão eficaz do tempo e a definição de um Produto Mínimo Viável (MVP) para assegurar a entrega das funcionalidades essenciais dentro dos prazos da disciplina. A abordagem RAD foi crucial para construir uma solução funcional em prazos limitados, com validação contínua e feedback como pilares do processo.

8. Resultados e Funcionalidades

A aplicação desenvolvida materializa-se como uma ferramenta robusta para o acesso e gerenciamento de dados da COVID-19 no Brasil, resolvendo o desafio de consultar e

interagir com grandes volumes de informações epidemiológicas. As funcionalidades implementadas são:

- **Carga de Dados:** O sistema permite o carregamento inicial do dataset `caso_full.csv.gz` para o banco de dados SQLite. Esta etapa é fundamental para preparar os dados para as operações de consulta e gerenciamento, garantindo que a aplicação trabalhe com informações otimizadas para performance.
- **Consulta e Visualização (Read):** Esta é a funcionalidade central da aplicação, oferecendo uma interface intuitiva para exploração dos dados.
 - **Filtragem por Estado e Município:** Usuários podem consultar dados específicos filtrando por estados e municípios, permitindo análises regionalizadas.
 - **Filtragem por Período (Data):** É possível refinar as consultas por períodos específicos, obtendo informações sobre a evolução da pandemia em datas ou intervalos de tempo determinados.
 - **Exibição de Métricas Chave:** Os resultados das consultas apresentam métricas cruciais como o número acumulado de casos confirmados, óbitos, e a contagem diária de novos casos e óbitos, provendo uma visão clara do cenário epidemiológico. A interface gráfica facilita a apresentação desses resultados de forma compreensível.
- **Gerenciamento (CRUD - Criação, Atualização, Exclusão):** Embora a ênfase principal seja na consulta, a aplicação foi projetada com capacidade para as operações de Criação, Atualização e Exclusão, contextualizadas para o cenário de um dataset epidemiológico:
 - **Criação/Inserção:** Capacidade de adicionar novos registros ao banco de dados local. Isso pode ser útil para fins de teste, simulação de novos dados, ou para a importação de versões atualizadas do dataset, garantindo que o banco de dados esteja sempre alinhado com as informações mais recentes.
 - **Atualização:** Permite a modificação de registros existentes. Esta funcionalidade é aplicável para correção de dados, atualização de informações ou para integrar dados de outras fontes, mantendo a integridade e a acurácia do repositório local.
 - **Exclusão:** Possibilita a remoção de registros específicos. Útil para limpeza de dados inconsistentes, remoção de entradas duplicadas ou gerenciamento do volume de dados armazenados.

A aplicação se destaca por oferecer uma interface amigável que democratiza o acesso a dados complexos, permitindo que estudantes e pesquisadores realizem análises eficazes sem a necessidade de conhecimento aprofundado em programação ou banco de dados. Prints das telas e demonstrações em GIF das funcionalidades estão disponíveis no repositório GitHub do projeto, conforme o [README.md](#).

9. Conclusão

O desenvolvimento da "Aplicação de Consulta e Gerenciamento de Dados da COVID-19 no Brasil" foi uma jornada de aprendizado e aplicação prática de conceitos fundamentais da Ciência da Computação e do Desenvolvimento Rápido de Aplicações. Tecnicamente, a experiência consolidou conhecimentos sobre a integração entre frameworks web (Flask) e bibliotecas de interface gráfica (Tkinter), a manipulação eficiente de grandes datasets com Pandas, e o design e gestão de bancos de dados relacionais com SQLite. Do ponto de vista metodológico, o projeto demonstrou a eficácia dos princípios de RAD, como a prototipagem ágil e a iteração contínua, permitindo a entrega de um sistema funcional em um prazo limitado, e aprimorou as habilidades de gerenciamento de um projeto de software do início ao fim.

Entre os aspectos que funcionaram bem, destacam-se a robustez da integração com o banco de dados, que garantiu consultas eficientes e confiáveis; a usabilidade e a clareza da interface gráfica, que tornou a interação com os dados acessível a diferentes públicos; e a eficiência nas operações de consulta e filtragem, que permitiu a rápida obtenção de informações relevantes. A capacidade de entregar um Produto Mínimo Viável (MVP) funcional, atendendo aos requisitos da disciplina, foi um sucesso notável.

Para futuras iterações da aplicação, identificam-se oportunidades de melhoria que incluem: a otimização de performance para lidar com datasets ainda maiores de forma mais rápida; a implementação de funcionalidades de visualização avançadas, como gráficos interativos, para enriquecer a análise dos dados; a expansão das opções de filtragem e agregação, oferecendo maior flexibilidade nas consultas; e a incorporação de mais fontes de dados epidemiológicos para enriquecer o acervo de informações.

10. Repositório GitHub

O código-fonte completo e a documentação do projeto estão disponíveis no repositório GitHub:

<https://github.com/eversonfilipe/ALERTA-19>

O repositório contém:

- O código-fonte organizado da aplicação.
- O arquivo **README.md** com instruções detalhadas de uso.
- Imagens e GIFs demonstrativas do sistema em operação.
- O presente relatório técnico-acadêmico em formato **.pdf**.