Emerson Carvalho Azevedo

ANÁLISE DA MORBIDADE HOSPITALAR POR HIV NO BRASIL: UMA ABORDAGEM COM DADOS DO DATASUS E VISUALIZAÇÃO INTERATIVA EM POWER BI

ALEGRE - ES

2025

Emerson Carvalho Azevedo

ANÁLISE DA MORBIDADE HOSPITALAR POR HIV NO BRASIL: UMA ABORDAGEM COM DADOS DO DATASUS E VISUALIZAÇÃO INTERATIVA EM POWER BI

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Espírito Santo

Centro de Ciências Exatas, Naturais e da Saúde

Departamento de Computação

Orientador: Prof. Dsc. Bruno Vilela Oliveira

ALEGRE - ES

2025

Emerson Carvalho Azevedo

Análise da Morbidade Hospitalar por HIV no Brasil: Uma Abordagem com Dados do DATASUS e Visualização Interativa em Power BI

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 22 de agosto de 2025.

COMISSÃO EXAMINADORA

Documento assinado digitalmente
BRUNO VILELA OLIVEIRA
Data: 23/08/2025 19:59:16-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Dr. Bruno Vilela Oliveira
Universidade Federal do Espírito Santo
Orientador

Documento assinado digitalmente
GERALDO REGIS MAURI
Data: 22/08/2025 13:55:16-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo

Documento assinado digitalmente
HELDER DE AMORIM MENDES
Data: 22/08/2025 12:21:22-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Dr. Helder de Amorim Mendes
Universidade Federal do Espírito Santo

Agradecimentos

Primeiramente, agradeço a Deus e, em especial, aos meus pais por terem tornado este sonho possível, fornecendo toda a força e incentivo necessários ao longo da minha graduação. Também estendo meus agradecimentos aos demais familiares que sempre me apoiaram e acreditaram no meu potencial.

Sou grato a todos os professores da graduação pelos ensinamentos que contribuíram para minha formação acadêmica e pessoal. Um agradecimento especial ao meu orientador, Bruno, pelas dicas valiosas, pelo suporte constante, pela paciência durante a realização deste trabalho e pela amizade construída ao longo dessa jornada.

Agradeço também aos professores Geraldo e Helder por aceitarem fazer parte da banca examinadora e pela amizade que compartilhamos.

Aos amigos que fiz durante estes anos de faculdade, obrigado pelos momentos de alegria, apoio e descontração. Agradeço em especial a Otto, Cassio, Rafael, Andre, Romerson e Talles pelo companheirismo, pelas risadas e por se tornarem verdadeiros irmãos para mim.

Aos amigos que fiz na cidade de Alegre, saibam que levarei as memórias que compartilhamos sempre comigo.

Resumo

Este trabalho tem como objetivo propor uma abordagem para coleta, tratamento e visualização de dados de morbidade hospitalar no Brasil, com foco nas internações relacionadas à infecção pelo HIV, visando apoiar a tomada de decisão em saúde pública. Os dados foram extraídos do sistema DATASUS e tratados por meio de uma abordagem quantitativa e descritiva, envolvendo etapas de limpeza, padronização e estruturação dos registros, antes de serem armazenados em um banco de dados relacional. Para a etapa de visualização, foi utilizado o *Power BI*, permitindo a criação de dashboards interativos que facilitam a compreensão de padrões epidemiológicos e tendências de internações hospitalares ao longo do tempo. Os resultados evidenciam a importância do uso de ferramentas de *Business Intelligence* na área da saúde, destacando como a integração entre dados públicos, técnicas de tratamento de dados e visualização analítica podem contribuir para uma gestão mais eficiente e informada. O trabalho reforça o papel da tecnologia da informação como aliada no enfrentamento de desafios no setor de saúde.

Palavras-chave: Morbidade Hospitalar, DATASUS, *Power BI*, Visualização de Dados, Saúde Pública.

Abstract

This work aims to propose an approach for the collection, processing, and visualization of hospital morbidity data in Brazil, focusing on hospitalizations related to HIV infection, with the goal of supporting decision-making in public health. The data were extracted from the DATASUS system and processed through a quantitative and descriptive approach, involving stages of cleaning, standardization, and structuring of records before being stored in a relational database. For the visualization stage, Power BI was used, allowing the creation of interactive dashboards that facilitate the understanding of epidemiological patterns and trends in hospitalizations over time. The results highlight the importance of using Business Intelligence tools in the health sector, emphasizing how the integration of public data, data processing techniques, and analytical visualization can contribute to more efficient and informed management. The work reinforces the role of information technology as an ally in addressing challenges in the healthcare sector.

Keywords: Hospital Morbidity, DATASUS, Power BI, Data Visualization, Public Health.

Lista de Figuras

Figura 1 – Quadro *Kanban*. 21

Figura 2 – Arquitetura. 27
Figura 3 – Morbidade Hospitalar do SUS – Por local de Residência - Brasil 31
Figura 4 – Estrutura da Tabela Morbidade Hospitalar no PostgreSQL 37

Figura 5 – *Dashboard* Gráficos Interativos 39
Figura 6 – *Dashboard* Gráficos Interativos 40

Figura 7 – Morbidade Hospitalar do SUS – Por local de Residência - Brasil 47
Figura 8 – Períodos Disponíveis 47
Figura 9 – Seleção de variáveis categóricas para filtragem. 48
Figura 10 –Seleção das variáveis “Regime de Atendimento”e “Capítulo CID-10”. . 49
Figura 11 –Seleção de causas específicas na lista MORB CID-10. 49
Figura 12 –Seleção das variáveis “Faixa Etária 1”e “Faixa Etária 2”. 50
Figura 13 –Seleção das variáveis “Sexo”e “Cor/Raça”. 50

Lista de Tabelas

Tabela 1 – Testes Realizados Durante o Desenvolvimento 42

Lista de Listagens

3.1 Código de leitura e tratamento de arquivos CSV com Pandas 31
3.2 Bibliotecas 33
3.3 Configuração da Pasta e Lista de Arquivos 33
3.4 Leitura dos Arquivos 34
3.5 Junção dos Dados 34
3.6 Padronização dos Nomes das Colunas 35
3.7 Padronização dos Nomes das Colunas 35
3.8 Classificação do Tipo Local 35
3.9 Seleção das Colunas de Interesse 36

Lista de Abreviaturas e Siglas

AIDS	Síndrome da Imunodeficiência Adquirida
ANSI	Instituto Nacional Americano de Padrões
CID	Classificação Internacional de Doenças
CSV	Valores Separados por Vírgula
DATASUS	Departamento de Informação e Informática do SUS
DCCI	Doenças de Condições Crônicas e Infecções Sexualmente Transmissíveis
EDA	Análise Exploratória de Dados
ETL	Extração, Transformação e Carga
HIV	Vírus da Imunodeficiência Humana
IBGE	Instituto Brasileiro de Geografia e Estatística
JSON	Notação de Objeto JavaScript
KDD	Knowledge Discovery in Databases
LGPD	Lei Geral de Proteção de Dados
ML	Machine Learning
MORB	Morbidade
Power BI	Business Intelligence
RDBMS	Sistema de Gerenciamento de Banco de Dados Relacional
SICLOM	Sistema de Controle Logístico de Medicamentos
SISCEL	Sistema de Controle de Exames Laboratoriais
SIM	Sistema de Informações sobre Mortalidade
SINAN	Sistema de Informação de Agravos de Notificação
SQL	Linguagem de Consulta Estruturada
SUS	Sistema Único de Saúde
UF	Unidade Federativa
UFBA	Universidade Federal da Bahia

Sumário

1 Introdução **12**
1.1 O Problema e sua Importância 12
1.2 Objetivo Geral 13
1.3 Objetivos Específicos 14

2 Revisão de Literatura **15**
2.1 Análise de Dados em Morbidade Hospitalar e HIV 15
2.2 Trabalhos relacionados 16
2.3 Levantamento e Análise de Requisitos 17
2.4 Requisitos Funcionais 17
2.5 Requisitos Não Funcionais 18
2.6 Tecnologias Envolvidas 18
2.6.1 Jupyper Notebook 18
2.6.2 Python 19
2.6.3 Pandas 19
2.6.4 PostgreeSQL 19
2.6.5 Power BI 20
2.7 Kanban 20

3 Metodologia **22**
3.1 Processo de Desenvolvimento 22
3.1.1 Aplicação do *Kanban* na Organização do Projeto 22
3.2 Levantamento de Requisitos 22
3.2.1 Requisitos Funcionais 23
3.2.2 Requisitos Não Funcionais 23
3.3 Obtenção dos dados 24
3.4 Tratamento e preparação dos dados 24
3.5 Armazenamento dos dados 26
3.6 Visualização com *Power BI* 26
3.7 Arquitetura 26
3.7.1 Fonte de Dados (*Data Source*): 27
3.7.2 Camada de Extração e Tratamento (*Python*) 27
3.7.3 Armazenamento Estruturado (Banco de Dados) 28
3.7.4 Visualização Interativa (*Power BI*) 28
3.8 Delimitação do Projeto 29
3.9 Modelagem Banco de Dados 29

3.10 Obtenção e Preparação dos Dados 30
3.11 Metodologia do Desenvolvimento 30
3.12 Extração Manual de Dados do DATASUS 30
3.13 Pré-processamento e Limpeza dos dados 31
3.13.1 Bibliotecas Importadas 33
3.13.2 Configuração da Pasta e Lista de Arquivos 33
3.13.3 Leitura dos Arquivos 34
3.13.4 Junção dos Dados 34
3.13.5 Padronização dos Nomes das Colunas 34
3.13.6 Normalização do Nome da Coluna de Território 35
3.13.7 Classificação do Tipo Local 35
3.13.8 Seleção das Colunas de Interesse 35
3.14 Estrutura do Projeto Banco de Dados 36
3.15 Integração e construção do *Dashboard* no *Power BI* 37

4 Resultados 39
4.1 *Dashboard* Gráficos Interativos 39
4.2 Testes 40
4.2.1 Testes de Integração de Dados 41
4.2.2 Testes de Interface e Funcionalidade 41
4.2.3 Validação dos Indicadores 41

5 Conclusão 43
5.1 Trabalhos Futuros 43

Referências 45

A – Extração de dados TABNET 47

1 Introdução

A análise de dados sempre esteve presente no processo de tomada de decisões, mesmo que de forma intuitiva. Como exemplifica Carvalho, Menezes e Bonidia (2024), o médico Ignaz Semmelweis, no século XIX, ao examinar registros hospitalares, identificou a relação entre a falta de higienização das mãos e a alta mortalidade por febre puerperal. Hoje com o avanço das técnicas de análise, esse processo tornou-se sistemático e passou a apoiar áreas estratégicas, como a saúde pública, permitindo identificar padrões, tendências e fatores de risco. Ferramentas de *Business Intelligence* (BI) potencializam essa análise ao possibilitar a coleta, organização e interpretação de grandes volumes de informações de forma rápida e eficiente, favorecendo decisões baseadas em evidências.

No Brasil, Ministério da Saúde (2025), por meio do Departamento de Doenças de Condições Crônicas e Infecções Sexualmente Transmissíveis (DCCI), disponibiliza painéis interativos com indicadores e dados básicos sobre HIV/aids, sífilis e hepatites virais, abrangendo todos os municípios brasileiros. As informações são provenientes de sistemas oficiais, como o Sistema de Informação de Agravos de Notificação (SINAN), o Sistema de Controle de Exames Laboratoriais (SISCEL), o Sistema de Controle Logístico de Medicamentos (SICLOM), o Sistema de Informações sobre Mortalidade (SIM) e os dados populacionais do Instituto Brasileiro de Geografia e Estatística (IBGE).

Esses painéis têm como objetivo oferecer subsídios qualificados para a análise epidemiológica e a tomada de decisões nas esferas municipal, estadual e federal, muitas vezes as informações disponibilizadas não se encontram de forma estruturada ou suficientemente detalhada para permitir análises mais aprofundadas. Ainda assim, a iniciativa reforça a importância da informação como ferramenta estratégica para o planejamento e o monitoramento em saúde pública.

Este trabalho tem como objetivo analisar dados de internações hospitalares por HIV no Brasil, estruturando-os em um banco de dados relacional e desenvolvendo *dashboards* interativos no *Power BI*. A análise busca identificar variações temporais, padrões de distribuição e áreas de maior concentração de casos, oferecendo subsídios visuais e analíticos que possam apoiar gestores e profissionais de saúde na definição de ações de prevenção e controle da doença.

1.1 O Problema e sua Importância

O HIV representa, há décadas, um grande desafio para a saúde pública no Brasil e no mundo. Apesar dos avanços no diagnóstico e tratamento, o HIV ainda impacta de maneira

significativa a população, especialmente em comunidades vulneráveis e em regiões com menor acesso aos serviços de saúde.

Segundo a UNAIDS (2025), em 2018, havia cerca de 900 mil pessoas vivendo com HIV no Brasil. Dessas, 66% tinham acesso ao tratamento.

Ainda segundo as estimativas, 53 mil pessoas foram infectadas pelo HIV em 2018. Os dados indicam um aumento de 21% em comparação a 2010.

Dessa forma, conhecer e interpretar corretamente as informações epidemiológicas associadas à doença torna-se uma tarefa crucial para o planejamento e para a execução de políticas públicas eficazes.

Atualmente, o Sistema Único de Saúde (SUS) e demais instituições responsáveis por consolidar dados de saúde no país disponibilizam informações sobre casos registrados de HIV, abrangendo aspectos como idade, gênero, local de residência e ano do diagnóstico. No entanto, esse volume de dados, quando não tratado e estruturado de maneira clara, torna-se de difícil compreensão e pouco aproveitado para a tomada de decisões estratégicas.

Diante desse cenário, o desafio consiste em transformar dados brutos provenientes de bases públicas do DATASUS, especificamente do Sistema de Informações Hospitalares (SIH/SUS), em informações organizadas e significativas, por meio de tratamento, estruturação e integração em um banco de dados, e uma apresentação gráfica interativa no *Power BI*. Dessa maneira, profissionais de saúde poderão analisar padrões e tendências epidemiológicas com mais clareza, facilitando a identificação de áreas prioritárias para intervenção e o planejamento de estratégias mais eficazes para o controle e a prevenção do HIV no país. Ao contrário dos painéis já disponibilizados pelo Ministério da Saúde, que apresentam dados consolidados de forma ampla e com recursos limitados de exploração, este trabalho busca preencher a lacuna de análises mais detalhadas e interativas a partir dos dados do SIH/SUS. Por meio da organização das informações em um banco de dados relacional e da criação de visualizações no *Power BI*, é possível aplicar filtros, cruzar variáveis e identificar tendências de forma mais clara e aprofundada. Esse diferencial favorece a detecção de padrões relevantes e oferece uma base consistente para estudos e avaliações epidemiológicas sobre o HIV no Brasil.

1.2 Objetivo Geral

Desenvolver uma solução para consolidar e estruturar dados públicos de morbidade hospitalar por HIV do sistema DATASUS, organizando-os em um banco de dados relacional e apresentando-os em *dashboards* interativos no *Power BI*, a fim de facilitar a análise e interpretação de indicadores como internações, óbitos e taxas de mortalidade.

1.3 Objetivos Específicos

Para alcançar esse objetivo geral, o trabalho foi realizado considerando os seguintes objetivos específicos:

a) Levantar requisitos para a elaboração do *dashboard*.

b) Coletar dados públicos relacionados à morbidade hospitalar por HIV no sistema DATASUS, com foco em internações, óbitos e taxas de mortalidade.

c) Preparar os dados por meio de processos de limpeza, padronização e organização lógica, assegurando sua qualidade e consistência para posterior análise.

d) Estruturar e armazenar os dados tratados em um banco de dados relacional, possibilitando consultas eficientes e seguras.

e) Desenvolver *dashboards* e gráficos interativos no *Power BI* para visualização clara e acessível das informações.

2 Revisão de Literatura

Esta seção apresenta os fundamentos teóricos e os estudos que embasam o desenvolvimento do trabalho, assim como estudos e trabalhos relacionados que ajudam a contextualizar o tema e a abordagem utilizada.

2.1 Análise de Dados em Morbidade Hospitalar e HIV

A morbidade hospitalar está relacionada às doenças ou condições que resultam em internações, refletindo diretamente a carga de doenças que demandam atenção intensiva do sistema de saúde. No contexto da infecção pelo HIV, esse indicador permite identificar complicações associadas, como infecções oportunistas, e avaliar a efetividade das políticas de prevenção e tratamento.

Segundo o Boletim Epidemiológico de HIV e Aids, número especial de dezembro de 2024, mais de 540 mil casos de infecção por HIV foram notificados no Brasil entre 2007 e 2024, com aumento expressivo entre homens jovens, especialmente aqueles que fazem sexo com homens. Em 2023, os casos cresceram 4,5% em relação ao ano anterior, sendo que mais da metade das notificações ocorreram entre pessoas negras. Além disso, a AIDS já acumula mais de 1,1 milhão de casos desde 1980, com mais de 10 mil óbitos registrados somente em 2023 (Ministério da Saúde, 2024).

Diante desse cenário, o HIV continua exigindo respostas estratégicas sustentadas por dados epidemiológicos consistentes, especialmente no que diz respeito à morbidade hospitalar. A identificação de padrões de internação relacionados ao HIV possibilita uma melhor compreensão dos desdobramentos clínicos da infecção, contribuindo para a alocação de recursos e o aprimoramento das estratégias de cuidado.

A análise de dados hospitalares relacionados ao HIV depende de processos estruturados de coleta, tratamento e interpretação da informação. Bases públicas, como o DATASUS, fornecem dados valiosos para essa finalidade, permitindo a aplicação de técnicas de análise estatística e visualização de informações por meio de ferramentas de *Business Intelligence*, como o *Power BI*. Tais abordagens facilitam o monitoramento das internações por HIV, a avaliação do impacto das políticas públicas e o desenvolvimento de ações mais direcionadas.

2.2 Trabalhos relacionados

Apesar de não terem sido identificados estudos que unam diretamente a construção de *dashboards* interativos com foco específico na visualização de dados da saúde pública, há pesquisas que exploram temas complementares e que servem de referência para o presente projeto. Esses trabalhos abordam boas práticas de design, interatividade e usabilidade, bem como a aplicação de ferramentas de análise visual em diferentes contextos.

O estudo de Hanna (2025), concentra-se na análise e visualização de dados educacionais de um campus universitário, com o intuito de apoiar a tomada de decisão por parte dos gestores. A pesquisa foi motivada pelo grande volume de dados disponíveis, que, quando bem explorados, podem gerar informações valiosas para a comunidade acadêmica. Para o tratamento dos dados, foi utilizada a metodologia *Knowledge Discovery in Databases* (KDD), que se trata de um processo que envolve a seleção, transformação, mineração e interpretação de dados com o objetivo de extrair conhecimento útil. As visualizações foram construídas no *Power BI*. Como resultado, foram criados sete *dashboards* interativos, abordando aspectos como perfil demográfico, admissões, evasão, colação de grau, desempenho dos alunos, das disciplinas e dos departamentos.

Já o trabalho desenvolvido por Lima (2025) teve como objetivo criar um *dashboard* interativo para apoiar a gestão do curso de Engenharia de Controle e Automação de Processos da UFBA. Utilizando a linguagem *Python* para tratamento dos dados e o *Power BI* para a visualização, o painel permite acompanhar indicadores como taxa de aprovação e reprovação, relação entre disciplinas e seus pré-requisitos, além do desempenho individual dos alunos. A ferramenta também auxilia no planejamento acadêmico, oferecendo sugestões de disciplinas com base nos históricos dos estudantes. O *dashboard* segue boas práticas de usabilidade e pode ser atualizado semestralmente, agregando valor às análises e às decisões do colegiado.

No campo da saúde pública, destaca-se o trabalho de Felício (2024), que utilizou dados do Sistema de Informações Hospitalares do SUS (SIH/SUS) para analisar as internações e causas de óbito ao longo de dez anos em um município de Minas Gerais. O autor descreve o perfil epidemiológico local, identifica as principais causas de morbidade e mortalidade hospitalar e compara os indicadores municipais com as médias estaduais. Os resultados evidenciaram discrepâncias importantes e reforçam a utilidade dos dados do DATASUS para orientar políticas públicas baseadas em evidências.

A proposta deste trabalho se inspira nessas duas vertentes. Assim como nos trabalhos de Hanna (2025) e Lima (2025), utiliza-se o Power BI para construção de *dashboards* interativos, priorizando a clareza visual e a organização das informações. Ao mesmo tempo, segue a linha de Felício (2024) ao utilizar dados provenientes do DATASUS, demonstrando sua relevância e aplicabilidade. No entanto, diferentemente de Felício (2024), este projeto

não realiza análises estatísticas ou comparativas, seu foco está na extração, limpeza, padronização e visualização dos dados. O objetivo é facilitar o acesso às informações públicas de saúde por meio de uma interface visual acessível, que contribua para a compreensão dos dados por parte de gestores, pesquisadores e demais interessados.

2.3 Levantamento e Análise de Requisitos

A análise de requisitos é uma fase crucial na engenharia de software, conforme Pressman e Maxim (2021a) descrevem. Durante a definição dos requisitos, é fundamental revisar cuidadosamente o que foi elaborado para evitar erros, informações incompletas ou ambíguas. Os requisitos precisam estar alinhados aos objetivos do projeto e descritos com o grau de detalhamento compatível com a etapa de desenvolvimento, por exemplo, de forma mais geral na fase de concepção e com especificações minuciosas nas etapas de implementação e testes.

É importante verificar se cada requisito é realmente necessário, se pode ser implementado com os recursos disponíveis e se poderá ser testado posteriormente. Também é essencial que cada um deles tenha uma origem bem definida, pois essa análise ajudará a garantir que os requisitos estejam bem estruturados, sem conflitos entre si, e que sirvam como base sólida para as próximas etapas do desenvolvimento. Um bom modelo de requisitos deve representar com clareza as informações, funções e comportamentos esperados do sistema e, sempre que possível, utilizando padrões que facilitem a compreensão e a validação por todas as partes envolvidas.

2.4 Requisitos Funcionais

De acordo com Morais e Zanin (2020), os requisitos funcionais descrevem o que o sistema deve fazer. Ou seja, eles definem os serviços e comportamentos esperados da solução, como ela deve reagir a determinadas entradas e o que ela deve entregar em resposta. Esses requisitos são essenciais para garantir que o sistema atenda às necessidades específicas de quem vai usá-lo.

Na prática, um requisito funcional não apenas diz o que o sistema precisa realizar, mas também especifica de que forma isso deve ser feito, considerando o contexto e os objetivos do projeto. Além de definir o que deve ser feito, os requisitos funcionais precisam ser claros, realistas e completos, evitando ambiguidades e lacunas. Eles formam a base para a implementação das funcionalidades e ajudam a garantir que o sistema se comporte conforme o esperado, respeitando os limites e objetivos definidos desde o início.

2.5 Requisitos Não Funcionais

Os requisitos não funcionais de um sistema tratam-se das condições e restrições que influenciam como o sistema irá operar como um todo. Essa distinção é amplamente discutida por Sommerville (2025a), que apresenta uma visão aprofundada sobre o papel e a importância desses requisitos no processo de desenvolvimento.

Os requisitos não funcionais não estão diretamente ligados às funcionalidades específicas do sistema, mas sim às suas características globais, como desempenho, confiabilidade, usabilidade, segurança e conformidade com normas legais e organizacionais. Eles podem definir o tempo máximo de resposta para uma consulta em banco de dados, os níveis de disponibilidade de uma aplicação ou os padrões de proteção de dados exigidos por legislações como a LGPD.

Esses requisitos podem surgir por diversos fatores, incluindo necessidades dos usuários, limitações técnicas, políticas internas da organização ou exigências externas como leis e regulamentações. Sommerville (2025a) observa que, muitas vezes, um único requisito não funcional pode impactar significativamente a arquitetura do sistema ou dar origem a diversos requisitos funcionais, como ocorre em demandas relacionadas à segurança, que exigem autenticação, controle de acesso e registro de atividades.

Sommerville (2025a) ainda destaca que o não atendimento a um requisito não funcional pode comprometer o sistema por completo, mesmo que suas funcionalidades estejam corretas. Um sistema de dados que atenda aos requisitos funcionais de análise e visualização, mas que não seja confiável ou seguro, pode ser considerado inadequado para uso, especialmente em contextos sensíveis como o da saúde.

2.6 Tecnologias Envolvidas

Nesta seção serão apresentadas as tecnologias que foram utilizadas para o desenvolvimento desse trabalho.

2.6.1 Jupyper Notebook

Jupyter Notebook é uma ferramenta interativa amplamente utilizada por cientistas, analistas de dados e estudantes, que permite combinar texto, código, gráficos e visualizações em um mesmo ambiente (Project Jupyter, 2025). A plataforma, acessada via navegador e de código aberto, possibilita a criação e o compartilhamento de documentos que integram código executável, equações, visualizações e recursos multimídia, sendo muito útil em tarefas como análise exploratória de dados (EDA), tratamento e transformação de dados, modelagem estatística e aplicações de aprendizado de máquina e profundo (Databricks, 2025).

Um dos principais componentes do *Jupyter* é o *kernel*, responsável por executar o código em tempo real e manter os dados em memória durante a sessão. Isso permite ao usuário testar, ajustar e reexecutar trechos de código de forma contínua, facilitando uma abordagem incremental e dinâmica no desenvolvimento das análises.

2.6.2 Python

Python é uma linguagem de programação versátil e de fácil leitura, amplamente utilizada em aplicações *web*, desenvolvimento de *software*, automação de tarefas, ciência de dados e aprendizado de máquina (Amazon Web Services, 2025). Sua sintaxe simples e próxima da linguagem humana facilita o aprendizado, atraindo tanto profissionais da área de tecnologia quanto pessoas de outras áreas que precisam lidar com programação ou análise de dados (CARVALHO, 2025).

Devido a essas características, o *Python* se tornou uma das linguagens mais populares do mundo. Ele permite realizar desde tarefas básicas, como limpeza de dados e criação de visualizações, até processos mais complexos, como modelagem estatística, seleção de variáveis e rotulagem de dados. Ferramentas gráficas, como tabelas, gráficos de linhas, barras, histogramas e gráficos de pizza, também são facilmente produzidas com bibliotecas da linguagem, tornando os resultados mais acessíveis e compreensíveis (Amazon Web Services, 2025).

2.6.3 Pandas

Almeida (2023a), Pandas é uma biblioteca de código aberto construída sobre *Python*, que oferece estruturas robustas e flexíveis para trabalhar com dados rotulados, como tabelas e séries temporais. O nome "Pandas"deriva de *Panel Data*, um conceito da econometria, e não do animal, o nome *Python* faz referência à trupe de comédia *Monty Python*. A biblioteca é amplamente utilizada em tarefas como limpeza de dados, análise exploratória (EDA), integração com bancos relacionais, suporte a *machine learning*, visualização de dados, *web scraping*, entre outras. Também possui integração com diversas bibliotecas populares da Ciência de Dados.

2.6.4 PostgreeSQL

O PostgreSQL é um banco de dados relacional de código aberto, conhecido por ser confiável, seguro e robusto. Ele usa SQL e oferece recursos avançados para lidar com grandes volumes de dados. É compatível com todos os sistemas operacionais, tem mais de 35 anos de desenvolvimento e pode ser expandido com ferramentas como o *PostGIS* para dados geoespaciais. Por isso, é amplamente usado por pessoas e empresas no mundo todo (PostgreSQL Global Development Group, 2025).

Ao contrário de outros sistemas de gerenciamento de banco de dados relacionais (RDBMS), o *PostgreSQL* oferece suporte tanto a dados relacionais quanto a tipos de dados mais flexíveis, como JSON, *arrays* e dados geoespaciais. Também é compatível com os padrões ANSI SQL e suporta diversas linguagens de programação, como *Python*, Java, C/C++ e PHP. Essa compatibilidade e flexibilidade contribuem para sua ampla adoção e estabilidade no mercado. Com mais de 35 anos de desenvolvimento contínuo, o *PostgreSQL* consolidou-se como uma das soluções de banco de dados mais confiáveis e tecnicamente robustas disponíveis atualmente (IBM, 2025).

2.6.5 Power BI

De acordo com Almeida (2023b) e Microsoft (2025), o *Power BI* é uma coleção de serviços de *software*, aplicativos e conectores que fornece funcionalidades de visualização de dados e geração de relatórios interativos, sendo amplamente utilizado no campo de *Business Intelligence* (BI). Ele foi desenvolvido para atender à crescente demanda de organizações que adotam uma abordagem *data-driven*, ou seja, que baseiam suas decisões em dados e evidências concretas para aumentar a eficiência e a precisão dos seus processos decisórios. O *Power BI* permite transformar diversas fontes de dados, como planilhas do *Excel*, bancos de dados *SQL Server* e *data warehouses* locais ou em nuvem, por meio de painéis interativos, filtros dinâmicos e integração com múltiplas fontes, em relatórios acessíveis e visualmente atrativos, facilitando tanto a análise quanto o compartilhamento de *insights*.

2.7 Kanban

Pressman e Maxim (2021b) descrevem o *Kanban* como uma abordagem visual e incremental de gerenciamento de fluxo de trabalho, usada dentro das metodologias ágeis. O *Kanban* tem como foco principal o controle do trabalho em progresso e a entrega contínua, promovendo maior transparência, produtividade e adaptação. O quadro *Kanban* é dividido em colunas que representam os estágios do fluxo de trabalho, permitindo a visualização do status de cada tarefa, o que facilita o monitoramento visual do progresso, a identificação de gargalos e a melhoria contínua o processo. Seu uso contribui para uma gestão mais eficiente, com foco nas tarefas prioritárias e redução de desperdícios. Os autores ainda destacam que o *Kanban* não exige grandes mudanças estruturais e pode ser implementado de forma gradual em qualquer projeto. A Figura 1 ilustra o quadro *Kanban*.

Figura 1 – Quadro *Kanban*.

Fonte: (SEBRAE, 2023)

3 Metodologia

A metodologia deste trabalho foi estruturada em quatro etapas principais: obtenção dos dados, preparação e limpeza, armazenamento estruturado, e visualização interativa com *dashboards*.

3.1 Processo de Desenvolvimento

Para o gerenciamento e organização das tarefas ao longo do desenvolvimento deste trabalho, adotou-se a metodologia ágil baseada no *Kanban*, por ser uma abordagem simples, prática e facilmente adaptável a projetos individuais. A escolha se deu também pelo fato de o autor já ter familiaridade com essa ferramenta, o que contribuiu para sua adoção sem a necessidade de uma curva de aprendizado ou mudanças estruturais no processo.

3.1.1 Aplicação do *Kanban* na Organização do Projeto

A adoção do *Kanban* neste projeto contribuiu diretamente para a organização e o acompanhamento das etapas envolvidas na análise dos dados de morbidade hospitalar. Por meio da divisão visual do fluxo de trabalho em colunas como "A fazer", "Em andamento" e "Concluído", foi possível acompanhar o progresso das atividades de forma clara e contínua. Essa visualização facilitou a identificação de gargalos, o replanejamento de tarefas e a priorização de etapas mais relevantes, especialmente durante a coleta, o tratamento dos dados e a construção dos *dashboards*. Essa abordagem foi fundamental para manter a produtividade, cumprir os prazos e conduzir o projeto de forma estruturada e eficiente.

3.2 Levantamento de Requisitos

O levantamento de requisitos tem como objetivo identificar as funcionalidades essenciais do sistema e as características desejáveis para seu funcionamento, garantindo que a solução atenda às necessidades de análise e visualização de forma clara, eficiente e acessível.

A seguir, serão apresentados os requisitos funcionais e não funcionais que orientaram as etapas do projeto, desde a coleta e o tratamento dos dados até sua apresentação por meio de *dashboards* interativos no *Power BI*.

3.2.1 Requisitos Funcionais

Os requisitos funcionais deste trabalho foram definidos com base na compreensão do problema e nos objetivos traçados ao longo do projeto. Por se tratar de um desenvolvimento individual, sem a participação direta de usuários externos, esses requisitos foram pensados pelo próprio autor, levando em conta o contexto da morbidade hospitalar por HIV.

A definição dos requisitos buscou seguir princípios como clareza, viabilidade e alinhamento com os resultados esperados, garantindo que cada funcionalidade contribuísse para o propósito final da solução.

Os requisitos funcionais para o desenvolvimento foram:

- RF01 – O sistema deve exibir um mapa interativo com a divisão por regiões geográficas
- RF02 – Ao selecionar uma região no mapa, o sistema deve filtrar os dados dos gráficos e indicadores para exibir apenas os valores referentes àquela região.
- RF03 – O sistema deve exibir a quantidade total de internações e óbitos referentes à região selecionada
- RF04 – O sistema deve exibir um gráfico de barras com dados mensais de internações e óbitos.
- RF05 – O sistema deve exibir o número total de óbitos e internações do mês selecionado.
- RF06 – O sistema deve exibir cartões de resumo com a quantidade total de interações e óbitos para o período e região selecionados.

3.2.2 Requisitos Não Funcionais

De acordo com Morais e Zanin (2020), enquanto os requisitos funcionais descrevem o que o sistema deve fazer, os requisitos não funcionais definem como o sistema deve se comportar ou quais características ele deve apresentar. Os requisitos não funcionais são atributos de qualidade, desempenho, segurança ou restrições gerais que impactam diretamente na usabilidade, eficiência e confiabilidade da solução.

Em outras palavras, eles não estão ligados a funcionalidades específicas, mas sim à forma como essas funcionalidades devem operar. Por exemplo, no contexto deste trabalho, um requisito não funcional seria garantir que os *dashboards* gerados no *Power BI* sejam de fácil compreensão e tenham tempo de carregamento adequado para o usuário. Ou ainda, que as ferramentas utilizadas sejam acessíveis, preferencialmente gratuitas, e compatíveis com a infraestrutura disponível.

Esses requisitos são fundamentais para garantir uma boa experiência do usuário e para assegurar que o sistema atenda padrões mínimos de qualidade. Eles também ajudam a limitar o escopo do projeto e a orientar escolhas tecnológicas, metodológicas e operacionais.

- RNF01 – O *dashboard* deve carregar completamente em até 15 segundos após a abertura, considerando o conjunto de dados carregado na memória.
- RNF02 – Ao aplicar um filtro (região ou mês), os gráficos e indicadores devem ser atualizados em no máximo 8 segundos.
- RNF03 – As legendas, rótulos e títulos devem ser claros e utilizar linguagem acessível para usuários não técnicos.
- RNF04 - O uso de cores deve seguir um padrão visual coerente, garantindo legibilidade e diferenciação entre internações e óbitos.

3.3 Obtenção dos dados

Os dados utilizados neste trabalho foram extraídos manualmente do sistema DATASUS, especificamente do banco de dados de internações hospitalares (SIH/SUS). Foram considerados arquivos mensais no formato CSV, um tipo de arquivo baseado em texto simples utilizado para armazenar dados tabulares, no qual cada linha representa um registro e os campos são separados por vírgulas ou outro caractere delimitador, como o ponto e vírgula. Os arquivos, referentes ao período de abril de 2024 a março de 2025, foram obtidos diretamente do site oficial do Departamento de Informática do SUS. Cada arquivo representa um mês de registro de internações por causas diversas, com foco específico nas ocorrências relacionadas à infecção pelo HIV.

3.4 Tratamento e preparação dos dados

Após a coleta dos arquivos, foi realizado um processo de tratamento e padronização utilizando a linguagem Python, com a biblioteca Pandas. As principais etapas executadas incluem:

- Leitura automatizada dos arquivos CSV e extração de dados com codificação adequada;
- Remoção de registros inválidos, definidos como linhas contendo valores nulos em todos os campos ou registros com conteúdo incompleto em colunas essenciais (como ausência de nome da unidade da federação ou valores numéricos ausentes para internações e óbitos);

- Eliminação de linhas descritivas, como rodapés com termos não estruturados como “Fonte”, “Notas”e “Total”, que não compõem registros observacionais e poderiam distorcer as análises;
- Padronização dos nomes das colunas, removendo acentos, espaços e substituindo caracteres especiais para facilitar a manipulação dos dados;
- Criação da coluna mês origem, a partir do nome do arquivo, com o objetivo de rastrear o período de cada conjunto de dados;
- Normalização dos campos territoriais, como nome do estado ou região, com remoção de símbolos de pontuação e caracteres de espaço duplicados;
- Criação da variável tipo_local, classificando cada linha como pertencente a uma “Região”ou a um “Estado”, com base no conteúdo textual.

Além das etapas acima, foi realizada a seleção das colunas relevantes para a análise, com base em critérios de utilidade analítica e coerência com os objetivos do trabalho. Foram mantidas apenas as colunas:

- Regiao_Unidade_da_Federacao: identifica o território analisado, seja estado ou região agregada;
- Internacoes: indica o número absoluto de internações hospitalares, sendo um indicador direto de morbidade;
- Obitos: representa a quantidade de óbitos associados às internações, importante para avaliar a gravidade dos casos;
- Taxa_mortalidade: indicador calculado que relaciona óbitos com internações, permitindo comparações proporcionais entre locais e períodos;
- mes_origem: identifica o mês e o ano de referência dos dados, essencial para a análise temporal;
- tipo_local: variável derivada, cuja inclusão visa possibilitar cruzamentos e segmentações entre níveis geográficos distintos.

Essa filtragem teve como foco manter apenas os campos que efetivamente contribuem para os objetivos da pesquisa, reduzindo ruído informacional e facilitando a interpretação dos resultados.

3.5 Armazenamento dos dados

Os dados tratados foram consolidados em um único *DataFrame* utilizando a biblioteca pandas, sendo posteriormente estruturados com colunas padronizadas, tipos de dados coerentes (ex.: inteiros para contagens, *floats* para taxas) e formato tabular compatível com sistemas de gerenciamento de banco de dados relacionais.

Essa normalização do *DataFrame* permitiu a criação de uma base coesa, com colunas consistentes e sem ambiguidade semântica, facilitando sua utilização em ferramentas de *Business Intelligence* (BI), como *Power BI* ou *Tableau*. Durante os testes, o armazenamento foi mantido localmente no formato .csv, possibilitando rápida visualização e análise exploratória.

No entanto, o modelo está preparado para exportação para bancos relacionais, como PostgreSQL. Essa exportação pode ser realizada por meio da biblioteca *SQLAlchemy*, que permite a conexão entre o *Python* e diversos SGBDs, viabilizando a criação automática de tabelas e inserção dos dados diretamente a partir do *DataFrame*. Essa abordagem garante escalabilidade e integração futura com ambientes de produção, caso necessário.

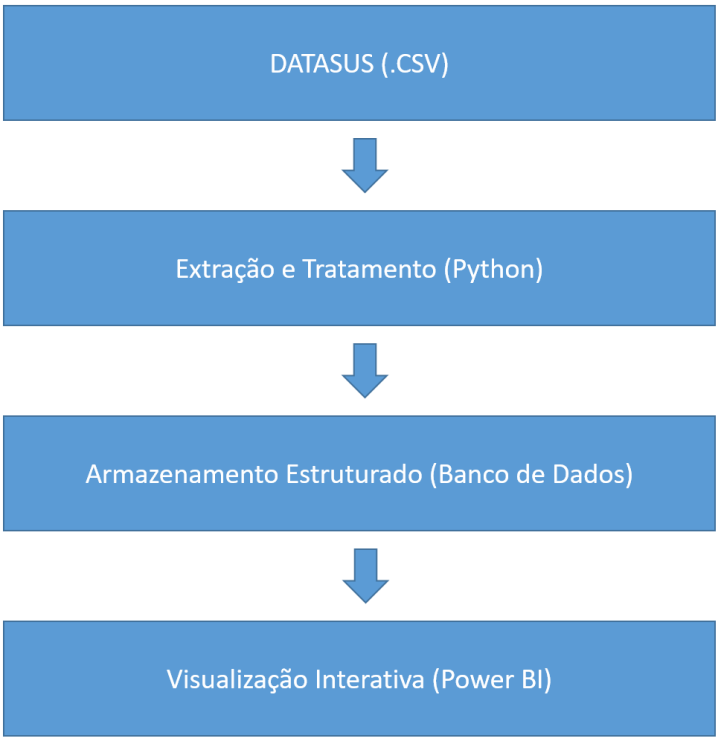
3.6 Visualização com *Power BI*

Com os dados estruturados, a próxima etapa consistiu na visualização interativa utilizando o *Microsoft Power BI*, com foco na construção de *dashboards* interativos que facilitam a interpretação dos dados de internações hospitalares por HIV. Os painéis desenvolvidos permitem a análise temporal por mês, geográfica por estado/região e comparativa internações, óbitos e taxas de mortalidade. A navegação foi estruturada de modo a permitir consultar os dados de forma clara, com filtros e segmentações visuais.

3.7 Arquitetura

A arquitetura do projeto foi estruturada como um *pipeline* de dados dividido em quatro camadas principais: Fonte de Dados, Extração e Tratamento, Armazenamento Estruturado e Visualização Interativa, conforme ilustrado na Figura 2. Cada etapa possui um papel específico e tecnologias adequadas ao seu propósito.

Figura 2 – Arquitetura.



Fonte: Do Autor

3.7.1 Fonte de Dados (*Data Source*):

A origem dos dados é o DATASUS, mais especificamente os conjuntos disponibilizados via TABNET, como o Sistema de Informações Hospitalares (SIH/SUS). Os dados são extraídos em formato CSV, com codificação Latin-1, contendo colunas como UF, Município, CID, Internações, Óbitos, entre outras. As planilhas são segmentadas por período e região.

- Formato: CSV, separado por ";"
- Atualização: manual, com periodicidade mensal ou trimestral, conforme disponibilidade do DATASUS
- Volume Médio: entre 5 KB e 10 KB por arquivo

3.7.2 Camada de Extração e Tratamento (*Python*)

- A extração de dados é feita de forma manual a partir do DATASUS e, posteriormente, os arquivos CSV são carregados no ambiente *Jupyter Notebook* e tratados com a biblioteca *Python* Pandas. Em seguida, foi realizada a remoção de inconsistências, a padronização das colunas e a unificação dos arquivos por período e região.

- Etapas do tratamento:
  - Padronização dos nomes das colunas e remoção de acentos
  - Conversão de tipos (str → *int/float*, *date*)
  - Unificação de múltiplos arquivos em um único *dataframe* por ano/região
  - Remoção de registros com campos nulos ou inconsistentes (ex: UF inválida)
- *Scripts* reutilizáveis são criados para cada etapa do ETL.
- Saída: um *DataFrame* limpo pronto para inserção no banco de dados

3.7.3 Armazenamento Estruturado (Banco de Dados)

Os dados tratados são inseridos em uma base de dados relacional PostgreSQL, hospedada localmente, estruturada para facilitar consultas e visualizações no *Power BI*.

- Estrutura: Tabela única contendo os dados tratados, com índices criados em campos-chave como ano, uf, cid e tipo_internacao para otimizar as consultas.
- Tabela principal:
  - morbidade_hospitalar
- Tipos de dados utilizados:
  - *VARCHAR* para nomes e descrições
  - *INTEGER* para contagens
  - *DATE* para datas
  - *FLOAT* para taxas

3.7.4 Visualização Interativa (*Power BI*)

O *Power BI* se conecta diretamente ao PostgreSQL por meio de seu conector nativo, permitindo a atualização automática dos relatórios. Os *dashboards* interativos apresentam gráficos e filtros por região, estado e outros indicadores epidemiológicos, com foco em internações, óbitos e taxa de mortalidade por HIV.

- *Dashboards*:
  - Filtros por UF e Região
  - Gráficos de barra de internações e óbitos
  - Mapas com densidade de casos por região

- Indicadores calculados:
  - Taxa de mortalidade (óbitos/internações)
  - Variação percentual entre meses

3.8 Delimitação do Projeto

A abordagem deste projeto está limitada às etapas de extração, tratamento, armazenamento e visualização dos dados, utilizando ferramentas como *Jupyter*, *Power BI* e a linguagem *Python* para a limpeza e organização das informações.

Foram utilizados exclusivamente dados obtidos no sistema DATASUS, mais especificamente do Sistema de Informações Hospitalares (SIH/SUS), em formato CSV, referentes às internações hospitalares por causas relacionadas ao HIV. Trata-se de dados públicos, abertos e anonimizados, não envolvendo informações sigilosas, identificáveis ou sensíveis. O objetivo é fornecer uma visão geral, acessível e interativa, que possa apoiar a tomada de decisões em saúde pública.

3.9 Modelagem Banco de Dados

De acordo com Silberschatz (2020) a representação visual da estrutura de um banco de dados é uma etapa importante no planejamento de qualquer sistema que use dados. Ela ajuda a entender como os dados estão organizados e facilita a comunicação entre quem conhece a parte técnica e quem entende do conteúdo, mesmo que não saiba modelar dados.

A base de dados criada para este projeto consiste em uma única tabela, estruturada para armazenar informações de morbidade hospitalar relacionadas ao HIV, extraídas do DATASUS. As colunas foram definidas com base nas variáveis disponíveis, permitindo segmentações por região, número de internações, óbitos, taxa de mortalidade, entre outras.

Foi criada uma única tabela para reunir de forma organizada as informações necessárias para a análise, de acordo com os objetivos do trabalho. Essa estrutura facilita a visualização e a compreensão dos dados de morbidade hospitalar relacionados ao HIV.

Vale destacar que, em versões futuras do projeto, especialmente se houver necessidade de ampliar o sistema ou torná-lo mais robusto, pode ser interessante adotar uma modelagem com mais tabelas e seguir regras de normalização, garantindo uma estrutura mais adequada para sistemas maiores.

3.10 Obtenção e Preparação dos Dados

Nesta seção, descrevem-se os processos realizados para obtenção, extração e pré-processamento dos dados, além da estruturação do banco de dados utilizado no projeto. Também será abordada a integração com o *Power BI* e a construção do *dashboard* interativo desenvolvido para visualização das informações.

3.11 Metodologia do Desenvolvimento

Após o levantamento de requisitos e a escolha das tecnologias a serem utilizadas, o processo de desenvolvimento foi organizado com o auxílio da metodologia ágil *Kanban*, que permitiu acompanhar o andamento das tarefas de forma visual e prática. O quadro *Kanban* foi dividido em quatro principais etapas: extração dos dados, tratamento dos dados, armazenamento em banco de dados e criação do *dashboard*.

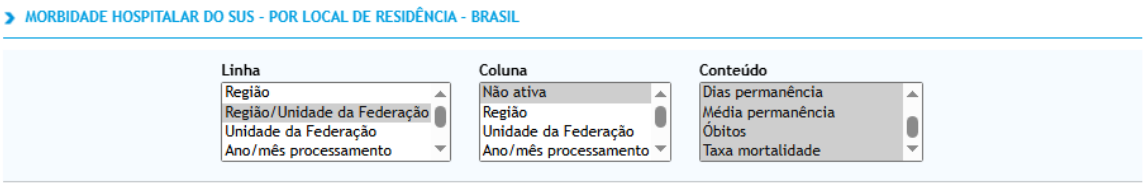
O uso do *Kanban* foi essencial para a organização do projeto, permitindo manter o foco nas prioridades, acompanhar o progresso das atividades e realizar ajustes sempre que necessário, contribuindo para a entrega dentro do prazo planejado.

3.12 Extração Manual de Dados do DATASUS

A extração dos dados foi realizada manualmente por meio do site do DATASUS, utilizando a ferramenta TABNET, que permite a tabulação personalizada de informações de saúde pública. Por meio dela, foi possível acessar dados de morbidade hospitalar do sistema SIH/SUS, aplicando filtros como o local de residência selecionando "Brasil por Região e Unidade da Federação" para obter os dados desejados. Os dados foram exportados mês a mês, no período de março de 2024 a março de 2025, em arquivos no formato CSV.

A Figura 3 apresenta a interface do DATASUS com os filtros aplicados para a extração dos dados de morbidade hospitalar por local de residência, com destaque para a seleção da linha "Região/Unidade da Federação", a configuração da coluna como "Não ativa", e a escolha dos indicadores na aba Conteúdo, como dias de permanência, média de permanência, óbitos e taxa de mortalidade. As demais extrações estão descritas no Apêndice A.

Figura 3 – Morbidade Hospitalar do SUS – Por local de Residência - Brasil



Fonte: DATASUS

3.13 Pré-processamento e Limpeza dos dados

Com os dados extraídos, iniciou-se a etapa de pré-processamento e limpeza, realizada no ambiente *Jupyter Notebook*, utilizando a linguagem de programação *Python*. Para isso, foram empregadas bibliotecas como o Pandas, que permitiram a leitura, organização e padronização das informações de forma eficiente.

As principais ações realizadas incluíram:

- Conversão de tipos de dados para garantir consistência;
- Padronização dos nomes dos estados e meses;
- Cálculo da taxa de mortalidade (óbitos/internações);
- Criação de colunas auxiliares para ordenação cronológica.

Na Listagem 3.1 é apresentado o código utilizado no *Jupyter Notebook*. Essa estrutura automatiza a leitura, limpeza e padronização de múltiplos arquivos CSV com dados públicos de morbidade hospitalar, permitindo uma análise consolidada e eficiente dos dados no *Power BI* ou em outras ferramentas de visualização. Nos próximos tópicos, são fornecidas as definições correspondentes.

Listagem 3.1 – Código de leitura e tratamento de arquivos CSV com Pandas

```
#bibliotecas
import pandas as pd
import os
from unidecode import unidecode

# Caminho da pasta com os arquivos
pasta_arquivos = r'C:\Users\Emerson\Desktop\Jupyter'

# Lista dos nomes dos arquivos
nomes_arquivos = [
    'Abr2024.csv', 'Ago2024.csv', 'Dez2024.csv', 'Fev2025.csv', 'Jan2025.csv',
    'Jul2024.csv', 'Set2024.csv', 'Out2024.csv', 'Nov2024.csv', 'Mar2025.csv',
    'Mar2024.csv', 'Mai2024.csv', 'Jun2024.csv'
]
```

```python

dfs = []

for nome_arquivo in nomes_arquivos:
    caminho_completo = os.path.join(pasta_arquivos, nome_arquivo)
    try:
        df = pd.read_csv(
            caminho_completo,
            skiprows=4,
            encoding='latin1',
            sep=';',
            on_bad_lines='skip'
        )

        # Remove linhas com valor nulo na primeira coluna
        df = df[df[df.columns[0]].notna()]

        # Remove linhas indesejadas por conteudo
        df = df[~df[df.columns[0]].str.contains("Fonte|Notas|Dados referentes|Total", na=False)]

        # Adiciona a coluna de origem do mes
        df['mes_origem'] = nome_arquivo.replace('.csv', '')

        dfs.append(df)
        print(f" Arquivo processado: {nome_arquivo}")
    except Exception as e:
        print(f" Erro ao ler {nome_arquivo}: {e}")

# Junta todos os dados
df_final = pd.concat(dfs, ignore_index=True)

# Funcao para tratar nomes de colunas
def tratar_nome_coluna(col):
    col = unidecode(col)
    col = col.replace('/', '_').replace('-', '_').replace(' ', '_')
    while '__' in col:
        col = col.replace('__', '_')  # Evita multiplos underlines
    col = col.strip('_')  # Remove _ no inicio/fim
    return col

# Aplica o tratamento aos nomes das colunas
df_final.columns = [tratar_nome_coluna(c) for c in df_final.columns]

# Nome da coluna da regiao/estado
col_territorio = df_final.columns[0]

# Remove pontos e espacos da coluna e corrige nomes
df_final[col_territorio] = df_final[col_territorio].str.lstrip('.').str.strip()

# Cria coluna tipo_local: Estado ou Regiao
df_final['tipo_local'] = df_final[col_territorio].apply(
    lambda x: 'Regiao' if isinstance(x, str) and x.startswith('Regiao') else 'Estado'
)

# Garante que os nomes das colunas usadas existem no DataFrame
```

```
colunas_desejadas = [
    'Regiao_Unidade_da_Federacao',
    'Internacoes',
    'Obitos',
    'Taxa_mortalidade',
    'mes_origem',
    'tipo_local'
]

# Filtra apenas as colunas desejadas (ignora as que nao existem)
colunas_existentes = [col for col in colunas_desejadas if col in df_final.
    columns]
df_final = df_final[colunas_existentes]
```

3.13.1 Bibliotecas Importadas

O código inicia com a importação das bibliotecas necessárias, como mostrado na Listagem 3.2.

Listagem 3.2 – Bibliotecas

```
#bibliotecas
import pandas as pd
import os
from unidecode import unidecode
```

- Pandas: biblioteca fundamental para manipulação e análise de dados em *Python*.
- Os: utilizada para manipular caminhos de arquivos e diretórios.
- *Unidecode*: empregada para remover acentuação dos nomes das colunas.

3.13.2 Configuração da Pasta e Lista de Arquivos

Define-se o caminho da pasta onde os arquivos CSV estão salvos como demonstrado na Listagem 3.3. onde é criada uma lista com os nomes desses arquivos, representando diferentes meses e anos.

Listagem 3.3 – Configuração da Pasta e Lista de Arquivos

```
# Caminho da pasta com os arquivos
pasta_arquivos = r'C:\Users\Emerson\Desktop\Jupyter'

# Lista dos nomes dos arquivos
nomes_arquivos = [
    'Abr2024.csv', 'Ago2024.csv', 'Dez2024.csv', 'Fev2025.csv', 'Jan2025.csv',
    'Jul2024.csv', 'Set2024.csv', 'Out2024.csv', 'Nov2024.csv', 'Mar2025.csv',
    'Mar2024.csv', 'Mai2024.csv', 'Jun2024.csv'
]
```

3.13.3 Leitura dos Arquivos

Os arquivos são lidos em um *loop* "*for*", com tratamento de exceções para lidar com possíveis erros de leitura, Listagem 3.4.

Listagem 3.4 – Leitura dos Arquivos

```
for nome_arquivo in nomes_arquivos:
    caminho_completo = os.path.join(pasta_arquivos, nome_arquivo)
    try:
        df = pd.read_csv(
            caminho_completo,
            skiprows=4,
            encoding='latin1',
            sep=';',
            on_bad_lines='skip'
        )

        # Remove linhas com valor nulo na primera coluna
        df = df[df[df.columns[0]].notna()]

        # Remove linhas indesejadas por conteudo
        df = df[~df[df.columns[0]].str.contains("Fonte|Notas|Dados referentes|
    Total", na=False)]

        # Adiciona a coluna de origem do mes
        df['mes_origem'] = nome_arquivo.replace('.csv', '')

        dfs.append(df)
        print(f" Arquivo processado: {nome_arquivo}")
    except Exception as e:
        print(f" Erro ao ler {nome_arquivo}: {e}")
```

Durante a leitura dos arquivos, são realizadas as seguintes ações::

- Remoção a primeira linha se contiver apenas valores nulos.
- Exclusão das linhas com conteúdo indesejados, como "Fonte", "Notas" ou "Total".
- Adição uma nova coluna "mes_origem" com o nome do arquivo (sem o CSV).

3.13.4 Junção dos Dados

Na Listagem 3.5, todos os DataFrames lidos são unidos em um único DataFrame final.

Listagem 3.5 – Junção dos Dados

```
# Junta todos os dados
df_final = pd.concat(dfs, ignore_index=True)
```

3.13.5 Padronização dos Nomes das Colunas

Para facilitar o uso posterior, os nomes das colunas são padronizados, Listagem 3.6.

Listagem 3.6 – Padronização dos Nomes das Colunas

```
# Funcao para tratar nomes de colunas
def tratar_nome_coluna(col):
    col = unidecode(col)
    col = col.replace('/', '_').replace('-', '_').replace(' ', '_')
    while '__' in col:
        col = col.replace('__', '_')  # Evita m ltiplos underlines
    col = col.strip('_')  # Remove _ no in cio/fim
    return col

# Aplica o tratamento aos nomes das colunas
df_final.columns = [tratar_nome_coluna(c) for c in df_final.columns]
```

Essa função:

- Remove acentos.
- Substitui espaços e pontuações por *underscores*.
- Elimina *underscores* duplicados ou desnecessários.

3.13.6 Normalização do Nome da Coluna de Território

Como demonstrado na Listagem 3.7. O código localiza a coluna que representa região ou estado e remove pontos ou espaços indesejados no início/fim dos valores.

Listagem 3.7 – Padronização dos Nomes das Colunas

```
# Nome da coluna da regiao/estado
col_territorio = df_final.columns[0]

# Remove pontos e espacos da coluna e corrige nomes
df_final[col_territorio] = df_final[col_territorio].str.lstrip('.').str.strip()
```

3.13.7 Classificação do Tipo Local

Na Listagem 3.8, cria-se uma nova coluna chamada "tipo_local" que classifica os registros como "Região" ou "Estado" com base no conteúdo do nome.

Listagem 3.8 – Classificação do Tipo Local

```
# Cria coluna tipo_local: Estado ou Regiao
df_final['tipo_local'] = df_final[col_territorio].apply(
    lambda x: 'Regiao' if isinstance(x, str) and x.startswith('Regi o') else '
    Estado'
)
```

3.13.8 Seleção das Colunas de Interesse

Garante-se que apenas as colunas relevantes sejam mantidas, conforme mostrado na Listagem 3.9.

Listagem 3.9 – Seleção das Colunas de Interesse

```
# Garante que os nomes das colunas usadas existem no DataFrame
colunas_desejadas = [
    'Regiao_Unidade_da_Federacao',
    'Internacoes',
    'Obitos',
    'Taxa_mortalidade',
    'mes_origem',
    'tipo_local'
]

# Filtra apenas as colunas desejadas (ignora as que nao existem)
colunas_existentes = [col for col in colunas_desejadas if col in df_final.
    columns]
df_final = df_final[colunas_existentes]
```

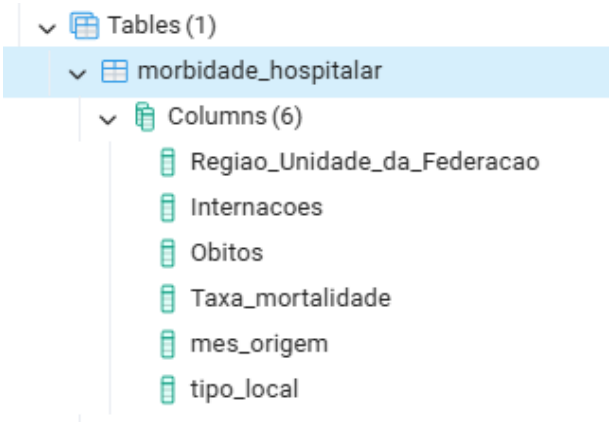
Colunas relevantes:

- Região ou Estado
- Internações
- Óbitos
- Taxa de mortalidade
- Mês de origem
- Tipo de local

3.14 Estrutura do Projeto Banco de Dados

Após o processo de limpeza e transformação, os dados tratados sobre morbidade hospitalar por HIV no Brasil foram organizados em uma tabela relacional chamada morbidade_hospitalar, armazenada em um *schema* exclusivo no banco de dados PostgreSQL. Essa estrutura simplificada facilita tanto a realização de consultas quanto a integração com ferramentas de visualização, como o *Power BI*, conforme ilustrado na Figura 4.

Figura 4 – Estrutura da Tabela Morbidade Hospitalar no PostgreSQL



Fonte: Do Autor

A tabela foi projetada com foco na simplicidade e objetividade, reunindo as informações essenciais para análise estatística e visual. Abaixo estão descritas suas colunas principais:

- Regiao_Unidade_da_Federacao: identifica se o dado corresponde a uma região (ex: Sudeste) ou unidade federativa (ex: Espírito Santo);
- Internacoes: representa o número de internações hospitalares registradas no período;
- Obitos: quantidade de óbitos decorrentes da morbidade analisada (HIV);
- Taxa_mortalidade: proporção de óbitos em relação às internações, expressa como taxa percentual;
- mes_origem: indica o mês de referência dos dados, permitindo análises temporais;
- tipo_local: classifica o nível territorial da linha (Estado ou Região), facilitando a segmentação por granularidade geográfica.

3.15 Integração e construção do *Dashboard* no *Power BI*

Na última etapa do projeto, os dados armazenados no PostgreSQL foram integrados à ferramenta de visualização *Power BI*. A conexão foi realizada diretamente com o banco de dados, permitindo a criação de um *dashboard* interativo, com navegação fluida e interface amigável.

O painel foi desenvolvido priorizando a clareza visual por meio da utilização de cores contrastantes, legendas explicativas e organização lógica dos elementos, facilitando a rápida compreensão das informações apresentadas. Para garantir a acessibilidade, foram adotados recursos como fontes legíveis em tamanho adequado, além de filtros interativos

que permitem aos usuários selecionar e explorar os dados de acordo com suas necessidades específicas.

O processo de construção do *dashboard* incluiu a escolha dos tipos de gráficos mais adequados (como gráficos de barra e mapas), a definição de cores para melhor legibilidade e a disposição dos elementos de forma a destacar as informações mais relevantes.

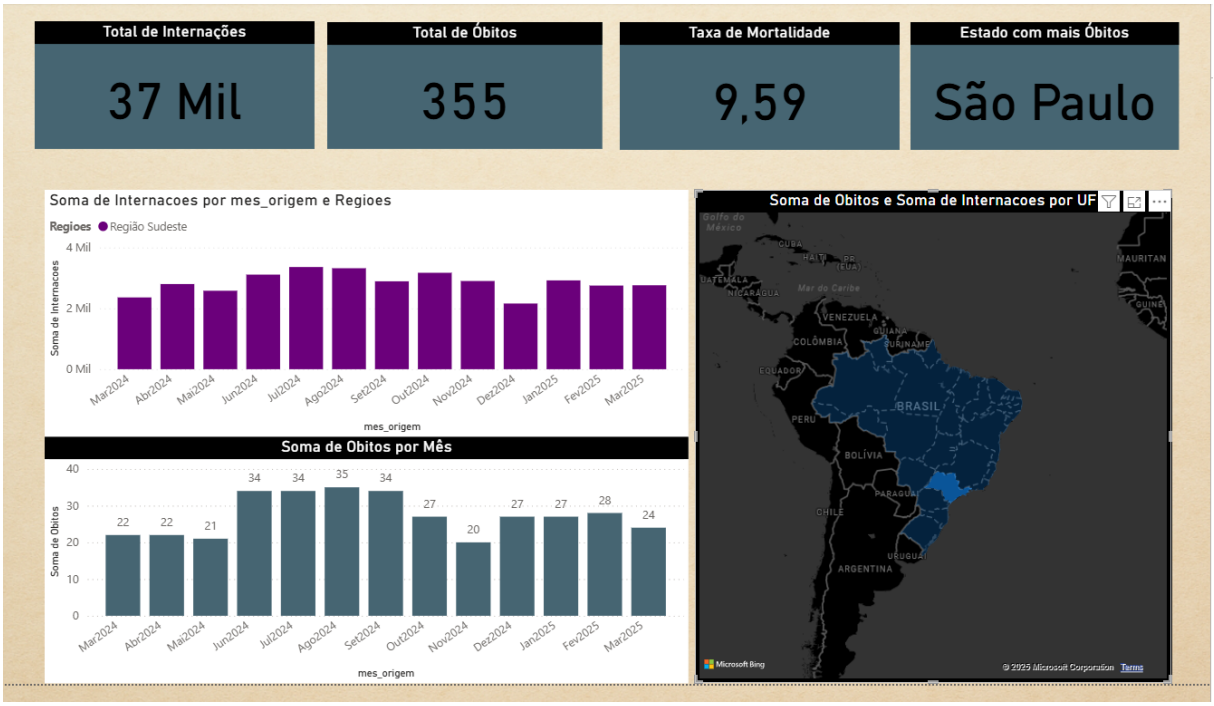
4 Resultados

Esta seção apresenta os resultados obtidos a partir do processamento e análise dos dados, com destaque para as visualizações desenvolvidas. São ilustradas as principais interfaces criadas no *Power BI*, evidenciando como os dados foram organizados e explorados para gerar informações relevantes sobre a morbidade hospitalar por HIV.

4.1 *Dashboard* Gráficos Interativos

As Figuras 5 e 6 ilustram o *dashboard* interativo desenvolvido no *Power BI* para análise dos dados de morbidade hospitalar por HIV. A interface foi projetada para facilitar a visualização de indicadores-chave relacionados à morbidade hospitalar, como o total de internações, números de óbitos, taxa de mortalidade, séries temporais e distribuição geográfica por unidade federativa. Os dados podem ser explorados com ou sem filtros regionais, permitindo a identificação de padrões e variações ao longo do tempo e entre diferentes estados. Na visualização com filtro aplicado para a Região Nordeste, destacam-se os seguintes elementos, conforme mostrado na Figura 5.

Figura 5 – *Dashboard* Gráficos Interativos

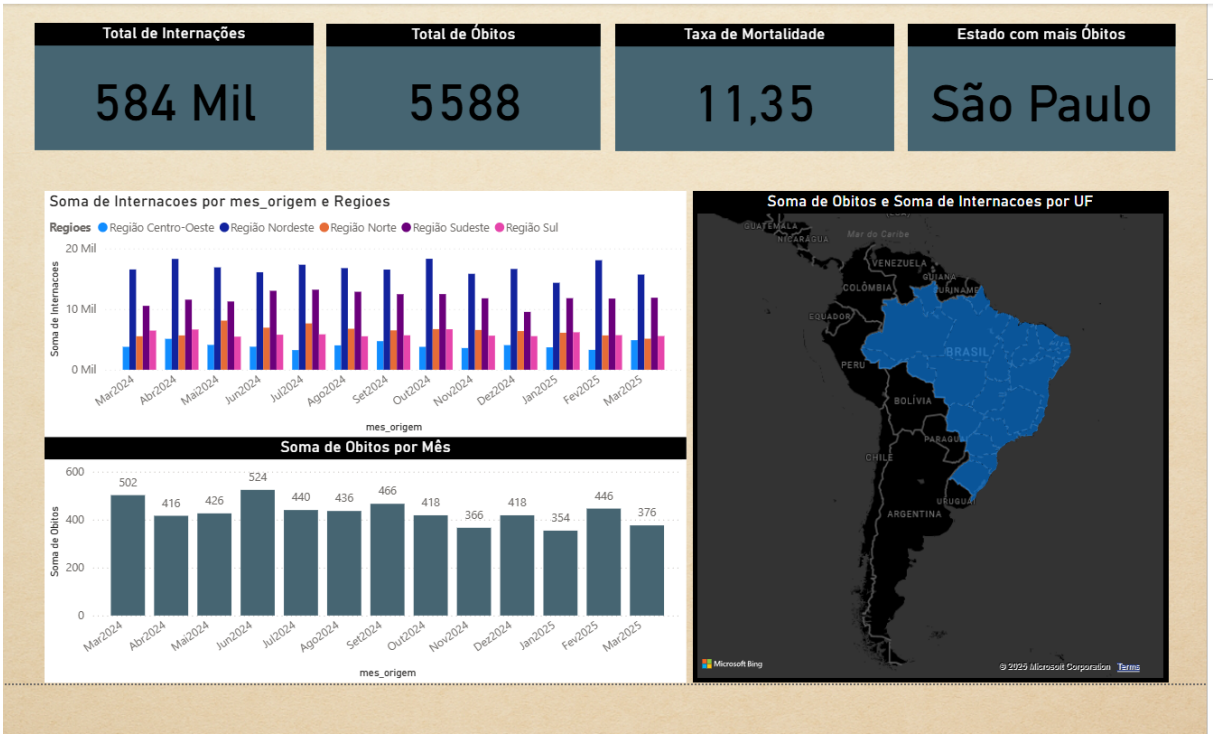


Fonte: Do Autor

Já na Figura 6, o filtro é removido, mostrando os dados consolidados de todo o ter-

ritório nacional. Ao remover o filtro regional, é possível observar a comparação entre diferentes regiões do país, como a predominância de internações na Região Nordeste e a variação no número de óbitos ao longo dos meses. O estado com maior número absoluto de óbitos continua sendo destacado na parte superior direita do painel.

Figura 6 – *Dashboard* Gráficos Interativos



Fonte: Do Autor

4.2 Testes

Segundo Sommerville (2025b), os testes são essenciais para verificar se o sistema atende corretamente aos requisitos definidos e se comporta conforme o esperado diante de diferentes cenários, o que reforça a importância de validar todo o processo de desenvolvimento, mesmo em projetos individuais.

No contexto deste trabalho, a fase de testes teve como objetivo geral assegurar a confiabilidade, integridade e funcionalidade de todas as etapas do projeto, desde a extração e tratamento dos dados até a exibição final nos *dashboards* interativos.

Com base nas boas práticas descritas pelo Codidata (2025), compreende-se que essa etapa vai além de conferir se os gráficos aparecem corretamente. É necessário garantir que os dados utilizados sejam precisos e preservem a integridade em relação às fontes originais, assegurando que as conexões com o banco de dados funcionem adequadamente, que os filtros produzam resultados coerentes e que não ocorram distorções nos indicadores. Essa

validação contínua é fundamental para garantir que as visualizações representem fielmente a realidade dos dados e ofereçam suporte a interpretações confiáveis.

Dessa forma, os testes realizados foram organizados em três categorias principais: integração de dados, interface e funcionalidade e validação dos indicadores.

4.2.1 Testes de Integração de Dados

Essa etapa verificou se os dados eram corretamente extraídos, tratados e carregados no banco PostgreSQL, preservando sua integridade e consistência em relação às fontes originais. Foram avaliados:

- Qualidade e padronização dos arquivos importados;
- Correção dos tipos de dados e valores numéricos;
- Integridade das tabelas após a carga no banco de dados.

Exemplos de testes realizados:

- Carregamento de arquivos CSV no *Jupyter Notebook*, tratando caracteres especiais e linhas corrompidas com on_bad_lines='skip' e encoding='latin1';
- Padronização de nomes de colunas, eliminando *underscores* duplicados e símbolos no início ou fim;

4.2.2 Testes de Interface e Funcionalidade

Essa categoria avaliou o funcionamento das conexões e interações entre as ferramentas utilizadas, especialmente entre *PostgreSQL* e *Power BI*.

- Conferência da conexão *Power BI* *PostgreSQL* e da correta visualização das tabelas;
- Revisão das *queries* de inserção no Pandas com .to_sql() e verificação da presença de dados no banco ($SELECT\ COUNT(*)$).

4.2.3 Validação dos Indicadores

Nessa etapa, foram realizados testes de validação para assegurar que os *dashboards* desenvolvidos no *Power BI* estivessem representando os dados de forma precisa. Verificou-se o correto funcionamento dos filtros, a fidelidade dos gráficos em relação aos dados tratados, e a coerência dos indicadores exibidos. Esse processo garante que a análise apresentada esteja alinhada com os dados originais, contribuindo para uma interpretação confiável.

Os problemas identificados e suas respectivas soluções durante essa fase de testes são descritos na Tabela 1.

Tabela 1 – Testes Realizados Durante o Desenvolvimento

Teste Realizado	**Falha Encontrada**	**Motivo**	**Solução**
Carregar arquivos csv no *Jupyter Notebook*	Erro de leitura em alguns arquivos	Presença de caracteres especiais e linhas corrompidas no final dos arquivos	Utilização do parâmetro *on_bad_lines*='*skip*' e codificação *encoding*='latin1' para ignorar erros de leitura e tratar corretamente os acentos
Padronizar os nomes das colunas	Algumas colunas foram renomeadas com *underscores* duplicados ou nomes incompletos	A função de normalização estava aplicando múltiplos *replace* sem controle	Adição de um laço *while* que substitui múltiplos *underscores* seguidos (__) por apenas um _, além do uso de *strip*('_') para remover símbolos no início ou fim
Conectar o Power BI ao banco PostgreSQL	Conexão bem-sucedida, mas os dados não apareciam na visualização	A tabela carregada estava vazia porque o script Python não populou a tabela corretamente	Revalidação da *query* de inserção no Pandas usando .to_sql() com *if_exists*='*replace*', e conferência da existência de dados no banco com uma *query* simples (*SELECT COUNT*(*))
Exibir taxa de mortalidade no *Power BI*	Indicador de taxa aparecia como 0% ou 100% em todos os casos	Problemas de formatação da coluna Taxa_mortalidade, que foi lida como *string* e não foi convertida corretamente	Conversão da coluna com df['Taxa_mortalidade'] = df['Taxa_mortalidade'].str.*replace*(',', '.').*astype*(*float*) antes de exportar os dados

Fonte: Do Autor

5 Conclusão

Este projeto atingiu seu objetivo principal ao desenvolver uma ferramenta de visualização interativa dos dados de morbidade hospitalar por HIV no Brasil, utilizando dados públicos do DATASUS. Através da extração, tratamento e organização estruturada dos dados em um banco relacional, que possibilita consultas ágeis e análises temporais e geográficas, foi possível criar um *dashboard* no *Power BI* que facilita a exploração dinâmica dos indicadores, como internações, óbitos e taxa de mortalidade.

As tecnologias empregadas, especialmente *Python* para o processamento dos dados e *Power BI* para a visualização, mostraram-se adequadas para lidar com grandes volumes de informações e construir painéis intuitivos. A modelagem relacional utilizada no banco PostgreSQL permitiu organizar os dados de forma lógica, garantindo qualidade e eficiência nas consultas.

Entre as limitações, o projeto não se aprofunda na análise epidemiológica nem desenvolve modelos estatísticos preditivos, focando exclusivamente na organização e disponibilização visual dos dados.

Como potencial de expansão, o sistema pode incorporar novas variáveis clínicas, períodos históricos adicionais ou integrar outras bases públicas de saúde, além de utilizar tecnologias mais avançadas de análise de dados e inteligência artificial.

Em resumo, o trabalho oferece uma base sólida para o monitoramento de indicadores de saúde pública e representa um avanço prático no uso da ciência de dados para apoiar decisões estratégicas, contribuindo potencialmente para o aprimoramento de políticas públicas relacionadas ao HIV no Brasil.

5.1 Trabalhos Futuros

Embora o trabalho desenvolvido tenha atingido os objetivos propostos, identifica-se a oportunidade de expandir suas funcionalidades para abranger ainda mais as necessidades. A seguir, serão apresentadas propostas que podem enriquecer o sistema, tornando-o mais completo e interativo, e proporcionando uma experiência ainda mais eficiente.

- Inclusão de novas variáveis como faixa etária e sexo, o que permitiria uma análise mais segmentada e aprofundada dos dados;
- Integração com outras bases de dados, possibilitando uma visão mais ampla da situação de saúde pública.

- Disponibilização do *dashboard* em um sistema web, permitindo que profissionais da saúde, gestores e cidadãos em geral tenham acesso às informações em tempo real, a partir de qualquer dispositivo com acesso à internet.

Essas melhorias visam não apenas aumentar a utilidade prática do sistema, mas também contribuir para a democratização do acesso à informação, ampliando o impacto social da ferramenta e fortalecendo o uso de dados abertos na construção de políticas públicas baseadas em evidências.

Referências

ALMEIDA, M. *Pandas: o que é, para que serve e como instalar*. 2023. https://www.alura.com.br/artigos/pandas-o-que-e-para-que-serve-como-instalar. Publicado em: 12 jan. 2023. Acesso em: 08 jul. 2025.

ALMEIDA, M. *Power BI: o que é, para que serve e como usar essa ferramenta*. 2023. https://www.alura.com.br/artigos/power-bi?srsltid=AfmBOopdEJTIWkJKF76lA9meNUaqLOeKZp9ggoupy9QJjD5zDc67P8Ht. Publicado em: 22 fev. 2023. Acesso em: 27 jun. 2025.

Amazon Web Services. *O que é Python? — AWS*. 2025. https://aws.amazon.com/pt/what-is/python/. Acesso em: 27 jun. 2025.

CARVALHO, A. C. P. L. F. de; MENEZES, A. G.; BONIDIA, R. P. *Ciência de Dados - Fundamentos e Aplicações*. Rio de Janeiro: LTC, 2024. E-book. p. 5. ISBN 9788521638766. Disponível em: https://integrada.minhabiblioteca.com.br/reader/books/9788521638766/. Acesso em: 23 jun. 2025.

CARVALHO, C. *Python — Artigos Alura*. 2025. https://www.alura.com.br/artigos/python?srsltid=AfmBOopcDGOMPqChN3oRbyx7cTEblNpCE2JOlb_rBGS_Mf6OyRNAZ-gZ. Acesso em: 27 jun. 2025.

Codidata. *Lições sobre Controle de Qualidade em Dashboards*. 2025. https://codidata.com.br/glossario/licoes-sobre-controle-de-qualidade-em-dashboards/?utm_source=chatgpt.com. Acesso em: 07 jul. 2025.

Databricks. *Jupyter Notebook - Glossary*. 2025. https://www.databricks.com/glossary/jupyter-notebook. Acesso em: 27 jun. 2025.

FELíCIO, L. R. Perfil de mortalidade e morbidade hospitalar: Análise de uma década em um município brasileiro de médio porte com base no datasus. *Revista Foco*, v. 17, n. 1, p. e4107, 2024. Acesso em: 24 jun. 2025. Disponível em: <https://ojs.focopublicacoes.com.br/foco/article/view/4107>.

HANNA. *Desenvolvimento de Dashboard Educacional*. 2025. https://www.monografias.ufop.br/bitstream/35400000/3604/1/MONOGRAFIA_DesenvolvimentoDashboardEducacional.pdf. Acesso em: 15 jul. 2025.

IBM. *PostgreSQL — IBM Think*. 2025. https://www.ibm.com/br-pt/think/topics/postgresql. Acesso em: 27 jun. 2025.

LIMA. *Repositório Institucional da UFBA - Trabalho Acadêmico*. 2025. https://repositorio.ufba.br/handle/ri/38870. Acesso em: 15 jul. 2025.

Microsoft. *Visão geral do Power BI*. 2025. https://learn.microsoft.com/pt-br/power-bi/fundamentals/power-bi-overview. Acesso em: 27 jun. 2025.

Ministério da Saúde. *Boletim Epidemiológico número especial — dezembro de 2024*. 2024. https://www.gov.br/aids/pt-br/central-de-conteudo/

boletins-epidemiologicos/2024/boletim_hiv_aids_2024e.pdf/view. Atualizado em: 11 dez. 2024, 11h59. Acesso em: 06 jul. 2025.

Ministério da Saúde. *Vigilância Epidemiológica — Saúde de A a Z*. 2025. https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/dst/vigilancia-epidemiologica. Acesso em: 06 jul. 2025.

MORAIS, I. S.; ZANIN, A. *Engenharia de software*. Porto Alegre: SAGAH, 2020. E-book. p. 96. ISBN 9788595022539. Disponível em: https://integrada.minhabiblioteca.com.br/reader/books/9788595022539/. Acesso em: 25 jun. 2025.

PostgreSQL Global Development Group. *About PostgreSQL*. 2025. https://www.postgresql.org/about/. Acesso em: 27 jun. 2025.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*. 9. ed. Porto Alegre: AMGH, 2021. E-book. p. 122. ISBN 9786558040118. Disponível em: https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/. Acesso em: 25 jun. 2025.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*. 9. ed. Porto Alegre: AMGH, 2021. E-book. p. 49. ISBN 9786558040118. Disponível em: https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/. Acesso em: 25 jun. 2025.

Project Jupyter. *What is Jupyter?* 2025. https://docs.jupyter.org/en/latest/what_is_jupyter.html. Acesso em: 27 jun. 2025.

SEBRAE. *Método Kanban*. 2023. https://sebrae.com.br/Sebrae/Portal%20Sebrae/Arquivos/ebook_sebrae_metodo-kanban.pdf. E-book Sebrae. Publicado em: 07 ago. 2023. Acesso em: 29 jun. 2025.

SILBERSCHATZ, A. *Sistema de Banco de Dados*. 7. ed. Rio de Janeiro: GEN LTC, 2020. E-book. p. 157. ISBN 9788595157552. Disponível em: https://integrada.minhabiblioteca.com.br/reader/books/9788595157552/. Acesso em: 25 jun. 2025.

SOMMERVILLE, I. *Engenharia de Software*. 2025. https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf. Acesso em: 25 jul. 2025.

SOMMERVILLE, I. *Engenharia de Software — Capítulo 8*. 2025. https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf. Acesso em: 07 jul. 2025.

UNAIDS. *Country Focus: Brazil*. 2025. https://www.unaids.org/en/20191011_country_focus_Brazil. Acesso em: 06 jul. 2025.

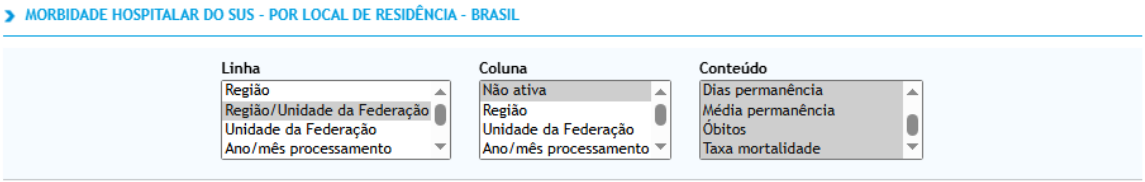
A – Extração de dados TABNET

Este apêndice descreve, passo a passo, o processo de extração de dados do DATASUS utilizado para a criação dos *dashboards* apresentados neste trabalho.

A Figura 7 mostra a interface do DATASUS com os filtros aplicados para a obtenção dos dados de morbidade hospitalar por local de residência. Destacam-se:

- Seleção da linha "Região/Unidade da Federação";
- Configuração da coluna como "Não ativa";
- Escolha dos indicadores na aba Conteúdo, incluindo dias de permanência, média de permanência, óbitos e taxa de mortalidade.

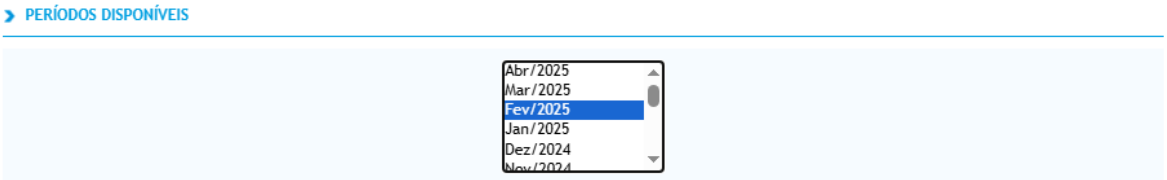
Figura 7 – Morbidade Hospitalar do SUS – Por local de Residência - Brasil



Fonte: DATASUS

A seleção do filtro "Períodos disponíveis", etapa em que foi definido o mês de fevereiro de 2025 como intervalo temporal para a extração dos dados, é exibida na Figura 8. Essa escolha garante que a análise seja realizada com base nas internações hospitalares registradas especificamente nesse período.

Figura 8 – Períodos Disponíveis

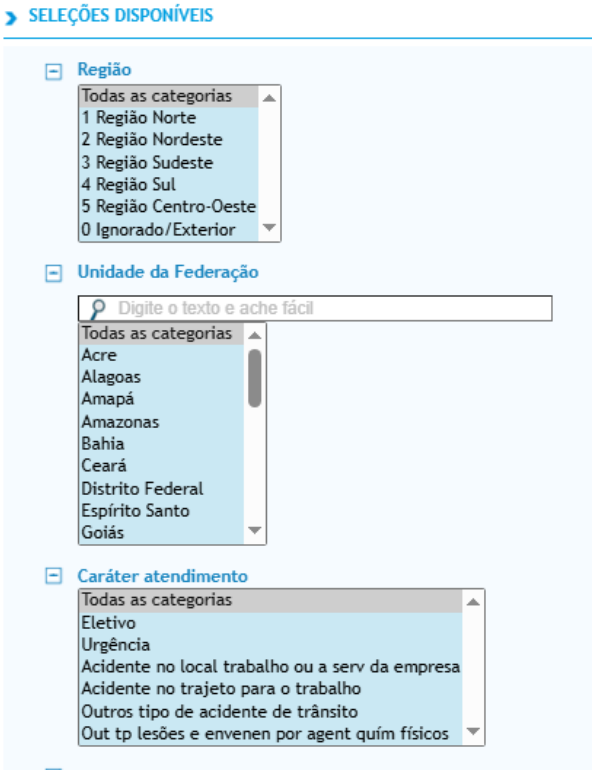


Fonte: DATASUS

Na Figura 9, é apresentada a etapa de configuração das variáveis "Regiões", "Unidade da Federação" e "Caráter do Atendimento" no sistema do DATASUS. Todas as opções dessas categorias foram selecionadas para garantir uma cobertura abrangente dos dados.

Essa configuração assegura que os dados extraídos incluam todas as regiões do país, unidades federativas e tipos de atendimento hospitalar registrados no período especificado.

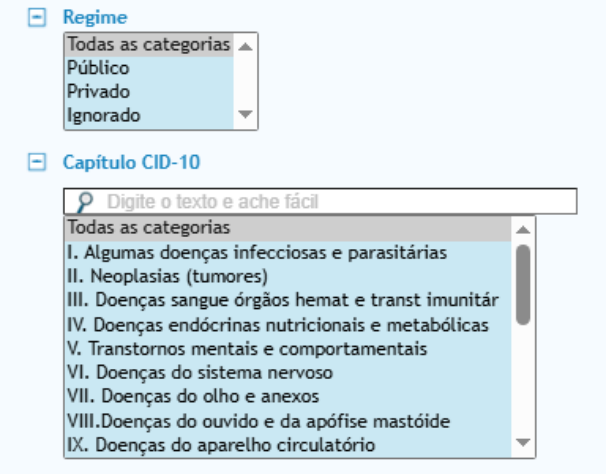
Figura 9 – Seleção de variáveis categóricas para filtragem.



Fonte: DATASUS

A etapa de configuração das variáveis "Regime"e "Capítulo CID-10"no sistema do DATASUS, ambas definidas com a opção "Todas as categorias", garantindo abrangência total dos dados está ilustrada na Figura 10. Essa definição assegura que os dados coletados incluam todos os tipos de regime de atendimento hospitalar e todos os capítulos da Classificação Internacional de Doenças (CID-10), sem restrições por natureza clínica ou administrativa.

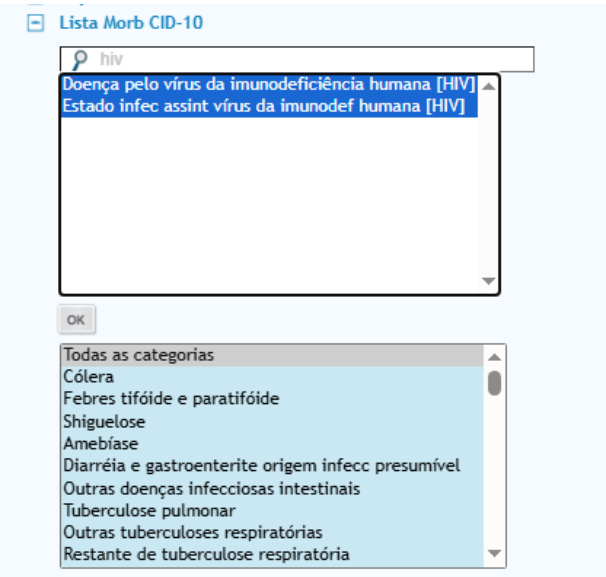
Figura 10 – Seleção das variáveis “Regime de Atendimento”e “Capítulo CID-10”.



Fonte: DATASUS

A Figura 11 apresenta a etapa de seleção de causas específicas na lista MORB CID-10, com destaque para as categorias relacionadas ao vírus da imunodeficiência humana (HIV). Foram selecionadas as opções “Doença pelo vírus da imunodeficiência humana (HIV) ” e “Estado infeccioso assintomático pelo vírus da imunodeficiência humana (HIV) ”. Na parte inferior da tela, a configuração permanece como “Todas as categorias”, garantindo que todos os desdobramentos clínicos vinculados a essas condições sejam incluídos na extração dos dados.

Figura 11 – Seleção de causas específicas na lista MORB CID-10.

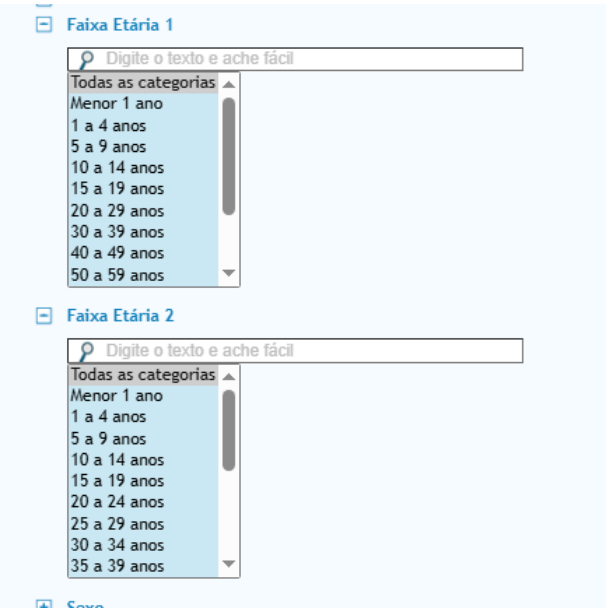


Fonte: DATASUS

A Figura 12 ilustra a etapa de definição dos filtros “Faixa Etária 1” e “Faixa Etária 2” no sistema do DATASUS, ambos configurados com a opção “Todas as categorias”. Essa

configuração possibilitou que os dados extraídos contemplassem todas as faixas etárias disponíveis, desde recém-nascidos até idosos, garantindo uma análise abrangente dos registros hospitalares.

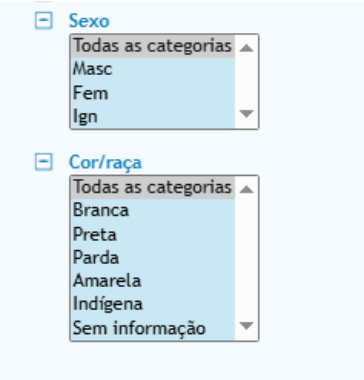
Figura 12 – Seleção das variáveis "Faixa Etária 1"e "Faixa Etária 2".



Fonte: DATASUS

A etapa de definição dos filtros "Sexo"e "Cor/Raça"no sistema do DATASUS, ambos configurados com a opção "Todas as categorias"está detalhada na Figura 13. Essa configuração assegura a inclusão de todos os registros independentemente do sexo ou da auto declaração racial dos pacientes, promovendo uma extração abrangente e sem segmentações demográficas.

Figura 13 – Seleção das variáveis "Sexo"e "Cor/Raça".



Fonte: DATASUS