UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Filipe Gabriel Oliveira da Cruz

SQL para Ciência de Dados aplicada em dados de saúde pública no Brasil

Uberlândia, Brasil

2026

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Filipe Gabriel Oliveira da Cruz

SQL para Ciência de Dados aplicada em dados de saúde pública no Brasil

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Dra. Maria Camila Nardini Barioni

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2026

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Computação
Av. João Naves de Ávila, 2121, Bloco 1B - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4393 - www.facom.ufu.br - facom@ufu.br

ATA DE DEFESA - GRADUAÇÃO

Curso de Graduação em:	Bacharelado em Ciência da Computação				
Defesa de:	Projeto de Graduação 2 - GBC082				
Data:	21/07/2026	Hora de início:	15:00	Hora de encerramento:	16:01
Matrícula do Discente:	12211BCC020				
Nome do Discente:	Filipe Gabriel Oliveira da Cruz				
Título do Trabalho:	SQL para Ciência de Dados aplicada em dados de saúde pública no Brasil				

A carga horária curricular foi cumprida integralmente?	(X) Sim ( ) Não

Reuniu-se de forma remota pela plataforma MS Teams, a Banca Examinadora, designada pelo Colegiado do Curso de Graduação em Bacharelado em Ciência da Computação, assim composta: Profa. Alessandra Aparecida Paulino - FACOM/UFU; Profa. Christiane Regina Soares Brasil - FACOM/UFU; e Profa. Maria Camila Nardini Barioni - FACOM/UFU, orientadora do candidato.

Iniciando os trabalhos, a presidente da mesa, Dr(a). Maria Camila Nardini Barioni, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao discente a palavra, para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do curso.

A seguir a senhora presidente concedeu a palavra, pela ordem sucessivamente, às examinadoras, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

(X) Aprovado(a) Nota [85]

OU

( ) Aprovado(a) sem nota.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Christiane Regina Soares Brasil**, **Professor(a) do Magistério Superior**, em 21/07/2026, às 16:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Maria Camila Nardini Barioni**, **Professor(a) do Magistério Superior**, em 21/07/2026, às 16:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Alessandra Aparecida Paulino**, **Professor(a) do Magistério Superior**, em 21/07/2026, às 16:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **7486839** e o código CRC **A672D68F**.

Referência: Processo nº 23117.047102/2026-24 SEI nº 7486839

Agradecimentos

Olhando para trás, percebo que iniciei esta caminhada focado apenas na linha de chegada e no acalento do retorno para casa. No entanto, o caminho me surpreendeu. O que começou como uma missão de entrega transformou-se em uma trajetória de descobertas e de uma felicidade sincera que eu não imaginava encontrar. Concluo este ciclo com o coração grato, sabendo que a saudade que sinto hoje é a maior prova de que cada passo valeu a pena.

A Deus, minha companhia em mar afora, a quem confio meu caminho, fonte de minha esperança diária e quem permitiu que eu encerrasse este ciclo com resiliência e fé.

À minha família, em especial aos meus pais, ao meu irmão e ao meu cachorrinho. Vocês são a maior bênção na minha vida, meu porto seguro e a razão de eu chegar até aqui.

Aos amigos que fiz nesta jornada, pelo apoio, pelos momentos vividos e por tornarem essa parte da minha história inesquecível.

À minha orientadora, Professora Dra. Maria Camila Nardini Barioni, por acreditar neste projeto e me guiar com tanta paciência e sabedoria. Aos demais professores do curso, minha gratidão pelo aprendizado compartilhado ao longo destes anos.

Resumo

Embora pipelines tradicionais de Ciência de Dados dependam da exportação de grandes volumes de informação para ambientes imperativos externos, os Sistemas de Gerenciamento de Bancos de Dados (SGBDs) relacionais evoluíram para suportar lógicas analíticas complexas. O paradigma de análise *In-database* surge como uma solução para eliminar os custos de movimentação e serialização de dados através da rede, aproveitando otimizadores lógicos nativos. Este trabalho tem como objetivo demonstrar a viabilidade de se executar um ciclo de vida analítico unificado utilizando exclusivamente comandos SQL nativos e recursos analíticos do PostgreSQL. Para validar empiricamente a robustez e a expressividade da linguagem, implementou-se um estudo de caso aplicado sobre as notificações de dengue no Brasil em 2025 (SINAN), integrando-as a indicadores demográficos (IBGE) e de classificação ocupacional (CBO). O método de trabalho compreendeu o desenho de uma arquitetura resiliente dividida em uma *Staging Area* tolerante a falhas, saneamento de chaves, normalização na Terceira Forma Normal (3FN) e transposição de matrizes esparsas utilizando o operador `CROSS JOIN LATERAL`. A infraestrutura de dados foi acelerada por meio de indexação física (B-Tree) e consolidada em uma *Materialized View*. Os resultados comprovam a eficiência prática do ecossistema relacional, que processou um volume superior a 1,65 milhão de registros com tempos de resposta na ordem de um segundo, computando taxas de incidência, séries temporais cumulativas via *Window Functions*, engenharia de atributos etários e correlações estatísticas de Pearson. O trabalho contribui ao desmistificar o uso do SGBD relacional como repositório passivo, apresentando um roteiro reprodutível e performático voltado à Ciência de Dados puramente estruturada em SQL.

Palavras-chave: Ciência de Dados, Linguagem SQL, Análise *In-database*, PostgreSQL, Saúde Pública.

Abstract

Although traditional Data Science pipelines rely on exporting large volumes of information to external imperative environments, relational Database Management Systems (DBMSs) have evolved to support complex analytical logic. The *In-database* analytics paradigm emerges as a solution to eliminate data movement and serialization costs across the network by leveraging native logical optimizers. This work aims to demonstrate the viability of executing a unified analytical lifecycle using exclusively native SQL commands and analytical resources of PostgreSQL. To empirically validate the language's robustness and expressiveness, a case study was implemented on dengue notifications in Brazil in 2025 (SINAN), integrating them with demographic indicators (IBGE) and occupational classification (CBO). The work methodology comprised designing a resilient architecture divided into a fault-tolerant *Staging Area*, key cleansing, Third Normal Form (3FN) normalization, and the transposition of sparse matrices using the `CROSS JOIN LATERAL` operator. The data infrastructure was accelerated through physical indexing (B-Tree) and consolidated into a *Materialized View*. The results prove the practical efficiency of the relational ecosystem, which processed a volume exceeding 1.65 million records with response times on the order of one second, computing incidence rates, cumulative time series via *Window Functions*, age attribute engineering, and Pearson statistical correlations. The work contributes to demystifying the use of the relational DBMS as a passive repository, presenting a reproducible and performant roadmap focused on Data Science purely structured in SQL.

Keywords: Data Science, SQL Language, In-database Analytics, PostgreSQL, Public Health.

DECLARAÇÃO SOBRE O USO DE INTELIGÊNCIA ARTIFICIAL

Eu, **Filipe Gabriel Oliveira da Cruz**, discente do curso de Bacharelado **em Ciência da Computação** da Faculdade de Computação da Universidade Federal de Uberlândia, regularmente matriculado sob o número **12211bcc020**, declaro para os devidos fins que as informações prestadas a seguir refletem de forma verídica e completa o uso ou não uso de ferramentas de Inteligência Artificial neste trabalho, em atenção às recomendações para o uso e desenvolvimento ético e responsável de inteligência artificial na Universidade Federal de Uberlândia, versão 7 de 2025.

SELECIONE UMA DAS OPÇÕES A SEGUIR:

☐ **NÃO HOUVE USO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL**
(Se marcar esta opção, não é necessário preencher os campos 1 a 4)
Declaro que não utilizei, em nenhuma etapa do desenvolvimento do presente trabalho, ferramentas de Inteligência Artificial para qualquer finalidade e assumo integral responsabilidade pelo conteúdo.

☒ **HOUVE USO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL**
(Preencher os campos obrigatórios abaixo, assinalando obrigatoriamente os itens 3 e 4)
Declaro que utilizei, durante o desenvolvimento do presente trabalho, as seguintes ferramentas de Inteligência Artificial e informações descritas nos itens de 1 a 4.

1. Ferramenta(s) e versão(ões) predominante(s):

Gemini 3.5 Flash

2. Propósito(s) (marque todos os que se aplicam):

☐ Exploração inicial de ideias.

☐ Busca de referências bibliográficas.

☒ Revisão de linguagem (por exemplo, melhoria de gramática ou correção de ortografia).

☒ Tradução técnica (conversão do texto ou partes dele para outro idioma).

☒ Geração automatizada de tabelas, gráficos ou figuras a partir de dados previamente analisados pelos autores.

☒ Programação: sugestão, depuração e documentação de códigos computacionais.

☐ Outros (especificar): _____

3. Autoria e validação humana:

☒ Informo que a autoria intelectual do manuscrito, bem como sua supervisão e validação, são de minha responsabilidade e que o uso de ferramentas de IA foi exclusivamente para os propósitos acima declarados.

4. Princípios éticos:

☒ Declaro que a utilização das ferramentas de IA seguiu práticas éticas, não incluindo atividades que comprometam a integridade científica tais como a geração de conteúdo original, manipulação de dados, plágio, interpretações ou análises pela IA.

☒ Declaro que respeitei direitos autorais, licenças, confidencialidade e políticas editoriais.

☒ Declaro que não enviei dados inéditos ou sensíveis a serviços que utilizam conteúdos para treinamento de modelos, exceto em plataformas institucionais ou com garantias contratuais de confidencialidade e não retenção, assegurando conformidade com a LGPD (Lei nº 13.709/2018) e demais normas de proteção de dados.

Uberlândia, 04 de agosto de 2026.

gov.br
Documento assinado digitalmente
FILIPE GABRIEL OLIVEIRA DA CRUZ
Data: 04/08/2026 10:11:47-0300
Verifique em https://validar.iti.gov.br

Filipe Gabriel Oliveira da Cruz

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

SGBD	Sistema Gerenciador de Banco de Dados
IBGE	Instituto Brasileiro de Geografia e Estatística
SINAN	Sistema de Informação de Agravos de Notificação
MTE	Ministério do Trabalho e Emprego
CBO	Classificação Brasileira de Ocupações

Sumário

1 INTRODUÇÃO

Com o avanço computacional e a exacerbada geração de dados, tornou-se essencial o uso de mecanismos que permitissem o armazenamento, a organização e a recuperação eficiente dessas informações. Desde a década de 1970, com a proposta do modelo relacional por Edgar F. Codd, os sistemas gerenciadores de banco de dados (SGBDs) passaram a representar um pilar fundamental na construção de sistemas computacionais, permitindo consultas e manipulações estruturadas por meio da linguagem SQL (*Structured Query Language*) (CODD, 1970). Mesmo após décadas, os SGBDs relacionais permanecem amplamente utilizados em empresas e instituições, figurando entre os mais populares segundo rankings atualizados como o DB-Engines (DB-ENGINES, 2026). Com o tempo, a linguagem SQL evoluiu significativamente: desde a padronização ANSI-SQL:1999, foram introduzidos recursos como *Common Table Expressions* (CTEs), *Window Functions* e agregações avançadas, que permitem realizar operações complexas sem a necessidade de linguagens externas (EISENBERG; MELTON, 1999). No ecossistema open-source, o PostgreSQL evoluiu além do armazenamento transacional, expandindo seus recursos nativos para atender à alta demanda analítica. Esses avanços demonstram a viabilidade técnica de se executar etapas da Ciência de Dados — como tratamento, análise e modelagem — diretamente no banco de dados, sem extrair os dados para ferramentas externas como Python ou R. Dessa forma, abre-se a oportunidade de explorar e divulgar práticas de Ciência de Dados puramente baseadas em SQL, com potencial de promover maior integração, segurança e desempenho nos fluxos de análise.

Nos cursos de Ciência da Computação, os conceitos e práticas de SGBDs são componentes centrais da formação acadêmica, frequentemente incluídos como disciplinas obrigatórias conforme as diretrizes curriculares da área. Os estudantes aprendem desde a modelagem relacional e consultas SQL até a otimização de desempenho e arquitetura de bancos relacionais, adquirindo competências que os tornam aptos a realizar diversas etapas do processo de Ciência de Dados diretamente em SQL. A literatura técnica reforça essa perspectiva ao afirmar que a estrutura lógica e declarativa da linguagem SQL permite implementar transformações, filtragens, agrupamentos e correlações sem recorrer, necessariamente, a linguagens imperativas como Python ou R (SOUZA, 2024). Além disso, o domínio da linguagem SQL figura entre as habilidades mais requisitadas para profissionais de dados, tanto em ambientes corporativos quanto acadêmicos. Segundo levantamento da Revista Ensino Superior, a linguagem SQL aparece como uma das competências técnicas mais exigidas em vagas para analistas e cientistas de dados no Brasil, destacando-se ao lado de ferramentas como Power BI, Excel e Python (SUPERIOR, 2024). Essa demanda justifica-se por sua alta eficiência na manipulação de dados estruturados, ampla adoção

em sistemas organizacionais e compatibilidade com as principais plataformas de bancos de dados relacionais. Nesse cenário, demonstrar a viabilidade de se realizar Ciência de Dados puramente com SQL, dentro do próprio SGBD, reforça não apenas a coerência pedagógica com a formação oferecida nos cursos de Computação, mas também oferece uma contribuição prática para alunos e profissionais que desejam aproveitar ao máximo os recursos já dominados.

Diversos trabalhos acadêmicos demonstram que etapas essenciais da Ciência de Dados, como limpeza, transformação e análise exploratória, podem ser realizadas diretamente com SQL. Na UFU, Vieira (2024) desenvolveu uma ferramenta que utiliza consultas SQL com GROUP BY, ROLLUP e CUBE para gerar cubos de dados e visualizações, sem a necessidade de ferramentas externas (VIEIRA, 2024). Já o projeto SQLShare, da Universidade de Washington, mostrou que cientistas de áreas diversas conseguem executar análises completas usando apenas SQL, evidenciando seu potencial para estruturar pipelines analíticos inteiros dentro do banco de dados (JAIN; BECLA; HALPERIN, 2016). Esses estudos reforçam a viabilidade prática e acadêmica da abordagem adotada neste TCC.

O objetivo geral deste Trabalho de Conclusão de Curso consistiu em demonstrar que etapas fundamentais da Ciência de Dados podem ser realizadas exclusivamente dentro de um sistema gerenciador de banco de dados relacional, utilizando comandos SQL nativos e recursos analíticos do PostgreSQL. Para isso, utilizou-se um conjunto de dados públicos contendo as notificações de dengue no Brasil, disponibilizado pelo Ministério da Saúde via Sistema de Informação de Agravos de Notificação (SINAN), integrado a indicadores demográficos municipais do Instituto Brasileiro de Geografia e Estatística (IBGE) e à estrutura da Classificação Brasileira de Ocupações (CBO). A partir desse ecossistema, aplicaram-se operações comuns da prática de Ciência de Dados, como limpeza, filtragem, modelagem relacional na Terceira Forma Normal (3FN), agregação e análise exploratória. O trabalho comprovou na prática a capacidade do SQL em atender a demandas reais de engenharia e análise de dados sem a necessidade de exportação para ambientes externos como Python ou R, oferecendo um roteiro técnico-didático para estudantes e profissionais interessados em explorar essa abordagem.

Durante o desenvolvimento deste trabalho, utilizaram-se ferramentas computacionais que viabilizaram a implementação da proposta de análise de dados puramente em SQL. O principal sistema de gerenciamento de banco de dados empregado foi o SGBD relacional PostgreSQL, executado nativamente em ambiente local sob o sistema operacional Windows 11. Para o acesso, modelagem, manipulação e execução de comandos SQL e scripts de auditoria, utilizou-se a ferramenta de interface gráfica oficial pgAdmin. Em relação aos dados brutos, realizou-se a ingestão em massa dos arquivos planos em formato CSV correspondentes ao ano epidemiológico de 2025 para tabelas de isolamento, permi-

tindo o carregamento integral na base e a posterior experimentação iterativa por meio de consultas analíticas.

Este trabalho caracteriza-se como uma Pesquisa Científica — Explicativa, conforme os critérios apresentados no Capítulo 5 de (WAZLAWICK, 2020). A hipótese que orientou esta investigação postulou que "etapas fundamentais da Ciência de Dados — como limpeza, transformação, modelagem estruturada e análise estatística — podem ser realizadas exclusivamente com SQL nativo no PostgreSQL, mantendo a integridade e a eficiência prática observadas em pipelines externos". Para testá-la, construiu-se um ambiente analítico completo dentro do SGBD, aplicando-se consultas SQL estruturadas e funções estatísticas diretamente sobre o volume consolidado de notificações de dengue e dados demográficos do IBGE. O estudo adotou os princípios da ciência empírica, baseando-se em dados observáveis (bases públicas do SINAN e IBGE), hipóteses testáveis (possibilidade de medir integridade relacional, performance analítica e reprodutibilidade), e objetividade (execução e documentação transparente das consultas e procedimentos). O enfoque esteve na validação técnica e prática da abordagem proposta como uma solução viável e adequada para análise de dados no contexto relacional.

Para além desta introdução, o presente trabalho está estruturado em seis capítulos subsequentes. O Capítulo 2 (Revisão Bibliográfica) apresenta a fundamentação teórica, abordando os conceitos de Ciência de Dados, Análise *In-database*, modelagem relacional e o comportamento epidemiológico da dengue. O Capítulo 3 dedica-se à análise dos trabalhos relacionados, contextualizando esta pesquisa frente ao estado da arte. Em seguida, o Capítulo 4 (Método de Trabalho) detalha minuciosamente o pipeline prático de Engenharia de Dados, compreendendo as etapas de ingestão na *Staging Area*, o processo de normalização lógica na Terceira Forma Normal (3FN) e as estratégias de otimização no PostgreSQL. O Capítulo 5 contempla a execução das consultas analíticas nativas e a apresentação e discussão dos resultados estatísticos obtidos. O Capítulo 6 é dedicado à Análise Crítica das Inconsistências do Dicionário Oficial (SINAN), discutindo os aspectos de qualidade de software e governança envolvidos no ecossistema de dados públicos. Por fim, no Capítulo 7, são expostas as considerações finais, as limitações identificadas no estudo e as sugestões para trabalhos futuros.

1.1 Organização da Monografia

Para além desta introdução, o presente trabalho está estruturado em seis capítulos subsequentes, organizados da seguinte forma:

- O **Capítulo 2 (Revisão Bibliográfica)** apresenta a fundamentação teórica sobre o Modelo Relacional, a evolução da linguagem SQL, a arquitetura dos SGBDs e o

paradigma de análise *In-database*, além de contextualizar a dinâmica epidemiológica da dengue no Brasil.

- O **Capítulo 3 (Trabalhos Correlatos)** analisa artigos e arquiteturas da literatura, destacando estudos focados no processamento analítico nativo e na aplicação de funcionalidades avançadas da própria linguagem SQL para Ciência de Dados.
- O **Capítulo 4 (Método de Trabalho)** detalha o fluxo completo de Engenharia de Dados implementado no PostgreSQL, desde a ingestão na *Staging Area* até a normalização na Terceira Forma Normal (3FN), transposição de matrizes clínicas e criação da *Materialized View* analítica.
- O **Capítulo 5 (Desenvolvimento e Análise dos Resultados)** apresenta a execução das consultas SQL nativas e a discussão dos resultados estatísticos, abordando taxas de incidência municipal, séries temporais, perfis demográficos, categorias ocupacionais e a correlação de Pearson.
- O **Capítulo 6 (Qualidade de Software e Governança de Dados Públicos)** traz uma análise crítica sobre as inconsistências do dicionário de dados oficial do SINAN e detalha a arquitetura de ingestão resiliente projetada para mitigar falhas.
- Por fim, o **Capítulo 7 (Conclusão e Recomendações para Trabalhos Futuros)** sintetiza as principais contribuições da pesquisa, expõe as limitações identificadas e sugere desdobramentos para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

O presente capítulo tem como objetivo apresentar os conceitos fundamentais que embasam o desenvolvimento deste trabalho, servindo como base teórica para a execução das etapas subsequentes. Foram discutidos tópicos relacionados à Ciência de Dados, Análise *In-database*, SQL para Análise de Dados, e técnicas de pré-processamento e modelagem de dados. Além disso, são abordadas referências que demonstram a evolução histórica e as tendências atuais da área, evidenciando práticas consolidadas e tecnologias emergentes.

A revisão busca contextualizar o uso do SQL como linguagem de Ciência de Dados, destacando a possibilidade de realizar análises complexas diretamente dentro do SGBD, sem necessidade de extração ou movimentação de dados. São mencionadas ferramentas e recursos nativos que permitem essa abordagem *In-database*, reforçando a ideia central do trabalho de explorar o potencial do SGBD como ambiente de Análise de Dados.

Essa fundamentação teórica permitiu uma compreensão sólida das ferramentas, métodos e técnicas utilizadas, garantindo que as escolhas técnicas adotadas no trabalho estejam alinhadas com os avanços recentes da área e com a proposta de utilizar o SGBD como principal ambiente de análise.

2.1 Modelo Relacional

O Modelo Relacional foi proposto por Edgar F. Codd no início da década de 1970 como uma nova forma de estruturar e manipular dados em computadores (CODD, 1970). Nesse modelo, os dados são organizados em relações, que podem ser representadas por tabelas, compostas por tuplas (linhas) e atributos (colunas). Cada relação possui uma chave primária que garante a unicidade dos registros e pode se relacionar com outras tabelas por meio de chaves estrangeiras.

Antes do modelo relacional, predominavam os Sistemas hierárquicos e os Sistemas de Rede. O Modelo Hierárquico, exemplificado pelo IMS da IBM, organizava os dados em estruturas de árvore, o que permitia acesso eficiente, mas dificultava consultas complexas e a flexibilidade de relacionamentos. Já o Modelo de Rede, formalizado pelo padrão CODASYL, representava dados como registros interligados por conjuntos, permitindo múltiplos relacionamentos, porém com grande complexidade de navegação (SILBERSCHATZ; KORTH; SUDARSHAN, 2020; WATT, 2014).

A simplicidade conceitual do Modelo Relacional, baseada em princípios da álgebra relacional e da lógica de predicados, superou essas limitações ao permitir que usuários

descrevessem o que desejavam consultar sem precisar especificar caminhos de acesso. Esse diferencial deu origem à linguagem SQL, que se consolidou como padrão de manipulação de bancos de dados (CODD, 1970; SILBERSCHATZ; KORTH; SUDARSHAN, 2020).

Na prática, o modelo relacional introduziu conceitos fundamentais que permanecem atuais, como a independência entre os dados e os programas que os manipulam, a normalização como técnica de eliminação de redundâncias e a integridade referencial. Esses princípios tornam os sistemas relacionais robustos e escaláveis, além de base para diversas extensões modernas.

Um exemplo prático dessa robustez pode ser observado na modelagem dos microdados do SINAN aplicada neste trabalho. Os arquivos planos brutos de notificações contêm colunas esparsas para sintomas e comorbidades que, sob os princípios da normalização relacional, foram decompostas em tabelas associativas e relacionamentos muitos-para-muitos. A presença de chaves primárias e estrangeiras permitiu vincular cada evento de agravo aos cadastros demográficos de municípios do IBGE e de ocupações da CBO, evidenciando a viabilidade prática do modelo relacional em estruturar e garantir a integridade de grandes volumes de dados de saúde pública (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025).

2.2 *Structured Query Language*

O SQL é uma linguagem declarativa e padronizada para gerenciamento e manipulação de bancos de dados relacionais. Diferente de linguagens imperativas, descreve o que deve ser feito com os dados, e não como fazê-lo. Sua base teórica é o modelo relacional proposto por Edgar Codd (CODD, 1970). O padrão mais recente é o SQL:2023 (ISO,).

Na década de 70, a IBM iniciou o desenvolvimento do SQL, quando Chamberlin e Boyce criaram o SEQUEL para o protótipo System R (CHAMBERLIN; BOYCE, 1974). Por questões de marca, o nome foi reduzido para SQL, que se tornou padrão ANSI (1986) e ISO (1987). Desde então, revisões como o SQL:2016 adicionaram suporte nativo a JSON (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2016), e o SQL:2023 incorporou recursos para grafos, mostrando como a linguagem continua a se adaptar a cenários analíticos modernos.

Nos últimos anos, o SQL consolidou-se como ferramenta essencial para Ciência de Dados, especialmente no pré-processamento e análise exploratória. O PostgreSQL estendeu suas capacidades nativas para além do armazenamento transacional tradicional, implementando construções analíticas avançadas como expressões de tabela comuns (*Common Table Expressions*) e funções de janela (*Window Functions*) conforme especificadas nos padrões ANSI/ISO (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025). Essas evoluções tornaram viável a execução de pipelines analíticos robustos diretamente

sobre o núcleo de armazenamento do SGBD, mitigando a necessidade de movimentação de grandes volumes de dados para ambientes de execução externos.

2.3 Sistemas Gerenciadores de Banco de Dados Relacionais

Os SGBDs são softwares que permitem criar, organizar e manipular dados de forma estruturada, baseando-se no modelo relacional de Edgar Codd (CODD, 1970). Eles garantem integridade, segurança e acesso concorrente por meio das propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade). A linguagem SQL é o principal meio de interação com esses sistemas.

Historicamente, os SGBDs sucederam os modelos hierárquicos e de rede. Projetos pioneiros como o System R (IBM) e o Ingres (UC Berkeley) demonstraram a viabilidade prática do modelo relacional, dando origem a sistemas comerciais como Oracle, DB2 e SQL Server (STONEBRAKER; WONG, 1976). O MySQL (ORACLE CORPORATION, 2025) e o PostgreSQL (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025), ambos com desenvolvimento iniciado na década de 1990, consolidaram-se como alternativas de código aberto que ampliaram significativamente o acesso às tecnologias de banco de dados relacionais.

Stonebraker et al. (STONEBRAKER; WONG, 1976) descrevem como o projeto Ingres implementou o modelo relacional de Codd em um sistema funcional, introduzindo consultas com linguagem de alto nível (QUEL) que inspiraram o SQL. Da mesma forma, a IBM aplicou conceitos semelhantes no System R, resultando em um protótipo que comprovou a eficiência do modelo relacional e abriu caminho para os SGBDs comerciais. Esses exemplos mostram como a pesquisa acadêmica e corporativa deu origem à base tecnológica que sustenta os SGBDs modernos.

Nos últimos anos, os SGBDs relacionais evoluíram para lidar com os desafios impostos pelo *Big Data* e pela diversidade de formatos de dados. Embora o modelo relacional tradicional seja baseado em esquemas rígidos e fortemente estruturados, sistemas modernos passaram a incorporar funcionalidades historicamente associadas aos bancos NoSQL, como o suporte nativo a JSON e XML introduzido no padrão SQL:2016 (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2016). Além disso, a computação em nuvem impulsionou o modelo de banco relacionais como serviço (*Database as a Service*, DBaaS), integrando de forma elástica recursos de análise e processos de ETL (*Extract, Transform, Load*). O PostgreSQL é um exemplo expressivo dessa evolução, destacando-se por sua capacidade de otimização de índices e integração nativa com fluxos analíticos complexos em larga escala (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025; STONEBRAKER, 2015).

2.4 Ciência de Dados

A Ciência de Dados é uma área interdisciplinar que combina Estatística, Ciência da Computação e conhecimento de domínio para extrair conhecimento de dados estruturados e não estruturados. Seu objetivo é apoiar decisões, prever tendências e resolver problemas complexos (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Diferencia-se da análise de dados tradicional ao incorporar métodos preditivos e fluxos complexos de engenharia de atributos.

Embora o termo tenha sido utilizado desde a década de 60, a consolidação da área ocorreu a partir dos anos 2000, com a explosão do *Big Data* e a popularização de linguagens interpretadas de script. Trabalhos como os de Donoho (2017) destacam essa evolução, enquanto Patil e Mason (2015) ajudaram a popularizar a profissão de cientista de dados (DONOHO, 2017; PATIL; MASON, 2012).

Avanços recentes incluem o processamento de grandes volumes em tempo real e a descentralização do cômputo, permitindo aplicações práticas que vão desde a detecção de anomalias financeiras até o mapeamento preditivo e monitoramento de surtos epidemiológicos de relevância global (GOODFELLOW; BENGIO; COURVILLE, 2016).

Donoho (DONOHO, 2017) discute como a evolução das bibliotecas de código aberto transformou a prática da Ciência de Dados, permitindo o gerenciamento de volumes em escalas massivas. Patil e Mason (PATIL; MASON, 2012) exemplificam o impacto da profissão na esfera de grandes plataformas corporativas, relatando como cientistas de dados aplicaram técnicas de mineração de dados e predição para estruturar serviços e processar fluxos heterogêneos de informação, evidenciando que o coração da disciplina reside na capacidade de transformar dados brutos em conhecimento inteligível e acionável.

2.5 Análise *In-database*

A análise *In-database* consiste em executar algoritmos estatísticos e de processamento analítico diretamente no SGBD, sem a necessidade de extrair os dados para ferramentas externas. O objetivo é eliminar o custo de movimentação de dados através da rede e reduzir a latência de computação, aproveitando a capacidade nativa do SGBD em lidar com grandes volumes de tuplas de forma otimizada e indexada (FAN; GEERTS, 2010).

O conceito ganhou força com o avanço do *Big Data* nos anos 2000, demonstrando a viabilidade de se realizar tarefas pesadas de agregação e transformação na própria camada onde o dado reside, utilizando a linguagem SQL estruturada. Além de acelerar o processamento físico, o paradigma *In-database* possibilita que todas as etapas de um pipeline de Ciência de Dados ocorram dentro do ambiente integrado do SGBD.

Badia (BADIA, 2020) descreve que tais etapas incluem: (i) *data cleaning*, para tratar inconsistências, tipos corrompidos e valores ausentes; (ii) *data wrangling*, que envolve a transformação, normalização e junções de tabelas; e (iii) *analytics*, fase em que são aplicadas as funções analíticas para extração de correlações e agrupamentos. Tradicionalmente, cada uma dessas fases era delegada a frameworks externos, mas a arquitetura moderna de SGBDs relacionais como o PostgreSQL demonstra que esses passos podem ser unificados de ponta a ponta na linguagem declarativa SQL, reduzindo custos de integração e garantindo a segurança e a consistência transacional do ecossistema (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025; FAN; GEERTS, 2010).

2.6 Consultas SQL Avançadas

As consultas SQL avançadas expandem a capacidade expressiva da linguagem para além das operações de projeção e filtragem simples, tornando possível a implementação de lógicas complexas de preparação e análise de dados diretamente no banco de dados. Três recursos centrais que fundamentam essa abordagem são as Expressões de Tabela Comuns (CTEs), as Funções de Janela (*Window Functions*) e os mecanismos de Agregação estruturada (INFORMATION. . . , 2003; POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025).

A fim de demonstrar a aplicabilidade prática e a sintaxe operacional desses recursos analíticos no contexto real deste trabalho, as subseções subsequentes apresentam exemplos dirigidos baseados no esquema relacional da visão materializada `fato_dengue`, indicando como cada funcionalidade foi empregada para estruturar o pipeline epidemiológico.

2.6.1 *Common Table Expressions*

De acordo com a documentação oficial do PostgreSQL, uma expressão de tabela comum (CTE) atua como uma subconsulta nomeada temporária que existe exclusivamente no escopo de execução de uma instrução principal (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025). Formalizadas a partir do padrão SQL:2003, as CTEs funcionam como "tabelas virtuais"que permitem quebrar consultas complexas em blocos lógicos modulares e legíveis, substituindo subconsultas aninhadas de difícil manutenção e permitindo, inclusive, a execução de estruturas recursivas.

O exemplo a seguir demonstra o uso de CTEs para isolar e segmentar um subconjunto de notificações do ano de 2025 por unidade federativa, permitindo cruzar e comparar a média de ocorrências de cada estado com a média nacional calculada de forma centralizada:

Listing 2.1 – Exemplo de consulta SQL utilizando Expressão de Tabela Comum (CTE).

```
WITH base_notificacoes AS (
    SELECT id_municip,
           (SELECT u.nome_uf FROM municipio m
            JOIN unidade_federativa u ON u.cod_uf = m.cod_uf
            WHERE m.id_municip = fd.id_municip) AS estado
    FROM fato_dengue fd
    WHERE mes >= '2025-01-01' AND mes < '2026-01-01'
),
contagem_por_uf AS (
    SELECT estado,
           COUNT(*) AS total_casos_uf
    FROM base_notificacoes
    GROUP BY estado
)
SELECT
    estado,
    total_casos_uf,
    (SELECT AVG(total_casos_uf)::numeric(12,2)
        FROM contagem_por_uf) AS media_nacional_uf
FROM contagem_por_uf
ORDER BY total_casos_uf DESC;
```

2.6.2 *Window Functions*

As funções de janela representam um recurso analítico avançado, padronizado no SQL:2003, que realiza cálculos sobre um conjunto de linhas que possuem algum relacionamento lógico com a linha atual (INFORMATION. . . , 2003). O diferencial pedagógico e técnico das *Window Functions*, conforme documentado no tutorial oficial do PostgreSQL, é que elas não colapsam as linhas do resultado em um único grupo como faz a cláusula `GROUP BY` tradicional; em vez disso, as linhas mantêm suas identidades individuais enquanto a função acessa o quadro (*frame*) da janela definido pela cláusula `OVER()` (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025).

A consulta abaixo exemplifica a utilização da função de janela `RANK()` combinada ao modificador `PARTITION BY`. A instrução realiza o ranqueamento interno dos municípios com maior volume absoluto de casos de dengue, reiniciando a contagem de posições de forma automática a cada mudança de estado (UF):

Listing 2.2 – Exemplo de consulta SQL utilizando Funções de Janela (*Window Functions*) com a cláusula RANK().

```
SELECT
    uf.nome_uf AS estado,
    m.nome_municip AS municipio,
    COUNT(*) AS total_casos,
    RANK() OVER (
        PARTITION BY uf.nome_uf
        ORDER BY COUNT(*) DESC
    ) AS ranking_no_estado
FROM fato_dengue fd
JOIN municipio m ON m.id_municip = fd.id_municip
JOIN unidade_federativa uf ON uf.cod_uf = m.cod_uf
GROUP BY uf.nome_uf, m.nome_municip
ORDER BY estado, ranking_no_estado;
```

2.6.3 Agregações Avançadas e Tendências Temporais

As funções de agregação clássicas (`COUNT`, `SUM`, `AVG`) realizam a vetorização e a sumarização de grandes volumes de dados brutos a partir de um critério de quebra estrutural. Quando associadas às cláusulas de ordenação temporal e restrição de linhas (`ROWS BETWEEN`), as agregações expandem-se para calcular métricas estatísticas dinâmicas e cumulativas, como médias móveis, sem a necessidade de loops ou processamentos imperativos externos (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025).

O exemplo a seguir ilustra o emprego de agregações avançadas e funções de janela temporais sobre os dados estruturados de 2025. A consulta realiza a contagem absoluta de notificações por mês e calcula, de maneira puramente *In-database*, a média móvel trimestral de casos da doença, olhando para o registro corrente e para as duas partições cronológicas imediatamente anteriores:

Listing 2.3 – Exemplo de consulta SQL utilizando agregações avançadas e cálculo de média móvel trimestral.

```
WITH mensal AS (
    SELECT DATE_TRUNC('month', mes)::date AS periodo_mes,
           COUNT(*) AS casos_no_mes
    FROM fato_dengue
    WHERE mes >= '2025-01-01' AND mes < '2026-01-01'
    GROUP BY DATE_TRUNC('month', mes)
)
SELECT
    periodo_mes,
    casos_no_mes,
    AVG(casos_no_mes) OVER (
        ORDER BY periodo_mes
        ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
    )::numeric(12,2) AS media_movel_trimestral
FROM mensal
ORDER BY periodo_mes;
```

Esses exemplos práticos demonstram como, no contexto unificado desta pesquisa, as CTEs organizaram a modularidade das subconsultas, as *Window Functions* permitiram ranqueamentos comparativos geográficos intrasetoriais e as agregações estruturaram a extração de tendências de séries temporais, operando em estrita conformidade com os padrões formais ANSI e a arquitetura relacional do PostgreSQL (INFORMATION..., 2003; POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025).

2.7 Considerações Finais do Capítulo

A fundamentação teórica exposta neste capítulo permitiu delinear o embasamento conceitual necessário para a execução do trabalho. A articulação entre o modelo relacional, os recursos analíticos avançados da linguagem SQL e os princípios da análise *In-database* forneceu os subsídios técnicos que sustentam a construção e a otimização da infraestrutura relacional desenvolvida.

3 TRABALHOS CORRELATOS

O presente capítulo apresenta a especificação e a análise comparativa dos trabalhos técnicos que fundamentaram o design do pipeline construído nesta pesquisa. A seleção das arquiteturas e estudos correlatos foi norteada por três eixos de engenharia de dados: (i) processamento de consultas analíticas *In-database*, voltado à execução de lógicas na camada de armazenamento; (ii) expressividade e custo computacional de funções de janela (*Window Functions*) nativas; e (iii) tolerância a falhas e estratégias de isolamento em processos de ingestão de grandes volumes.

O mapeamento técnico utilizou como critério de busca indexadores especializados em sistemas de computação, concentrando as consultas na *ACM Digital Library* e nos anais de conferências de engenharia de dados, tais como o *International Conference on Management of Data* (SIGMOD) e o *Proceedings of the VLDB Endowment* (PVLDB). Foram incluídos artigos com validação empírica de performance de algoritmos sobre o motor relacional e estudos que propusessem frameworks de execução baseados estritamente em dialetos SQL padronizados (ANSI/ISO). Como critério de exclusão, descartaram-se artigos focados unicamente em transações de baixa latência (OLTP) ou que dependessem de acoplamento com ambientes externos de processamento imperativo, garantindo que as arquiteturas revisadas servissem de insumo direto para o modelo construído no PostgreSQL. A descrição dos 5 trabalhos correlatos selecionados é apresentada a seguir em ordem cronológica.

3.1 Efficient Processing of Window Functions in Analytical SQL Queries (2015)

. O artigo (LEIS et al., 2015) investigou o processamento eficiente e o poder expressivo das funções de janela analíticas dentro de consultas SQL. O objetivo central do trabalho foi demonstrar como as construções baseadas na cláusula `OVER` permitem expressar de forma matemática, elegante e computacionalmente otimizada tarefas complexas como análises de séries temporais, partições geográficas, rankings e somas cumulativas diretamente no SGBD.

Do ponto de vista metodológico, os autores mapearam os algoritmos internos de ordenação e execução das funções de janela nos motores de busca relacionais. Eles argumentaram diretamente que a tentativa de realizar essas tarefas em pipelines utilizando o padrão SQL antigo (SQL-92 sem janelas) exige junções excessivamente caras e ineficientes,

enquanto a migração do volume bruto de tuplas para frameworks imperativos externos gera gargalos severos de serialização e processamento na rede.

Os resultados obtidos comprovaram que a execução nativa de funções analíticas de janela no banco de dados reduz drasticamente o custo computacional, aproveitando o paralelismo e os otimizadores lógicos do próprio sistema de armazenamento. O estudo evidenciou que estender a aplicação de consultas declarativas com partições nativas elimina a redundância arquitetural e o débito técnico de pipelines híbridos de dados.

A relação deste estudo com o presente trabalho é de fundamentação direta: o artigo validou rigorosamente a abordagem adotada nesta monografia ao lidar com o processamento das fichas do SINAN. A defesa dos autores quanto à elegância e eficiência das funções de janela ampara diretamente as consultas estruturadas neste TCC para computar a evolução mensal acumulada da dengue, bem como o ranqueamento dos municípios por taxas de incidência proporcional, chancelando a viabilidade técnica do PostgreSQL no suporte de ponta a ponta a pipelines de Ciência de Dados.

3.2 50 Years of Data Science (2017)

O artigo (DONOHO, 2017) fez uma análise histórica e conceitual da Ciência de Dados, discutindo suas origens, evolução e perspectivas de longo prazo. O objetivo principal foi compreender como a área se consolidou como disciplina distinta da Estatística clássica e da Ciência da Computação, destacando os fatores sociais, tecnológicos e acadêmicos que contribuíram para sua ascensão.

Metodologicamente, o autor adotou uma abordagem reflexiva e de revisão crítica, analisando publicações, tendências de pesquisa e práticas emergentes ao longo de cinco décadas. Ele identificou marcos importantes, como o crescimento da Estatística Computacional, a explosão de dados digitais e a popularização de linguagens de análise. Donoho também argumentou sobre a necessidade de diferenciar a Ciência de Dados da simples aplicação estatística rudimentar, ressaltando sua natureza interdisciplinar e orientada por problemas práticos.

Entre os resultados de sua análise, o artigo propôs que a Ciência de Dados deve ser reconhecida como uma área autônoma, com seus próprios métodos, desafios e currículos de formação, destacando a importância imperativa da reprodutibilidade, da transparência e da documentação aberta dos procedimentos de processamento. Essas ideias moldaram a forma como a disciplina é consolidada atualmente em ambientes acadêmicos e profissionais.

A relação com este TCC é fundamental: ao contextualizar a evolução histórica da área, o trabalho de Donoho conferiu legitimidade à proposta de explorar o SQL como

linguagem de Análise Científica. Se a Ciência de Dados é uma prática que une computação e problemas reais de domínio, demonstrar que Sistemas de Bancos de Dados Relacionais modernos servem como plataformas completas e reprodutíveis para engenharia e análise epidemiológica está em perfeita sintonia com a visão de expansão da área defendida pelo autor.

3.3 A Unified System for Data Analytics and In Situ Query Processing (2021)

O artigo (WATSON et al., 2021) propôs um sistema unificado que combina Análise de dados e *In Situ Query Processing*, isto é, a capacidade de realizar análises complexas diretamente sobre os dados onde eles estão armazenados, sem a necessidade de exportação para outras ferramentas. O objetivo central do trabalho foi diminuir o custo de movimentação de dados e a redundância entre Sistemas de Banco de Dados e Plataformas Analíticas, permitindo que ambos os tipos de processamento compartilhassem a mesma infraestrutura.

Do ponto de vista metodológico, os autores integraram técnicas tradicionais de otimização de consultas com novos operadores analíticos implementados dentro do próprio mecanismo de consultas. Dessa forma, funções estatísticas e agregações avançadas foram executadas como parte natural do plano de execução SQL. A proposta também envolveu um arcabouço de extensibilidade que facilitou a adição de novos operadores, sem quebrar a compatibilidade com o SQL padrão.

Os resultados apresentados demonstraram ganhos significativos de desempenho em cenários de volumes massivos de dados, especialmente quando comparados a arquiteturas baseadas em ETL (*Extract, Transform, Load*) que transferem dados para ferramentas externas de análise. Além da performance, o trabalho mostrou vantagens arquiteturais, como a redução da duplicação de dados, a consistência entre diferentes fases do pipeline e a simplificação do fluxo operacional. Em casos práticos de benchmarks, o sistema alcançou tempos de resposta competitivos, provando a viabilidade de unir consulta e análise em uma mesma camada.

A relação deste estudo com o presente TCC é direta: reforçou a tese de que a Ciência de Dados pode ser realizada de forma eficaz dentro de Sistemas de Gerenciamento de Banco de Dados (SGBD) Relacionais, sem depender exclusivamente de linguagens externas como Python ou R. A integração entre SQL e operações analíticas, defendida pelos autores, sustentou a proposta deste trabalho, demonstrando, através de conjuntos de dados públicos e reprodutíveis da dengue e do IBGE, que operações de exploração e preparação podem ser feitas inteiramente na camada relacional.

3.4 Transforming ML Predictive Pipelines into SQL with MASQ (2021)

(BUONO et al., 2021) apresentaram o *MASQ*, um sistema que converte *pipelines* de processamento e modelos analíticos já estruturados em comandos SQL padrão. O objetivo foi permitir que tanto o pré-processamento dos dados quanto a fase de extração estatística pudessem ser executados diretamente dentro do banco de dados, eliminando a dependência de frameworks imperativos externos durante a fase de consumo analítico.

A proposta funcionou traduzindo cada etapa de transformação categórica e agregação em consultas SQL equivalentes, que puderam ser processadas diretamente pelo próprio otimizador do SGBD. Os resultados apresentados mostraram que essa estratégia manteve a portabilidade entre diferentes motores relacionais, eliminou o custo operacional de transferir dados para fora do sistema de armazenamento e facilitou a integração de consultas complexas em aplicações corporativas que já utilizavam SQL nativo.

A relação com este TCC é evidente: o MASQ demonstrou de forma prática que é perfeitamente possível aplicar técnicas complexas de Ciência de Dados utilizando apenas a linguagem declarativa SQL. O trabalho dos autores reforçou o argumento central desta monografia de que o SGBD relacional não deve ser reduzido a um mero repositório passivo de armazenamento, mas sim aproveitado como o motor central de processamento e inteligência do pipeline.

3.5 Future Trends in SQL Databases and Big Data Analytics (2024)

O trabalho (ISLAM, 2024) discutiu as principais tendências na evolução dos bancos de dados SQL e sua integração com o ecossistema de *Big Data Analytics*. O objetivo foi analisar de que forma o SQL, mesmo após mais de quatro décadas de existência, permanece central no processamento de dados modernos e como sua linguagem e infraestrutura estão se adaptando a novos paradigmas analíticos.

Metodologicamente, o autor realizou uma revisão conceitual e exploratória sobre avanços recentes, destacando tecnologias como bancos de dados distribuídos, processamento vetorizado e o surgimento de capacidades analíticas nativas aprimoradas em SQL. A análise foi complementada com exemplos práticos de como o SQL se integra a arquiteturas modernas de dados, em especial no contexto de plataformas híbridas, onde a fronteira entre dados relacionais e analíticos tende a se diluir.

Os resultados da revisão mostraram que o SQL continuou evoluindo, incorporando funcionalidades como processamento vetorizado e otimizações específicas para dados em

larga escala. Além disso, o estudo apontou que a adoção de SQL em plataformas *cloud-native* reforça seu papel estratégico, pois oferece uma interface unificada para análise, reduzindo a barreira de entrada para cientistas e engenheiros de dados.

A relação deste trabalho com o TCC é significativa: ao mapear tendências e apontar que o SQL consolidou-se como uma linguagem de Ciência de Dados robusta, o artigo de Islam sustentou a relevância deste projeto, que demonstrou, por meio de implementações reais, como operações de engenharia de atributos e agregações podem ser realizadas unicamente com comandos relacionais. Dessa forma, o TCC dialogou com as tendências identificadas, mostrando que a abordagem moderna consiste em expandir a aplicação analítica dentro dos próprios SGBDs.

A Tabela 1 apresenta um comparativo entre os trabalhos relacionados mapeados nesta pesquisa em ordem cronológica, destacando o veículo de publicação, o ano de referência, o foco de investigação, as abordagens metodológicas utilizadas e suas respectivas contribuições conceituais para a validação deste TCC.

Tabela 1 – Comparativo entre trabalhos relacionados à análise de dados *In-database* em ordem cronológica e suas contribuições para o Trabalho de Conclusão de Curso

Trabalho / Referência	**Veículo (ano)**	**Foco / Objetivo**	**Abordagem / Métodos**	**Contribuições principais & relação com o TCC**
Leis et al. (2015) (LEIS et al., 2015)	PVLDB (2015)	Demonstrar a eficiência e a expressividade de funções de janela em consultas analíticas.	Análise de algoritmos internos de ordenação e otimização de partições em sistemas relacionais.	Valida a abordagem de usar *Window Functions* nativas para calcular acumulados móveis e rankings epidemiológicos neste TCC.
Donoho (2017) (DONOHO, 2017)	JCGS (2017)	Revisar 50 anos de evolução da ciência de dados.	Reflexão crítica sobre interdisciplinaridade, reprodutibilidade e princípios abertos.	Conferiu legitimidade acadêmica ao uso do SQL como plataforma científica e reprodutível de análise.
Watson et al. (2021) (WATSON et al., 2021)	PVLDB (2021)	Unificar análise de dados e consultas *in situ*.	Integra operadores analíticos ao otimizador SQL, rodando dentro do plano de consulta.	Demonstrou ganhos de desempenho ao evitar processos de ETL; fundamenta a execução do pipeline diretamente no SGBD.
Del Buono et al. (2021) (BUONO et al., 2021)	SIGMOD (2021)	Transformar pipelines analíticos em SQL padrão.	Tradução automática de lógicas de transformação e agregação categórica em consultas SQL.	Reforçou a viabilidade de se arquitetar pipelines analíticos robustos sem dependência de ambientes externos.
Islam (2024) (ISLAM, 2024)	SSRN (2024)	Discutir tendências futuras de SQL em cenários de Big Data e nuvem.	Revisão conceitual de processamento analítico nativo, vetorização e plataformas híbridas.	Provou que o SQL segue evoluindo para tarefas analíticas avançadas; sustenta a relevância da monografia.

3.6 Considerações Finais do Capítulo

A revisão dos trabalhos correlatos evidenciou o estado da arte e a evolução recente do processamento analítico *In-database*. A análise comparativa chancelou as escolhas arquiteturais deste TCC, comprovando que o uso de recursos nativos do SGBD, como *Window Functions* e visões materializadas, alinha-se às melhores práticas para a execução eficiente de pipelines de dados.

4 MÉTODO DE TRABALHO

Este capítulo apresenta o detalhamento das etapas do método de trabalho que nortearam a estruturação e a implementação do ambiente analítico objeto deste estudo. A definição de um método de trabalho rigoroso é essencial não apenas para garantir a reprodutibilidade técnica, mas também para validar a utilidade da infraestrutura de dados e do pipeline analítico em seu contexto de aplicação. Para a operacionalização de todas as etapas de engenharia e Ciência de Dados deste trabalho, adotou-se o Sistema de Gerenciamento de Banco de Dados (SGBD) relacional PostgreSQL como tecnologia central, servindo como o mecanismo principal de armazenamento, tratamento, modelagem e otimização das bases de dados públicas utilizadas.

No contexto deste trabalho, a abordagem sociotécnica fundamenta-se na compreensão de que o desenvolvimento de artefatos de computação deve ser analisado a partir de cenários práticos e aplicados. Para tanto, este estudo de caso concentra-se na viabilidade e na eficácia da utilização da linguagem SQL e do SGBD PostgreSQL para a execução de tarefas complexas de Ciência de Dados. A investigação busca demonstrar, por meio de um ambiente controlado e utilizando dados públicos reais de notificações de dengue do Ministério da Saúde e indicadores demográficos do IBGE, como o pipeline de engenharia e modelagem relacional pode estruturar volumes massivos de dados brutos. Assim, o foco central reside na validação técnica do SQL como uma ferramenta robusta e acessível para a transformação de dados em conhecimento, evidenciando o potencial analítico da tecnologia a partir de uma base de relevância social.

A ideia geral do trabalho consiste no estabelecimento de um fluxo fim a fim de inteligência de dados integralmente executado dentro do ambiente do SGBD PostgreSQL. O ciclo de vida do dado inicia-se com a ingestão em massa dos arquivos planos brutos (SINAN, IBGE e CBO) para tabelas de isolamento, configurando uma *Staging Area*. A partir desse ponto, o ecossistema utiliza scripts SQL para a higienização de chaves textuais e aplicação de tipagem estrita. Na sequência, o dado limpo é submetido ao processo de normalização lógica, decompondo a estrutura desnormalizada original em tabelas que atendem à Terceira Forma Normal (3FN), incluindo relacionamentos muitos-para-muitos para o mapeamento eficiente de sintomas clínicos. Por fim, a infraestrutura é otimizada por meio de índices estruturados em B-Tree e consolidada em uma *Materialized View* analítica, que serve como a base de alta performance para a extração dos indicadores estatísticos e epidemiológicos.

A fim de detalhar a execução prática desse processo, as seções subsequentes descre-

vem o pipeline técnico de forma sequencial. Inicialmente, na Seção 4.1, são apresentadas as especificidades de cada fonte de dados, seus metadados volumétricos e os critérios de coleta. A Seção 4.2 detalha os procedimentos de ingestão e a lógica de limpeza na *Staging Area*. Por fim, a Seção 4.3 expõe a modelagem lógica formal resultante da normalização, bem como as estratégias de indexação e a criação da visão materializada analítica no PostgreSQL.

4.1 Coleta e Fontes de Dados

A escolha dessa temática justifica-se pela magnitude e pela persistência histórica da patologia como um dos maiores desafios da saúde pública brasileira. Historicamente, a dengue é classificada como a segunda mais importante doença transmitida por vetor no mundo (BARRETO; TEIXEIRA, 2008). No Brasil, a série histórica evidencia ciclos epidêmicos críticos e crescentes: em 1998, o país registrou mais de 700 mil casos e, em 2002, esse volume atingiu aproximadamente 800 mil notificações, representando quase 80% das ocorrências de todo o continente americano na época (BARRETO; TEIXEIRA, 2008).

Ainda na atualidade, a dengue permanece como um problema de saúde pública de impacto extremo e crescente no país. Conforme destaca D'Alessandro (D'ALESSANDRO, 2025), longe de ser um desafio sob controle, a arbovirose atingiu patamares sem precedentes recentemente, sendo o ano de 2024 consolidado oficialmente como o período da maior epidemia de dengue de toda a história do Brasil. Esse cenário contemporâneo de hiperendemicidade reforça a necessidade de infraestruturas de dados robustas, uma vez que o vírus e seu vetor circulam hoje em grande parte do território nacional (BARRETO; TEIXEIRA, 2008). Portanto, a análise dos dados reais referentes ao ano de 2025, realizada neste trabalho, permite investigar a dinâmica epidemiológica no período imediatamente posterior ao seu maior pico histórico, fornecendo ideias sobre a continuidade da transmissão sob uma ótica de Ciência de Dados.

Para a viabilização deste trabalho, a coleta foi estruturada a partir de quatro pilares fundamentais de dados governamentais. É importante ressaltar que a utilização das três bases auxiliares (geográfica, demográfica e ocupacional) é tecnicamente indispensável devido à estrutura do conjunto de dados principal. Os registros do SINAN priorizam o uso de códigos identificadores padronizados para otimizar o armazenamento; por exemplo, a tabela de notificações não contém o nome por extenso do município onde ocorreu o caso, mas sim o seu código numérico oficial do IBGE. Dessa forma, a aplicação do modelo relacional permite realizar o cruzamento (*join*) entre essas tabelas por meio de códigos unificados, transformando chaves numéricas em informações inteligíveis para a análise.

As fontes utilizadas foram:

1. **Notificações Epidemiológicas (SINAN/MS):** Fonte primária contendo os registros individuais de dengue do ano de 2025. O arquivo `DENGBR25.csv` e o seu respectivo `Dicionário de dados - Dengue.pdf` foram obtidos por meio do Portal de Dados Abertos da Saúde, mantido pelo Ministério da Saúde (BRASIL. Ministério da Saúde, 2025).

2. **Estrutura Territorial (IBGE):** Dados oficiais da malha municipal e da hierarquia regional brasileira, extraídos do arquivo `RELATORIO_DTB_BRASIL_2024_MUNICIPIOS.xls`. Esta base está disponível publicamente no portal do Instituto Brasileiro de Geografia e Estatística, especificamente na seção da Divisão Territorial do Brasil (IBGE. Instituto Brasileiro de Geografia e Estatística, 2024).

3. **Estimativas Populacionais (IBGE):** Documento contendo as projeções populacionais para os municípios brasileiros relativas ao ano de 2025 (`POP2025_20260113.xls`). Os dados foram coletados diretamente do portal do IBGE, na seção dedicada a estatísticas de população (IBGE TITLE = Projeções e Estimativas da População dos Municípios,).

4. **Classificação Brasileira de Ocupações (CBO):** Base de dados contendo os perfis e títulos profissionais (`cbo2002-ocupacao.csv`). O conjunto de dados foi extraído do Portal de Dados Abertos do Governo Federal, sob a gestão e disponibilização originária do Ministério do Trabalho e Emprego (BRASIL. Ministério do Trabalho e Emprego, 2025).

4.1.1 Perfil Quantitativo e Estrutura das Fontes

Para detalhar a volumetria e as especificidades técnicas das bases coletadas, a Tabela 2 apresenta os metadados quantitativos de cada fonte de dados carregada no SGBD, especificando o formato original do arquivo, a quantidade total de atributos (colunas) e o volume de tuplas (linhas) importadas.

Tabela 2 – Perfil quantitativo e formato das fontes de dados importadas.

Fonte de Dados	Formato Original	Nº de Atributos	Nº de Tuplas (Linhas)
Notificações (SINAN)	CSV	121	1.655.637
Estrutura Territorial (IBGE)	XLS (convertido para CSV)	15	5.570
Estimativas Populacionais (IBGE)	XLS (convertido para CSV)	5	5.570
Ocupações (CBO)	CSV	2	2.622

Fonte: Elaborada pelo autor (2026).

A partir do universo de atributos disponibilizados pelas fontes primárias, selecionaram-se as variáveis críticas para o mapeamento relacional e para a execução do escopo

analítico. A Tabela 3 descreve os principais atributos extraídos, indicando o tipo de dado mapeado no PostgreSQL e a sua respectiva descrição lógica.

Tabela 3 – Dicionário simplificado dos principais atributos utilizados.

Atributo	**Tipo no SGBD**	**Descrição / Função no Modelo**
`id_notific`	`INTEGER (PK)`	Identificador único da notificação epidemiológica.
`dt_notific`	`DATE`	Data em que a notificação foi registrada.
`co_municip_n`	`VARCHAR(6)`	Código IBGE do município de notificação (chave de junção).
`nu_idade_n`	`INTEGER`	Idade do paciente codificada no momento da triagem.
`cs_sexo`	`VARCHAR(1)`	Gênero do indivíduo notificado (M, F, I).
`id_ocupa_n`	`VARCHAR(6)`	Código da ocupação do paciente (vínculo com a CBO).
`cod_municipio`	`VARCHAR(6) (PK)`	Código identificador oficial do município pelo IBGE.
`nome_municipio`	`VARCHAR(100)`	Nome por extenso do município brasileiro.
`populacao`	`INTEGER`	Volume de habitantes estimados para o ano de 2025.
`cbo_codigo`	`VARCHAR(6) (PK)`	Código identificador da ocupação na tabela CBO.
`cbo_descricao`	`VARCHAR(200)`	Título profissional correspondente à ocupação.

Fonte: Elaborada pelo autor (2026).

A definição clara das fontes de dados estabelece a matéria-prima do estudo, mas a condução metodológica exige o desenho detalhado do pipeline de engenharia e das metas analíticas traçadas. Para garantir que o SGBD operasse de forma eficiente, o fluxo de trabalho foi estruturado em etapas sequenciais, mapeando o ciclo de vida do dado desde a ingestão até a disponibilização para análise, conforme ilustrado na Figura 1.

4.1.2 Etapas de Processamento no PostgreSQL

Como mapeado na arquitetura da Figura 1, o ciclo de processamento dentro do PostgreSQL foi dividido em quatro macroetapas:

- **Ingestão de Dados Brutos:** Carregamento em massa dos arquivos CSV utilizando a instrução `COPY` para isolar o ambiente operacional.
- **Staging e Saneamento:** Execução de scripts SQL voltados à padronização de tipos de dados, tratamento de registros nulos e limpeza de strings por meio de expressões regulares nas tabelas temporárias.
- **Modelagem e Persistência Definitiva:** Distribuição dos registros higienizados nas tabelas do modelo relacional, efetuando a decomposição em tabelas menores para atingir a Terceira Forma Normal (3FN) e amarrando o ecossistema com chaves estrangeiras.
- **Otimização e Camada Analítica:** Aplicação de índices estruturados e consolidação dos dados agregados na *Materialized View* para servir de base às consultas complexas.

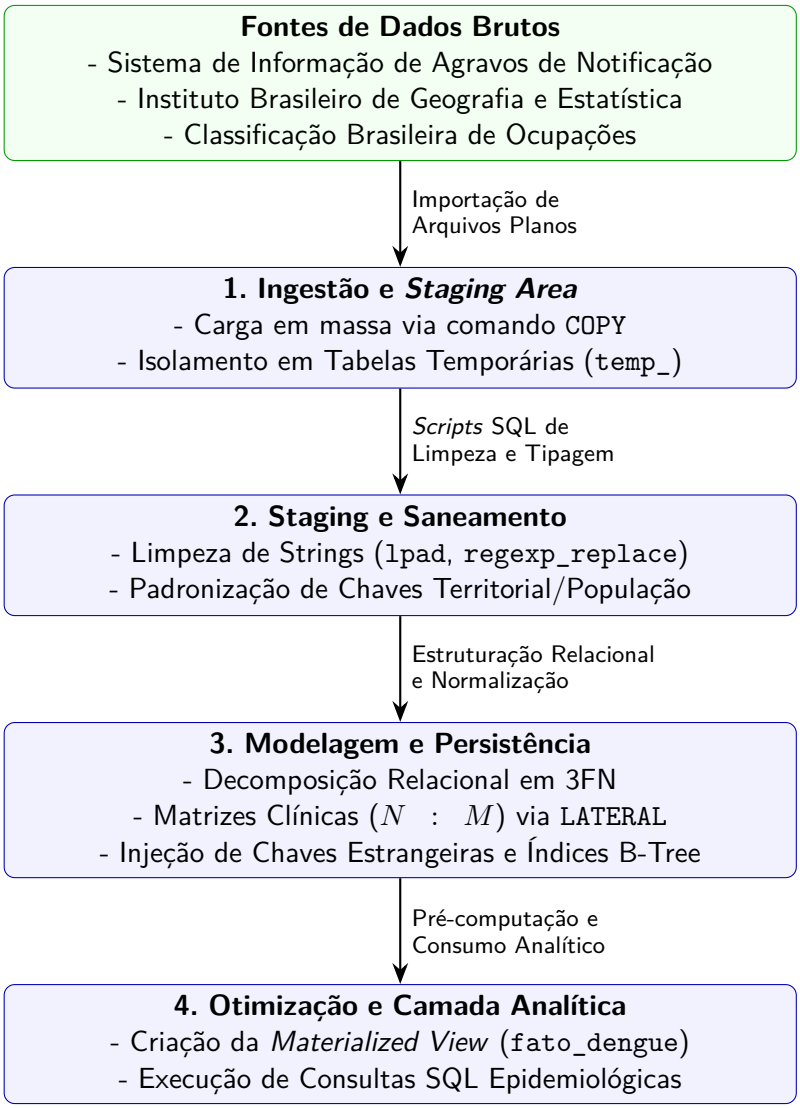

Figura 1 – Pipeline de Engenharia de Dados e Processamento Analítico no PostgreSQL.
Fonte: Elaborada pelo autor (2026).

4.1.3 Definição do Escopo Analítico em SQL

Com a infraestrutura de dados devidamente estruturada, delimitaram-se as análises de Ciência de Dados a serem implementadas nativamente por meio de consultas SQL. O objetivo do ecossistema foi responder a quatro eixos fundamentais de investigação epidemiológica:

- **Análise Geoespacial de Incidência:** Formulação de consultas com o cruzamento analítico (*join*) entre a população estimada do IBGE e o total de ocorrências por localidade, normalizando o indicador para identificar municípios em estado crítico de surto por 100 mil habitantes.
- **Análise de Evolução Temporal:** Agregação cronológica das notificações por períodos mensais para mapear o comportamento sazonal da patologia ao longo do ano de 2025.

- **Estruturação do Perfil Demográfico:** Segmentação dos indivíduos afetados a partir de recortes por gênero e faixas etárias específicas, utilizando técnicas de *feature engineering* em SQL para agrupar as idades registradas no momento da triagem.
- **Mapeamento do Perfil Ocupacional e Laboral:** Categorização e distribuição das notificações conforme os Grandes Grupos da Classificação Brasileira de Ocupações (CBO), aplicando transformações em SQL para avaliar o impacto da arbovirose sobre os diferentes setores da força de trabalho ativa.
- **Mapeamento de Correlação Linear:** Aplicação de funções estatísticas nativas do SGBD para avaliar a relação matemática de dependência entre as estimativas de densidade populacional e o volume absoluto de notificações geradas nas zonas urbanas.

4.2 Processamento e Limpeza

O processo de extração, transformação e carga (*Extract, Transform, Load* – ETL) constituiu a etapa mais crítica para assegurar a viabilidade e a integridade deste projeto. Os dados brutos provenientes de sistemas governamentais (SINAN e IBGE) frequentemente apresentam alto índice de ruído, formatações inconsistentes, ausência de padronização em chaves de ligação e dados esparsos. Para mitigar essas limitações, o pipeline de engenharia de dados foi estruturado em três fases consecutivas dentro do SGBD PostgreSQL: isolamento em área de transição, saneamento via expressões regulares e decomposição relacional.

4.2.1 Estratégia de Staging Area

Para evitar a contaminação do esquema definitivo do banco de dados com registros malformados ou tipos de dados corrompidos, adotou-se uma estratégia de *Staging Area*. Essa abordagem fundamenta-se na necessidade de isolar os processos de extração, limpeza e transformação de dados heterogêneos em um ambiente de retaguarda, mitigando o impacto direto nos sistemas operacionais de origem e garantindo que apenas dados consistentes sejam consolidados na Ciência de Dados (CHAUDHURI; DAYAL, 1997). Foram criadas tabelas temporárias de isolamento, identificadas pelo prefixo `temp_` (como `temp_notificacoes_dengue`, `temp_municipio` e `temp_municipio_populacao`), cujos atributos foram mapeados inicialmente com tipos genéricos como `TEXT` ou `VARCHAR`.

Essa abordagem de design de infraestrutura justifica-se por uma restrição operacional do comando `COPY` do PostgreSQL. Caso os arquivos planos (*flat files* CSV) fossem injetados diretamente nas tabelas definitivas, qualquer inconsistência pontual de tipagem ou violação de integridade referencial causaria a rejeição imediata e o aborto de toda a

operação de carga em massa. O isolamento na *Staging Area* garantiu que 100% dos dados brutos fossem internalizados no SGBD de forma integral e sem interrupções, transferindo a responsabilidade do saneamento e da conversão de tipos para o próprio mecanismo do banco de dados por meio de scripts SQL programáticos.

4.2.2 Padronização e Saneamento de Dados

A fase de transformação concentrou-se na unificação de chaves primárias e estrangeiras e na aplicação de tipagem estrita. Um desafio técnico central residiu na divergência estrutural entre a base de notificações do SINAN e as bases territoriais e demográficas do IBGE. O SINAN utiliza um código identificador de município de 6 dígitos, enquanto o cadastro oficial do IBGE adota o código completo de 7 dígitos.

Para viabilizar os cruzamentos relacionais (*joins*) sem perdas de registros, utilizou-se um conjunto avançado de funções nativas de manipulação de cadeias de caracteres e Expressões Regulares (*POSIX*). O código 4.1 exemplifica a lógica implementada para higienizar e unificar os identificadores populacionais:

Listing 4.1 – Script de saneamento e unificação de códigos do IBGE.

```
INSERT INTO municipio_populacao (cod_municipio_completo,
   data_referencia, populacao_estimada)
SELECT
  lpad(regexp_replace(cod_uf, '\D', '', 'g'), 2, '0')
  ||
  lpad(regexp_replace(cod_municipio, '\D', '', 'g'), 5, '0') AS
     cod_municipio_completo,
  DATE '2025-07-01' AS data_referencia,
  NULLIF(regexp_replace(populacao, '\D', '', 'g'), '')::INTEGER
     AS populacao_estimada
FROM temp_municipio_populacao
WHERE NULLIF(regexp_replace(populacao, '\D', '', 'g'), '') IS NOT
    NULL;
```

A engenharia por trás desse script baseia-se em três pilares justificáveis:

- `regexp_replace(..., '\D', '', 'g')`: Garante o expurgo de qualquer caractere não numérico (como pontos, traços ou espaços), deixando apenas os dígitos puros.
- `lpad(..., comprimento, '0')`: Resolve o problema crônico da perda de zeros à esquerda na conversão de arquivos de texto, garantindo que estados como Alagoas (código 2) sejam padronizados rigorosamente como '02', mantendo a integridade com o padrão do IBGE.

- `NULLIF(..., '')` combinado com `::INTEGER`: Captura campos vazios ou mascarados e os converte em valores `NULL` legítimos antes da coerção de tipo (*type casting*), impedindo que o interpretador SQL dispare exceções de execução e interrompa o pipeline analítico.

4.2.3 Engenharia de Dados e Normalização Relacional

O processo de transformação do conjunto de dados brutos em uma infraestrutura relacional estruturada fundamentou-se na aplicação rigorosa da teoria de normalização de bancos de dados relacionais. O objetivo central dessa etapa foi mitigar anomalias de inserção, atualização e deleção, além de eliminar as redundâncias estruturais e o desperdício de armazenamento físico presentes no arquivo original plano da base de dados do SINAN (`DENGBR25.csv`, descrito na Seção 4.1).

Inicialmente, os dados brutos foram ingeridos em uma única tabela desnormalizada denominada `temp_notificacoes_dengue`. Essa estrutura apresentava o cenário clássico de violação das restrições de integridade relacional, possuindo mais de uma centena de colunas que misturavam, em uma mesma entidade física, dados da ocorrência epidemiológica, informações socioeconômicas do paciente, dados residenciais e resultados laboratoriais.

Sob a ótica da teoria relacional, a tabela temporária continha graves violações às três primeiras formas normais:

- **Primeira Forma Normal (1FN):** Violação de atomicidade em colunas clínicas e de múltiplos exames laboratoriais executados ao longo do tempo para um mesmo paciente, além da presença de atributos compostos misturados na mesma partição física;
- **Segunda Forma Normal (2FN):** Presença de dependências parciais em tabelas de chaves compostas (como o cruzamento demográfico e populacional do IBGE), onde atributos não-chave dependiam apenas de uma fração da chave primária;
- **Terceira Forma Normal (3FN):** Ocorrência massiva de dependências transitivas. Por exemplo, na tabela desnormalizada, o atributo contendo o nome do município dependia diretamente do código do município, que por sua vez dependia do identificador da notificação (`id_notificacao`). Essa transitividade gerava uma redundância crônica, forçando o SGBD a persistir strings idênticas de nomes de cidades por mais de 1,6 milhão de registros, onerando os custos de I/O (*Input/Output*) em disco.

Para sanar essas limitações e transpor o ecossistema para a Terceira Forma Normal (3FN), o registro plano original foi decomposto em 29 entidades lógicas independentes,

organizadas por especialidade funcional e interconectadas via restrições de integridade referencial por chaves primárias (*Primary Keys – PK*) e chaves estrangeiras (*Foreign Keys – FK*). O catálogo descritivo a seguir detalha a especificação da arquitetura relacional completa implementada no banco de dados, particionada por seus respectivos domínios arquiteturais.

Domínio: Malha Territorial e Dados Demográficos (IBGE)

- `unidade_federativa`(cod_uf, nome_uf)
- `regiao_geo_intermediaria`(cod_regiao_geo_intermed, cod_uf, `nome_regiao_geo_intermed`)
  ↪ *FK: cod_uf → unidade_federativa(cod_uf)*
- `regiao_geo_imediata`(cod_regiao_geo_imediata, `cod_regiao_geo_intermed, nome_regiao_geo_imediata`)
  ↪ *FK: cod_regiao_geo_intermed → regiao_geo_intermediaria(cod_regiao_geo_intermed)*
- `municipio`(cod_municipio_completo, cod_uf, cod_regiao_geo_imediata, `cod_municipio, nome_municipio, id_municip`)
  ↪ *FK: cod_uf → unidade_federativa(cod_uf)*
  ↪ *FK: cod_regiao_geo_imediata → regiao_geo_imediata(cod_regiao_geo_imediata)*
- `municipio_populacao`(cod_municipio_completo, data_referencia, `populacao_estimada`)
  ↪ *FK: cod_municipio_completo → municipio(cod_municipio_completo)*

Domínio: Atributos e Tabelas de Domínio

- `ocupacao`(cod_ocupacao, titulo_ocupacao)
- `sexo`(codigo, descricao)
- `tipo_sintoma`(id_tipo_sintoma, codigo, descricao)
- `tipo_comorbidade`(id_tipo_comorbidade, codigo, descricao)
- `tipo_sinal_alarme`(id_tipo_sinal_alarme, codigo, descricao)
- `tipo_criterio_gravidade`(id_tipo_criterio_gravidade, codigo, descricao)
- `tipo_manifestacao_hemorragica`(id_tipo_manifestacao, codigo, descricao)
- `tipo_exame`(id_tipo_exame, codigo, descricao)

Domínio: Núcleo da Notificação Epidemiológica

- `notificacao_dengue`(id_notificacao, id_temp, tp_not, id_agravo, `dt_notific, dt_sin_pri, sem_not, nu_ano, sem_pri, dt_invest, classi_fin, criterio, evolucao, dt_obito, dt_encerra, hospitaliz, doenca_tra, clinic_chik, tp_sistema, nduplic_n, dt_digita, cs_flxret, flxrecebi, migrado_w`)
- `notificacao_pessoa`(id_notificacao, ano_nasc, nu_idade_n, cs_sexo, `cs_gestant, cs_raca, cs_escol_n, id_ocupa_n`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: cs_sexo → sexo(codigo)*
  ↪ *FK: id_ocupa_n → ocupacao(cod_ocupacao)*
- `notificacao_local`(id_notificacao, sg_uf_not, id_municip, id_regiona, `id_unidade`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_municip → municipio(cod_municipio_completo)*
- `notificacao_residencia`(id_notificacao, sg_uf, id_mn_resi, id_rg_resi, `id_pais`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_mn_resi → municipio(cod_municipio_completo)*
- `notificacao_internacao`(id_notificacao, dt_interna, uf_internacao, `municipio_internacao`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
- `notificacao_fonte_infeccao`(id_notificacao, tpautocto, coufinf, `copaisinf, comuninf`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*

Domínio: Matrizes Clínicas Pivoteadas

- `notificacao_sintoma`(id_notificacao, id_tipo_sintoma, valor)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_tipo_sintoma → tipo_sintoma(id_tipo_sintoma)*
- `notificacao_comorbidade`(id_notificacao, id_tipo_comorbidade, valor)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_tipo_comorbidade → tipo_comorbidade(id_tipo_comorbidade)*
- `notificacao_sinal_alarme`(id_notificacao, id_tipo_sinal_alarme, valor)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_tipo_sinal_alarme → tipo_sinal_alarme(id_tipo_sinal_alarme)*

- `notificacao_evento_alarme`(id_notificacao, dt_alarm)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
- `notificacao_criterio_gravidade`(id_notificacao, id_tipo_criterio_gravidade, `valor`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_tipo_criterio_gravidade → tipo_criterio_gravidade(id_tipo_criterio_gravidade)*
- `notificacao_evento_gravidade`(id_notificacao, dt_grav)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
- `notificacao_manifestacao_hemorragica`(id_notificacao, id_tipo_manifestacao, `valor`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_tipo_manifestacao → tipo_manifestacao_hemorragica(id_tipo_manifestacao)*
- `notificacao_hemorragica_resumo`(id_notificacao, mani_hemor, plaq_menor, `con_fhd, complica`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
- `notificacao_exame`(id_exame, id_notificacao, id_tipo_exame, data_coleta, `resultado, sorotipo`)
  ↪ *FK: id_notificacao → notificacao_dengue(id_notificacao)*
  ↪ *FK: id_tipo_exame → tipo_exame(id_tipo_exame)*

Para proporcionar uma visualização clara das entidades e relacionamentos que fundamentam diretamente as consultas de Ciência de Dados desenvolvidas neste estudo, a Figura 2 apresenta o Diagrama de Entidade-Relacionamento resumido. A estrutura destaca o acoplamento entre a visão materializada analítica (`fato_dengue`), as tabelas de domínio socioeconômico e territorial (`notificacao_pessoa`, `sexo`, `ocupacao`, `municipio`) e as estimativas populacionais do IBGE (`municipio_populacao`). Em virtude da elevada densidade de atributos e do expressivo número de entidades da arquitetura completa em Terceira Forma Normal (3FN), o modelo relacional integral contendo todas as 29 tabelas foi disponibilizado em alta resolução no repositório oficial do projeto no GitHub[1].

Um diferencial relevante e homogêneo aplicado a essa modelagem foi a abordagem arquitetural direcionada às variáveis clínicas multidimensionais: sintomas, comorbidades, sinais de alarme, critérios de gravidade e manifestações hemorrágicas. Na tabela plana original, cada uma dessas dezenas de manifestações (como febre, mialgia, diabetes, hipotensão, entre outras) ocupava uma coluna booleana individual estática. Essa modelagem

[1] O modelo relacional completo (DER) em alta resolução está disponível em: <https://github.com/filipegoc/tcc-sql-para-ciencia-de-dados/blob/main/img/modelo_relacional_erd.png>

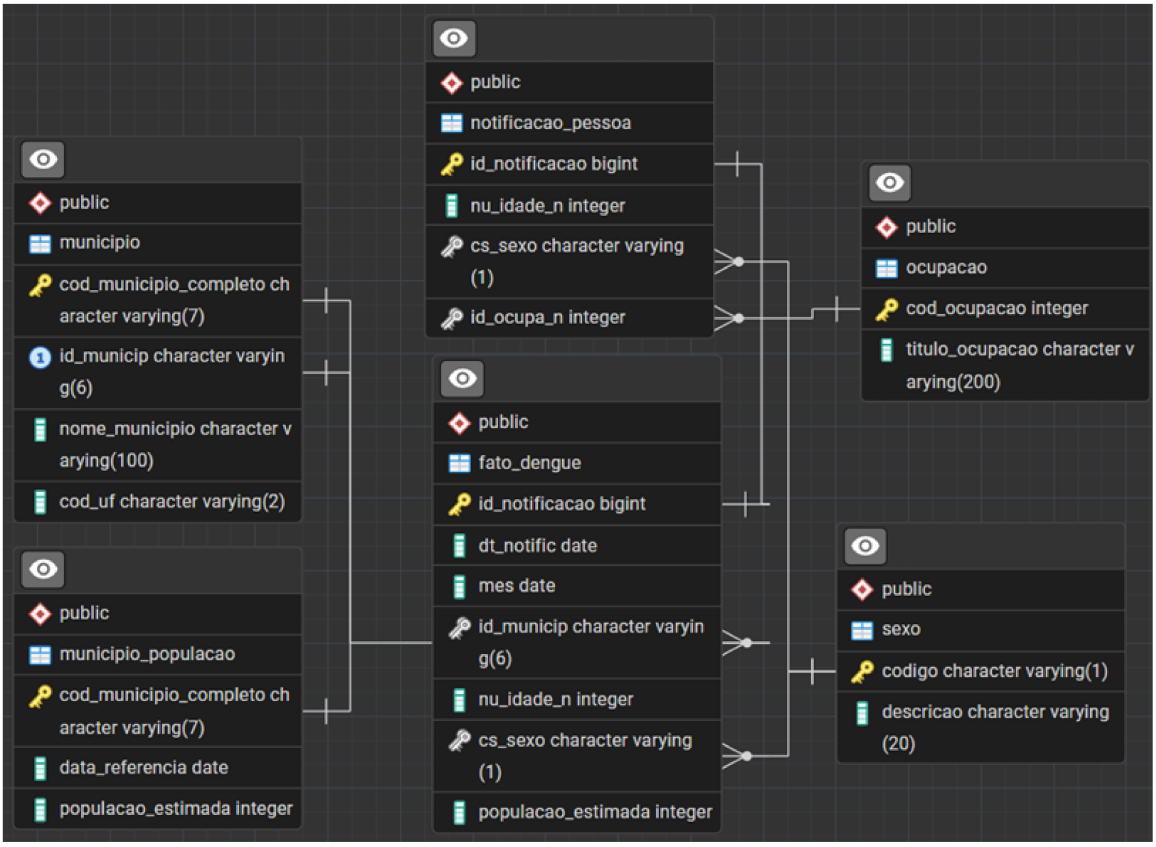

Figura 2 – Diagrama de Entidade-Relacionamento resumido do modelo analítico e suas dimensões.

Fonte: Elaborada pelo autor (2026), via *ERD Tool* do PostgreSQL.

desnormalizada resulta em esquemas extremamente rígidos (*schema rigidity*) que dificultam janelas de manutenção e geram tabelas esparsas ineficientes, onerando o armazenamento com bytes vazios ou nulos (`NULL`).

Para mitigar essa limitação de infraestrutura, implementou-se um padrão de projeto unificado baseado em tabelas de domínio e tabelas associativas de ligação muitos-para-muitos ($N : M$). As colunas originais do arquivo plano foram transpostas programaticamente para registros estruturados em linhas. O Código 4.2 ilustra a implementação desse padrão utilizando a cláusula `CROSS JOIN LATERAL` acoplada à estrutura de `VALUES`, tomando como referência a dispersão de sintomas:

Listing 4.2 – Transposição de colunas para linhas via CROSS JOIN LATERAL no PostgreSQL.

```
INSERT INTO notificacao_sintoma (id_notificacao, id_tipo_sintoma,
    valor)
SELECT n.id_notificacao, ts.id_tipo_sintoma, v.valor
FROM temp_notificacoes_dengue t
JOIN notificacao_dengue n ON n.id_temp = t.id_temp
CROSS JOIN LATERAL (
    VALUES
        ('febre', t.febre),
        ('mialgia', t.mialgia),
        ('cefaleia', t.cefaleia),
        ('exantema', t.exantema),
        ('vomito', t.vomito),
        ('nausea', t.nausea),
        ('dor_costas', t.dor_costas),
        ('conjuntvit', t.conjuntvit),
        ('artrite', t.artrite),
        ('artralgia', t.artralgia),
        ('petequia_n', t.petequia_n),
        ('leucopenia', t.leucopenia),
        ('laco', t.laco),
        ('dor_retro', t.dor_retro)
) AS v(codigo, valor)
JOIN tipo_sintoma ts ON ts.codigo = v.codigo
WHERE v.valor IS NOT NULL;
```

A execução do operador `LATERAL` funciona na engine do PostgreSQL de forma análoga a um laço de repetição *foreach*, iterando sobre a matriz de valores fornecida em disco para cada linha da tabela e expandindo as colunas em sub-registros verticais densos. Este mesmo padrão técnico de engenharia de dados foi replicado sistematicamente para povoar as tabelas `notificacao_comorbidade`, `notificacao_sinal_alarme`, `notificacao_criterio_gravidade` e `notificacao_manifestacao_hemorragica`.

Esta decisão de design confere ao ecossistema analítico duas vantagens principais:

- **Flexibilidade de Esquema (DML vs. DDL):** Caso as autoridades de saúde pública modifiquem as fichas epidemiológicas do SINAN, adicionando novas comorbidades monitoradas ou novos sintomas mutacionais do agravo da dengue, a expansão do sistema dar-se-á via manipulação de dados (*Data Manipulation Language – DML*) através de um comando simples de `INSERT` na tabela de domínio respectiva. Elimina-se a necessidade de travar tabelas de produção volumosas com comandos

estruturais (*Data Definition Language – DDL*) como o `ALTER TABLE`;

- **Eficiência de Indexação e Otimização de I/O:** Consultas analíticas complexas voltadas a calcular coocorrências de sintomas ou cruzamento de comorbidades graves deixam de realizar varreduras sequenciais completas (*Sequential Scans*) em centenas de colunas esparsas e passam a realizar buscas indexadas sobre chaves inteiras compactas, reduzindo drasticamente o consumo de memória e o tempo de resposta do pipeline científico.

4.3 Modelagem e Otimização do Banco de Dados

Após a etapa de limpeza e tratamento dos dados brutos, o projeto avançou para a estruturação definitiva do modelo relacional e a aplicação de técnicas avançadas de otimização física. Esta fase é fundamental para garantir que o SGBD PostgreSQL atue como um motor analítico eficiente para o volume massivo de dados processados, convertendo um esquema puramente transacional em uma infraestrutura resiliente e performática para Ciência de Dados.

4.3.1 Integridade Referencial e Saneamento de Chaves Estrangeiras

Para transformar as tabelas isoladas resultantes do processo de ETL em um modelo relacional coeso e normalizado na Terceira Forma Normal (3FN), estabeleceu-se uma malha estrita de chaves estrangeiras (*Foreign Keys – FK*). Este procedimento garante a consistência dos dados, aplicando restrições de integridade que impedem a existência de registros órfãos e permitem a realização de cruzamentos complexos (*joins*) de forma segura.

Contudo, a injeção imediata de chaves estrangeiras durante a carga massiva via comando `COPY` deterioraria severamente o desempenho do SGBD, uma vez que cada tupla inserida exigiria uma verificação síncrona na tabela referenciada. Por esse motivo, adotou-se uma estratégia de otimização baseada em restrições tardias: as tabelas foram povoadas em sua totalidade e, somente após a conclusão da carga, as restrições foram injetadas via comandos estruturais. As associações implementadas vincularam as tabelas `notificacao_pessoa`, `notificacao_local` e `notificacao_residencia` à tabela central `notificacao_dengue`.

Um desafio técnico central nessa etapa envolveu a variável de ocupação profissional do paciente (`id_ocupa_n`), mapeada a partir das notificações do SINAN. O campo original apresentava um alto índice de ruído de digitação, contendo espaços em branco, caracteres não numéricos e códigos inválidos que violavam a tabela de domínio oficial da Classificação Brasileira de Ocupações (CBO). Para viabilizar a criação da restrição de integridade

referencial sem abortar a integridade do banco de dados, desenvolveu-se uma rotina de higienização por Expressões Regulares (*POSIX*) e filtragem de chaves órfãs, conforme detalhado no Código 4.3:

Listing 4.3 – Script de higienização de chaves e injeção tardia de restrição de integridade.

```
-- 1. Atualiza para NULL tudo que nao for composto apenas por
   digitos
UPDATE notificacao_pessoa
SET id_ocupa_n = NULL
WHERE id_ocupa_n !~ '^[0-9]+$';

-- 2. Converte a coluna para INTEGER tratando espacos em branco
   residual
ALTER TABLE notificacao_pessoa
ALTER COLUMN id_ocupa_n TYPE INTEGER
USING CASE
    WHEN id_ocupa_n ~ '^[0-9]+$' THEN id_ocupa_n::INTEGER
    ELSE NULL
END;

-- 3. Limpa dados invalidos que nao correspondem a tabela CBO
   oficial
UPDATE notificacao_pessoa
SET id_ocupa_n = NULL
WHERE id_ocupa_n IS NOT NULL
AND id_ocupa_n NOT IN (SELECT cod_ocupacao FROM ocupacao);

-- 4. Criacao tardia da restricao de chave estrangeira
ALTER TABLE notificacao_pessoa
ADD CONSTRAINT fk_pessoa_ocupacao
FOREIGN KEY (id_ocupa_n)
REFERENCES ocupacao (cod_ocupacao);
```

Adicionalmente, solucionou-se a divergência crônica de chaves de ligação territoriais entre o SINAN (que adota códigos municipais de 6 dígitos) e o cadastro oficial de municípios do IBGE (baseado em identificadores completos de 7 dígitos). Para alinhar e unificar as fontes geográficas, a tabela definitiva `municipio` foi modificada estruturalmente para comportar o identificador de 6 dígitos truncado, aplicando-se uma restrição de unicidade (`UNIQUE`) para blindar o atributo. O código correspondente foi preenchido por meio da função nativa de extração de subcadeias de caracteres `LEFT(str, n)`, um operador declarativo que extrai os primeiros n caracteres de uma cadeia de texto a partir da extremidade esquerda. Ao aplicar a instrução `LEFT(cod_municipio_completo, 6)`, o

mecanismo do PostgreSQL isolou os seis dígitos iniciais da chave do IBGE, descartando o dígito verificador final e permitindo que as chaves de localidade (`id_municip`) e residência (`id_mn_resi`) das notificações fossem perfeitamente acopladas à malha de dados espaciais do país.

4.3.2 Estratégias de Otimização e Indexação Física

Considerando que o conjunto de dados consolidado ultrapassa a marca de 1,6 milhão de registros básicos de notificação, a recuperação eficiente de informações exige estruturas de acesso físico que minimizem o custo de operações de leitura e escrita em disco (*I/O – Input/Output*). Para tanto, desenhou-se um plano estruturado de indexação baseado na arquitetura de árvores balanceadas de pesquisa (B-Tree).

A escolha pela B-Tree justifica-se por ser a estrutura de indexação mais onipresente, robusta e versátil para sistemas de gerenciamento de bases de dados de larga escala. Segundo Comer (COMER, 1979), a B-tree oferece um desempenho consistente e altamente equilibrado para operações de busca, inserção e deleção, garantindo que o tempo de acesso permaneça eficiente mesmo sob o crescimento exponencial do volume de dados.

No contexto deste estudo, foram instanciados quatro índices físicos estratégicos no SGBD para otimizar as cláusulas de filtragem (`WHERE`), ordenação (`GROUP BY`) e junção (`JOIN`) utilizadas pelo pipeline científico:

- `idx_notificacao_data`: Aplicado sobre o campo `dt_notific` da tabela `notificacao_dengue`, acelerando consultas focadas em recortes temporais e análise de sazonalidade;
- `idx_local_municipio`: Criado sobre o campo `id_municip` da tabela `notificacao_local`, voltado para a segmentação espacial dos casos;
- `idx_pessoa_idade`: Instanciado sobre a coluna `nu_idade_n` da tabela `notificacao_pessoa`, otimizando o agrupamento por faixas demográficas;
- `idx_local_municipio_notif` (Índice Composto): Estruturado de forma multi-coluna cobrindo de forma coordenada os atributos `(id_municip, id_notificacao)` na tabela `notificacao_local`.

A criação do índice composto multi-coluna constitui um diferencial técnico relevante de otimização de consultas. Ao avaliar operações estatísticas de agregação territorial que necessitam contabilizar a quantidade absoluta de casos por município, o otimizador de consultas do PostgreSQL é capaz de interceptar a requisição e executar uma operação de varredura exclusiva no índice (*Index-Only Scan*). Como a árvore B-Tree armazena em seus nós folha os valores indexados combinados de ambos os atributos e o ponteiro físico

correspondente, o motor do banco de dados abdica da necessidade de ler as páginas da tabela física em disco. Essa otimização altera a complexidade computacional de busca de uma escala puramente linear $\mathcal{O}(N)$ para uma escala logarítmica $\mathcal{O}(\log N)$, processando consultas analíticas densas em frações de segundo.

4.3.3 Estatísticas Finais da Infraestrutura Relacional

Após a aplicação do modelo relacional e a execução das rotinas de carga e transformação, consolidou-se a volumetria definitiva do banco de dados no PostgreSQL. Apresentar as estatísticas finais de armazenamento é um procedimento metodológico essencial em engenharia de dados, pois demonstra a escala do ecossistema e valida a necessidade de estratégias de otimização analítica.

A Tabela 4 apresenta o inventário completo dos objetos instanciados no SGBD, detalhando o nome da tabela, sua função lógica no modelo e o volume real de tuplas (linhas) mensuradas após o encerramento do processo de ETL.

Tabela 4 – Estatísticas volumétricas finais das tabelas no PostgreSQL.

Nome da Tabela no SGBD	Função e Contexto Lógico	Total de Tuplas (Linhas)
Entidades Associativas e Clínicas (Muitos-para-Muitos)		
`notificacao_sintoma`	Relacionamento $N:M$ de Sintomas do Paciente	22.285.756
`notificacao_comorbidade`	Relacionamento $N:M$ de Comorbidades do Paciente	11.143.290
`notificacao_exame`	Registro de Exames Laboratoriais Realizados	5.258.001
Core Relacional Normalizado (3FN)		
`notificacao_dengue`	Entidade Central de Notificações Epidemiológicas	1.655.637
`notificacao_local`	Vínculos de Localidade do Agravo	1.655.637
`notificacao_pessoa`	Dados Demográficos e Socioeconômicos do Notificado	1.655.637
`notificacao_residencia`	Local de Residência do Paciente	1.655.627
`notificacao_fonte_infeccao`	Mapeamento de Provável Fonte de Infecção	1.207.785
`notificacao_sinal_alarme`	Sinais de Alarme Clínico Registrados	661.340
`notificacao_criterio_gravidade`	Critérios de Gravidade Observados	566.361
`notificacao_manifestacao_hemorragica`	Manifestações Hemorrágicas Identificadas	311.940
`notificacao_internacao`	Registros de Internação Hospitalar decorrentes	106.628
`notificacao_evento_alarme`	Histórico de Eventos de Alarme Monitorados	38.873
`notificacao_hemorragica_resumo`	Consolidação de Resumos Hemorrágicos	34.660
`notificacao_evento_gravidade`	Detalhamento de Surtos de Gravidade Crítica	3.259
Malha Territorial e Tabelas de Apoio		
`municipio`	Cadastro Oficial de Municípios (IBGE)	5.571
`municipio_populacao`	Estimativa Populacional por Cidade para 2025	5.571
`ocupacao`	Classificação Brasileira de Ocupações (CBO)	2.694
`regiao_geo_imediata`	Divisão Regional Imediata (IBGE)	510
`regiao_geo_intermediaria`	Divisão Regional Intermediária (IBGE)	133
Tabelas de Domínio e Metadados (Lookup Tables)		
`tipo_sintoma`, `tipo_comorbidade`, `sexo`, etc.	Catálogos Estáticos de Domínio de Atributos	< 50 (cada)
Área de Ingestão Provisória (Staging Area)		
`temp_notificacoes_dengue`	Tabela de Carga do Arquivo Flat CSV Primário	1.655.724
`temp_municipio_populacao`	Carga do Relatório de População do IBGE	5.602
`temp_municipio`	Carga do Relatório de Municípios do IBGE	5.571

Fonte: Elaborada pelo autor (2026), com base nas estatísticas nativas do PostgreSQL.

Os dados volumétricos revelam o impacto prático da normalização teórica discutida no método de trabalho. Ao transpor os sintomas e comorbidades que ocupavam colunas esparsas no arquivo plano original para uma estrutura relacional de linhas em tabelas associativas densas, a base de dados expandiu-se de 1,6 milhão de registros iniciais para

uma escala superior a **22 milhões de linhas** na tabela `notificacao_sintoma` e mais de **11 milhões de linhas** em `notificacao_comorbidade`.

Essa característica evidencia a hiperendemicidade e a complexidade clínica da dengue no período de 2025, visto que cada notificação individual gerou, em média, múltiplos registros de sintomas associados ($\approx 13,4$ manifestações por ficha). Do ponto de vista da Computação, essa magnitude massiva de dados valida em definitivo as estratégias de indexação em árvore B (*B-Tree*) aplicadas nas chaves primárias e estrangeiras; a ausência de índices adequados exigiria varreduras sequenciais (*Sequential Scans*) de altíssimo custo computacional, saturando os canais de I/O de disco e inviabilizando a execução das consultas analíticas complexas que fundamentam o mapeamento epidemiológico deste estudo.

4.3.4 Base Analítica e Visão Materializada de Fatos

O ponto de convergência entre a engenharia de dados e a análise científica foi a criação da *Materialized View* denominada `fato_dengue`. É importante ressaltar que, embora a nomenclatura faça alusão ao conceito de "tabela fato"comumente utilizado em arquiteturas de *Data Warehouse* sob o Modelo Estrela (*Star Schema*), o ecossistema deste trabalho mantém-se estritamente fundamentado no **Modelo Relacional Normalizado** em Terceira Forma Normal (3FN). A escolha do prefixo serve ao propósito de identificar o objeto analítico de consumo central, que atua como uma tabela desnormalizada e persistida fisicamente para acelerar as consultas do capítulo subsequente.

Diferente de visões lógicas convencionais, a visão materializada persiste os dados fisicamente em disco, o que representa uma estratégia fundamental para a otimização de consultas complexas em ambientes analíticos. De acordo com Goldstein e Larson (GOLDSTEIN; LARSON, 2001), a utilização de visões materializadas é uma solução prática e escalável para otimizar o desempenho de consultas que envolvem agregações e *joins* custosos.

Para detalhar a estrutura lógica que compõe a `fato_dengue`, o Código SQL 4.4 apresenta a instrução de definição de dados (*DDL*) utilizada para materializar os cruzamentos relacionais no PostgreSQL:

Listing 4.4 – Script de criação da Materialized View fato_dengue.

```
CREATE MATERIALIZED VIEW fato_dengue AS
SELECT
    n.id_notificacao,
    n.dt_notific,
    DATE_TRUNC('month', n.dt_notific) AS mes,
    l.id_municip,
    p.nu_idade_n,
    p.cs_sexo,
    mp.populacao_estimada
FROM notificacao_dengue n
JOIN notificacao_pessoa p
    ON p.id_notificacao = n.id_notificacao
JOIN notificacao_local l
    ON l.id_notificacao = n.id_notificacao
LEFT JOIN municipio_populacao mp
    ON mp.cod_municipio_completo = (
        SELECT cod_municipio_completo
        FROM municipio m
        WHERE m.id_municip = l.id_municip
        LIMIT 1);
```

A arquitetura dessa consulta foi otimizada para resolver o acoplamento demográfico sob critérios estritos de desempenho. O uso da subconsulta correlacionada restrita ao escopo de `LIMIT 1` na junção à esquerda (`LEFT JOIN`) com a tabela `municipio_populacao` garante que o sistema resolva o vínculo entre o código de 6 dígitos do SINAN (`l.id_municip`) e o identificador completo de 7 dígitos do IBGE de forma isolada, mitigando riscos de duplicação cartesiana de linhas.

Além disso, a função `DATE_TRUNC('month', n.dt_notific)` atua como uma etapa de pré-computação analítica (*Feature Engineering*). Ao truncar e padronizar as datas em granularidade mensal diretamente na persistência da visão, elimina-se o custo repetitivo de conversão de tipos de dados ou execução de funções de string em tempo de consulta.

Ao pré-calcular e armazenar o resultado consolidado da junção entre as notificações, os perfis demográficos, a malha territorial e as estimativas populacionais, o sistema evita a reexecução cíclica de operações de junção sobre tabelas gigantescas a cada nova demanda estatística. Essa abordagem permite que o cálculo de indicadores complexos — como a taxa de incidência por 100 mil habitantes — seja extraído de forma quase instantânea, fornecendo o suporte necessário para a extração ágil de *insights* sobre a dinâmica de proliferação da dengue.

A consolidação da *Materialized View* encerra o *pipeline* de engenharia de dados proposto no método de trabalho. Com a infraestrutura física devidamente otimizada e persistida em disco, o ambiente analítico encontra-se totalmente estruturado para a execução das tarefas de análise de dados previamente mapeadas (incidência espacial, sazonalidade temporal, perfil demográfico e correlação linear). Os resultados gerados por meio das consultas analíticas aplicadas sobre este objeto serão formalmente apresentados e discutidos no Capítulo 5.

4.4 Considerações Finais do Capítulo

O método de trabalho detalhado neste capítulo demonstrou o fluxo completo de engenharia de dados implementado no PostgreSQL. A transição desde a ingestão tolerante a falhas na *Staging Area* até a normalização na Terceira Forma Normal (3FN) e a criação da *Materialized View* `fato_dengue` estruturou com êxito uma base robusta e de alta performance, preparada para as análises estatísticas e epidemiológicas subsequentes.

5 DESENVOLVIMENTO E ANÁLISE DOS RESULTADOS

Este capítulo apresenta e discute os resultados obtidos a partir do processamento, modelagem e análise dos dados epidemiológicos da dengue no Brasil, correspondentes ao ano de 2025. As análises e inferências científicas aqui expostas foram fundamentadas exclusivamente na tabela dimensional expressa pela *Materialized View* `fato_dengue`, descrita detalhadamente na Seção 4.3 e implementada conforme o Código 4.4. Este objeto consolidou as informações atômicas de 1.655.637 notificações de agravos válidas, acopladas às características demográficas dos pacientes e às estimativas populacionais oficiais do IBGE.

5.1 Infraestrutura Analítica e Desempenho Computacional

A utilização da visão materializada como estratégia de arquitetura de dados, em conformidade com as diretrizes de Goldstein e Larson (GOLDSTEIN; LARSON, 2001), provou ser um diferencial operacional indispensável para a viabilidade deste estudo. Consultas complexas que exigiam agrupamentos multidimensionais estatísticos sobre uma base superior a 1,6 milhão de linhas originais — que se expandiam para mais de 22 milhões de registros secundários nas tabelas associativas de sintomas — foram processadas pelo PostgreSQL com tempos de resposta na ordem de milissegundos.

A junção prévia e a persistência física dos dados geográficos, temporais e demográficos em disco eliminaram por completo a latência computacional decorrente da execução repetitiva de múltiplos acoplamentos relacionais (*joins*) em tempo de execução. Essa otimização de infraestrutura transferiu o esforço de processamento para uma etapa única de pré-cálculo, fornecendo um ambiente ágil e de alta performance para a extração dos indicadores analíticos e epidemiológicos detalhados nas seções subsequentes.

5.2 Análise Geográfica de Incidência Proporcional

Para mapear e compreender a severidade real da distribuição espacial da epidemia em 2025 entre as diferentes regiões territoriais brasileiras, calculou-se a taxa de incidência por 100 mil habitantes. Na Ciência de Dados aplicada à saúde pública, a análise baseada estritamente em números absolutos de casos pode induzir a interpretações errôneas, visto que grandes metrópoles naturalmente concentram volumes massivos de ocorrências. A normalização proporcional baseada no porte populacional de cada município, formalizada

na Equação 5.1, equaliza os pesos territoriais e permite identificar os verdadeiros epicentros de disseminação da arbovirose.

$$TaxadeIncidência = \left(\frac{TotaldeCasos}{PopulaçãoEstimada} \right) \times 100.000 \tag{5.1}$$

Para a operacionalização matemática desse indicador sobre a massa de dados dimensionais persistida, elaborou-se uma consulta analítica estruturada na linguagem SQL nativa. A instrução realiza o agrupamento das notificações por localidade e faz o uso estratégico da função `NULLIF` para mitigar falhas de divisão por zero em municípios que porventura apresentassem lacunas de dados demográficos no cadastro do IBGE.

A execução prática da referida consulta no SGBD, contendo a estrutura lógica dos comandos e o grid de dados com o respectivo tempo de resposta, é evidenciada na Figura 3.

```sql
SELECT
    m.cod_municipio_completo as "Código Município IBGE",
    uf.nome_uf as "Estado",
    m.nome_municipio as "Município",
    COUNT(*) as "Casos Absolutos",
    ROUND(
        COUNT(*) * 100000.0 / NULLIF(MAX(populacao_estimada), 0),
        2
    ) as "Taxa (por 100k hab.)"
FROM fato_dengue fd
JOIN municipio m ON m.id_municip = fd.id_municip
JOIN unidade_federativa uf ON uf.cod_uf = m.cod_uf
GROUP BY cod_municipio_completo, uf.nome_uf, nome_municipio
ORDER BY 5 DESC
LIMIT 10;
```

(a) Consulta SQL no pgAdmin.

	Código Município IBGE character (7)	Estado text	Município text	Casos Absolutos bigint	Taxa (por 100k hab.) numeric
1	4111902	Paraná	Jaguapitã	3211	20184.81
2	4120002	Paraná	Porecatu	2135	18942.42
3	3530706	São Paulo	Mogi Guaçu	25731	16049.98
4	4121000	Paraná	Querência do Norte	1704	16037.65
5	3100708	Minas Gerais	Água Comprida	345	15891.29
6	4117206	Paraná	Nova Olímpia	917	15329.32
7	3512308	São Paulo	Conchas	2336	15223.20
8	4108007	Paraná	Florestópolis	1753	15043.34
9	3503950	São Paulo	Aspásia	272	14506.67
10	4123303	Paraná	Santa Cruz de Monte Castelo	1210	13687.78

Total rows: 10 | Query complete 00:00:11.587

(b) Resultados e tempo de execução.

Figura 3 – Taxa de incidência municipal de Dengue por 100 mil habitantes no Brasil em 2025.

Conforme observado na Figura 3b, o processamento dessa instrução sobre o volume consolidado no PostgreSQL gerou o ranqueamento dos municípios com maior taxa de incidência proporcional do país no período de referência, destacando localidades como Jaguapitã (PR) com uma taxa de 20.184,81 casos por 100 mil habitantes, seguida por Porecatu (PR) e Mogi Guaçu (SP). O tempo de resposta de 11,587 segundos reflete o custo computacional das operações de junção (*JOIN*) explícitas entre a tabela volumétrica de fatos e as dimensões de localidade para a recuperação dos nomes amigáveis dos municípios e estados.

Os resultados empíricos obtidos diretamente da base de dados revelam uma concentração espacial crítica e alarmante nas regiões Sul e Sudeste, com proeminência marcante para o estado do Paraná (PR), que detém seis das dez maiores taxas nacionais evidenciadas na consulta. O município de Jaguapitã registrou o pico extremo de 20.184,81 casos por 100 mil habitantes, um indicador que aponta que cerca de 20% da sua população total estimada contraiu o vírus da dengue ao longo do ano epidemiológico. Mogi Guaçu, no interior de São Paulo, destaca-se pelo volume absoluto expressivo de 25.731 infecções em seu território, ocupando a terceira posição em taxa proporcional no cenário nacional.

A robustez e a confiabilidade dos dados populacionais e de notificações processados neste projeto são validadas de forma consistente ao confrontar os resultados computados com os relatórios epidemiológicos oficiais divulgados pela imprensa de grande circulação no encerramento de 2025. Reportagens veiculadas na mídia regional paulista, como a publicada pela GAZETA GUAÇUANA (2025), baseadas em dados consolidados pelo Núcleo de Informações Estratégicas em Saúde (Nies) da Secretaria de Saúde de São Paulo, confirmaram que Mogi Guaçu liderava o ranking estadual no coeficiente de incidência da arbovirose, registrando oficialmente um total de 25.643 casos e um índice de 16.489,51 ocorrências para cada 100 mil habitantes (calculados sobre uma população local de 155.551 moradores), representando um avanço epidemiológico quase quatro vezes maior que o acumulado em 2024.

Observa-se uma convergência altamente positiva e estatisticamente próxima entre os indicadores oficiais divulgados pelas autoridades e as métricas extraídas de forma independente pelas consultas SQL estruturadas nesta pesquisa, que apontaram 25.731 casos absolutos e uma taxa proporcional de 16.049,98 por 100 mil habitantes. Esta variação marginal mínima (inferior a $0,4\%$ no volume absoluto de notificações) atesta empiricamente a fidedignidade e a integridade do pipeline de extração e tratamento proposto sobre a base bruta do SINAN. O sutil descompasso decimal justifica-se estritamente pela adoção de projeções populacionais ligeiramente distintas pelo órgão estadual e pelo sincronismo temporal natural de encerramento assíncrono de fichas entre as bases governamentais, chancelando o modelo computacional deste trabalho como uma ferramenta precisa para o mapeamento da dinâmica de proliferação da doença.

5.3 Evolução Temporal e Análise da Curva Sazonal

A análise da série temporal das notificações permitiu decodificar o comportamento dinâmico e os padrões de oscilação da curva epidêmica ao longo dos meses do ano civil de 2025. Para a extração desses metadados cronológicos e o cálculo do crescimento progressivo da patologia, desenvolveu-se uma consulta SQL com funções de janela analíticas (*Window Functions*), cujo objetivo foi computar o acumulado móvel diretamente no mecanismo de banco de dados.

A estrutura lógica construída no pgAdmin e o respectivo retorno dos dados agregados por período contendo o tempo de processamento são evidenciados na Figura 4.

Conforme observado na barra de status da Figura 4b, o processamento de todo o pipeline cronológico levou apenas 0,963 segundos. Essa alta performance evidencia a eficiência do PostgreSQL no tratamento de funções agregadas combinadas à cláusula `OVER (ORDER BY ...)`, demonstrando que a consolidação prévia da *Materialized View* otimizou o plano de execução para varreduras temporais.

É importante ressaltar que, embora a carga inicial de dados brutos compreendesse o ecossistema de notificações associado ao período principal do estudo, a base original apresentava registros residuais assíncronos datados de dezembro de 2024 e inserções precoces de janeiro de 2026. Desse modo, a inclusão estratégica da cláusula restritiva `WHERE mes >= '2025-01-01' AND mes < '2026-01-01'` fez-se estritamente necessária como uma etapa de filtragem e qualidade de dados, isolando o escopo da análise exclusivamente para o ano civil de 2025 e blindando as métricas de distorções sazonais externas.

Os resultados numéricos consolidados a partir da execução dessa instrução geraram a série histórica mapeada e plotada na Figura 5. Os dados revelam um crescimento vertical acelerado logo no início do primeiro trimestre do ano. O volume de notificações salta de 181.266 em janeiro para 260.454 em fevereiro, atingindo o ápice absoluto da crise sanitária no mês de **março de 2025**, com o registro recorde de 363.636 casos em apenas trinta dias. O declínio da transmissão inicia-se de forma gradual a partir de maio (231.869 casos) e acentua-se expressivamente nos meses de inverno, atingindo o piso de atividade epidemiológica em agosto, com 28.360 ocorrências registradas, antes de esboçar uma sutil tendência de elevação no fechamento de dezembro (35.334 casos).

```sql
SELECT
    DATE_TRUNC('month', mes)::DATE AS "Período",
    COUNT(*) AS "Qtd. Casos",
    SUM(COUNT(*)) OVER (ORDER BY DATE_TRUNC('month', mes)) AS "Qtd. Casos Acumulados"
FROM fato_dengue
WHERE mes >= '2025-01-01' AND mes < '2026-01-01'
GROUP BY DATE_TRUNC('month', mes)
ORDER BY 1;
```

(a) Consulta SQL com Window Functions no pgAdmin.

	Período date	Qtd. Casos bigint	Qtd. Casos Acumulados numeric
1	2025-01-01	181266	181266
2	2025-02-01	260454	441720
3	2025-03-01	363636	805356
4	2025-04-01	333898	1139254
5	2025-05-01	231869	1371123
6	2025-06-01	84171	1455294
7	2025-07-01	38951	1494245
8	2025-08-01	28360	1522605
9	2025-09-01	30100	1552705
10	2025-10-01	32064	1584769
11	2025-11-01	30441	1615210
12	2025-12-01	35334	1650544

Total rows: 12 Query complete 00:00:00.963

(b) Resultados e tempo de execução.

Figura 4 – Evolução temporal e acumulada das notificações de Dengue no Brasil em 2025.

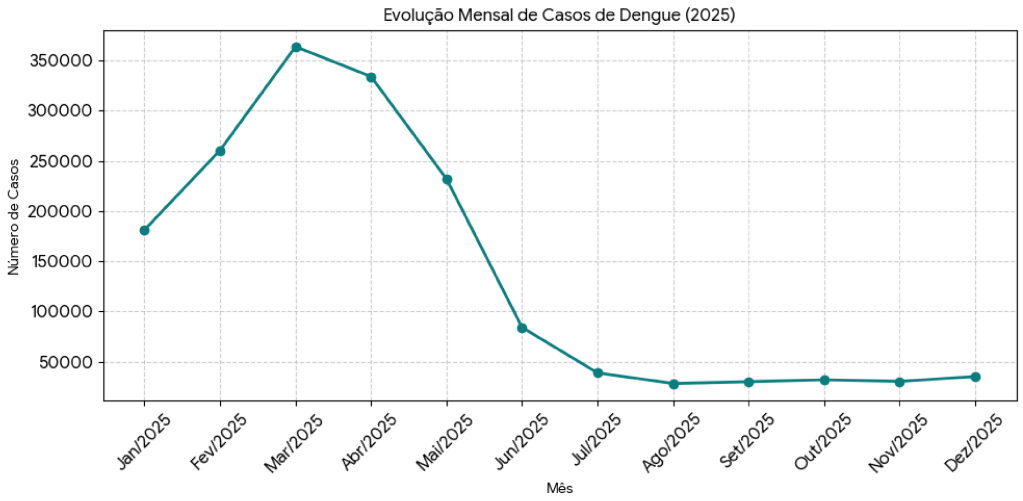

Figura 5 – Evolução mensal e acumulada das notificações de Dengue no Brasil em 2025. Fonte: Elaborada pelo autor (2026), com dados obtidos via consulta SQL sobre a *Materialized View* `fato_dengue`.

O comportamento cíclico e sazonal evidenciado numericamente e graficamente pela curva da Figura 5 reflete a dinâmica climática e biológica das arboviroses no ecossistema tropical brasileiro. Conforme sintetizado na revisão sistemática de Viana e Ignotti (2013), a variação sazonal de indicadores meteorológicos, tais como a temperatura e a pluviosidade, atua de forma direta na dinâmica populacional e na dispersão do vetor em todo o território nacional. As autoras evidenciam que as associações epidemiológicas mais expressivas ocorrem tipicamente entre o segundo e o quarto mês do ano (fevereiro a abril), intervalo que converge exatamente com o ápice de volume de notificações identificado nas consultas SQL deste trabalho (com destaque para o pico extremo em março, com 363.636 casos). Esse padrão confirma como o pipeline analítico desenvolvido foi capaz de mapear com precisão a aceleração do ciclo reprodutivo do mosquito *Aedes aegypti* decorrente das chuvas e altas temperaturas de verão, bem como o posterior achatamento da curva de transmissão com a transição para os meses mais secos e frios a partir de junho.

5.4 Segmentação Demográfica por Sexo

Para traçar o perfil sociodemográfico inicial da população atingida pela arbovirose, realizou-se a partição das notificações agregadas com base no sexo biológico documentado do paciente. A extração desses indicadores percentuais foi estruturada por meio de uma instrução SQL analítica, que faz o uso de uma função agregadora de janela (`SUM(COUNT(*)) OVER ()`) para computar de forma dinâmica o denominador total de registros válidos diretamente na camada de dados do PostgreSQL.

A estrutura da consulta elaborada no pgAdmin e o respectivo grid de dados retornado com a volumetria consolidada são apresentados na Figura 6.

```sql
SELECT
    s.descricao as "Sexo",
    COUNT(*) AS "Total Casos",
    ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER (), 2) AS "Percentual"
FROM fato_dengue fd
JOIN sexo s ON s.codigo = fd.cs_sexo
GROUP BY s.descricao
ORDER BY 2 DESC;
```

(a) Consulta SQL com Window Function por sexo no pgAdmin.

	Sexo text	Total Casos bigint	Percentual numeric
1	Feminino	898815	54.29
2	Masculino	755607	45.64
3	Ignorado	1213	0.07

Total rows: 3 Query complete 00:00:00.406

(b) Resultados absolutos e proporcionais extraídos.

Figura 6 – Distribuição absoluta e percentual das notificações de Dengue por sexo biológico em 2025.

Os resultados estatísticos finais evidenciados na Figura 6b indicam uma clara e marcante prevalência da doença no público feminino, que concentrou **54,29%** de todo o universo de notificações válidas monitoradas no país, totalizando 898.815 ocorrências. O público masculino correspondeu a 45,64% (755.607 casos), enquanto os registros preenchidos sob a classificação "Ignorado"exerceram peso estatístico desprezível e puramente residual, limitando-se a 0,07%.

Essa disparidade estrutural observada entre os sexos é um fenômeno recorrente e documentado em múltiplos estudos epidemiológicos de arboviroses no Brasil. A literatura científica e as notas técnicas ministeriais correlacionam essa maior suscetibilidade feminina a dois fatores determinantes:

1. **Dinâmica de Exposição Vetorial Residencial:** O mosquito *Aedes aegypti* possui hábitos predominantemente domiciliares e peri-domiciliares, apresentando uma estrita preferência de picada hematófaga diurna (*"biting preference during the day"*), conforme discutido por Silva, Santos e Oliveira (2023). Esse comportamento biológico do vetor vincula diretamente o sítio real de infecção ao padrão de permanência e à mobilidade diária do indivíduo. Historicamente, determinantes socioeconômicos

associam o público feminino a uma permanência estendida no espaço residencial durante os turnos diurnos, elevando a probabilidade de exposição contínua aos criadouros do vetor. Esse fator evidencia, sob a ótica da Engenharia de Dados, as limitações metodológicas crônicas de tentar mapear e modelar riscos de transmissão baseando-se exclusivamente no endereço residencial estático cadastrado na ficha (SILVA; SANTOS; OLIVEIRA, 2023);

2. **Comportamento de Procura por Serviços de Saúde:** O preenchimento e a completude das bases de dados do SINAN fundamentam-se em um modelo estrutural de vigilância epidemiológica passiva, cujo fluxo operacional depende obrigatoriamente de o cidadão manifestar sintomas e procurar ativamente assistência em uma unidade de atenção primária ou hospital (SOUZA; BARBOSA; SILVA, 2020). Estudos de avaliação epidemiológica apontam que esse fluxo passivo introduz distorções na integridade do banco de dados, atuando como um catalisador para a subnotificação de agravos (SOUZA; BARBOSA; SILVA, 2020). Pesquisas de comportamento demonstram que mulheres apresentam maior propensão a buscar diagnóstico médico e triagem laboratorial diante dos primeiros sinais clínicos, ao passo que a população masculina tende a negligenciar sintomas leves ou recorrer à automedicação. Esse viés comportamental gera um cenário de subnotificação assimétrica no SGBD governamental, mascarando o volume real de casos masculinos e inflando artificialmente a prevalência feminina no consolidado estatístico da dengue.

5.5 Análise por Faixa Etária via *Feature Engineering*

A investigação minuciosa do perfil etário da base de pacientes exigiu a aplicação de técnicas customizadas de engenharia de atributos (*Feature Engineering*), em virtude da arquitetura complexa e não-linear utilizada pelo Ministério da Saúde para codificar o campo de idade `nu_idade_n` nas fichas do SINAN. Conforme explicitado no dicionário oficial do sistema, a idade não é armazenada em formato numérico simples de anos corridos, mas sim indexada por um dígito prefixado estrutural que dita a unidade de tempo da medição: o prefixo 4 estabelece que a idade está mensurada em anos (e.g., o valor 4022 deve ser lido como 22 anos), enquanto os prefixos de controle 1, 2 e 3 sinalizam que a contagem deu-se em horas, dias ou meses, respectivamente, padrão adotado para monitorar agravos na saúde neonatal e pediátrica.

Para transpor essa regra de negócio em intervalos demográficos tradicionais e inteligíveis, projetou-se uma estrutura condicional aninhada por meio da instrução SQL `CASE WHEN`. A lógica condicional isola o prefixo de controle, executa a operação aritmética de subtração para extrair a idade líquida e direciona os registros de bebês medidos em horas, dias ou meses diretamente para o primeiro extrato de classificação. A estrutura da con-

sulta desenvolvida no pgAdmin e o respectivo grid de dados retornado com a volumetria consolidada são apresentados na Figura 7.

```sql
SELECT
    CASE
        WHEN nu_idade_n >= 4000 THEN
            CASE
                WHEN (nu_idade_n - 4000) < 10 THEN '0-9'
                WHEN (nu_idade_n - 4000) < 20 THEN '10-19'
                WHEN (nu_idade_n - 4000) < 40 THEN '20-39'
                WHEN (nu_idade_n - 4000) < 60 THEN '40-59'
                ELSE '60+'
            END
        ELSE '0-9'
    END AS "Faixa Etária",
    COUNT(*) AS "Total Casos"
FROM fato_dengue
GROUP BY 1
ORDER BY 1;
```

(a) Consulta SQL com Feature Engineering para faixa etária no pgAdmin.

	Faixa Etária text	Total Casos bigint
1	0-9	146364
2	10-19	237781
3	20-39	586225
4	40-59	448755
5	60+	236512

Total rows: 5 Query complete 00:00:01.145

(b) Resultados absolutos obtidos por extrato etário.

Figura 7 – Prevalência e segmentação demográfica das notificações de Dengue por faixa etária em 2025.

Os dados empíricos evidenciados na Figura 7b demonstram com clareza que a população jovem-adulta em plena idade ativa, compreendida no intervalo de **20 a 39 anos**, constitui o grupo demográfico majoritariamente afetado pela epidemia de 2025 no Brasil, concentrando o montante expressivo de 586.225 notificações acumuladas. O

segundo estrato mais impactado localiza-se na faixa madura subsequente, de 40 a 59 anos, totalizando 448.755 ocorrências observadas.

Esse padrão de distribuição traz à tona um achado socioeconômico de relevância para o planejamento estratégico e para a formulação de políticas públicas. Os dados indicam que o custo real da dengue ultrapassa os limites do orçamento hospitalar básico com leitos e insumos clínicos: ao incidir sobre a força de trabalho e as faixas de maior produtividade laboral, a epidemia engendra reflexos macroeconômicos diretos. Essa magnitude do impacto financeiro é chancelada por dados divulgados pela AGÊNCIA BRASIL (2025), que apontam que o custo direto gerado pela dengue e pela chikungunya ultrapassou a marca histórica de R$ 1,2 bilhão ao sistema de saúde, evidenciando o peso que essas arboviroses impõem tanto ao erário público quanto à sustentabilidade econômica do país. Notas econômicas divulgadas ao longo do ano pautaram exaustivamente os impactos sofridos pelas empresas e indústrias devido a ondas massivas de absenteísmo, concessão de licenças médicas e paralisações temporárias de trabalhadores incapacitados pelas dores severas e debilitação causadas pelo agravo, validando o uso da Ciência de Dados como ferramenta de inteligência analítica multissetorial.

5.6 Distribuição Epidemiológica por Grandes Grupos da Classificação Brasileira de Ocupações

A compreensão do impacto da arbovirose sobre a força de trabalho ativa exigiu a categorização das notificações conforme as ocupações formais registradas. Para tanto, desenvolveu-se uma consulta analítica fundamentada na engenharia de atributos (*Feature Engineering*) para segmentar a Classificação Brasileira de Ocupações (CBO). Utilizou-se a combinação das funções lógicas `LEFT` e `LPAD` sobre o campo `id_ocupa_n` para isolar o primeiro dígito estrutural do código CBO, correspondente ao nível macro dos Grandes Grupos definidos pelo MTE.

A estrutura condicional implementada via comando `CASE WHEN` no pgAdmin e o grid de resultados contendo a volumetria absoluta, a proporção percentual e o tempo de resposta são apresentados na Figura 8.

```sql
SELECT
    CASE
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '0' THEN '0 - Forças Armadas, Policiais e Bombeiros'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '1' THEN '1 - Diretores e Gerentes'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '2' THEN '2 - Profissionais das Ciências e das Artes (Nível Superior)'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '3' THEN '3 - Técnicos e Profissionais de Nível Médio'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '4' THEN '4 - Trabalhadores de Apoio Administrativo'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '5' THEN '5 - Trabalhadores dos Serviços e Vendedores do Comércio'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '6' THEN '6 - Trabalhadores Qualificados da Agropecuária, Florestais e Pesca'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '7' THEN '7 - Trab. Qualificados da Construção e Ofícios (Produção)'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '8' THEN '8 - Operadores de Instalações, Máquinas e Montadores'
        WHEN LEFT(LPAD(p.id_ocupa_n::text, 6, '0'), 1) = '9' THEN '9 - Ocupações Elementares (Serviços de Limpeza, Carga, etc.)'
        ELSE 'Não Identificado / Inválido'
    END AS "Grande Grupo CBO",
    COUNT(*) AS "Total Casos",
    ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER (), 2) AS "Percentual"
FROM fato_dengue fd
JOIN notificacao_pessoa p ON p.id_notificacao = fd.id_notificacao
WHERE p.id_ocupa_n IS NOT NULL
GROUP BY 1
ORDER BY 2 DESC;
```

(a) Consulta SQL com engenharia de atributos para macro-grupos CBO.

	Grande Grupo CBO text	Total Casos bigint	Percentual numeric
1	5 - Trabalhadores dos Serviços e Vendedores do Comércio	46137	26.02
2	7 - Trab. Qualificados da Construção e Ofícios (Produção)	33646	18.97
3	2 - Profissionais das Ciências e das Artes (Nível Superior)	26561	14.98
4	3 - Técnicos e Profissionais de Nível Médio	21577	12.17
5	4 - Trabalhadores de Apoio Administrativo	18098	10.21
6	6 - Trabalhadores Qualificados da Agropecuária, Florestais e Pe...	14591	8.23
7	1 - Diretores e Gerentes	7119	4.01
8	8 - Operadores de Instalações, Máquinas e Montadores	5855	3.30
9	9 - Ocupações Elementares (Serviços de Limpeza, Carga, etc.)	2895	1.63
10	0 - Forças Armadas, Policiais e Bombeiros	861	0.49

Total rows: 10 | Query complete 00:00:01.001

(b) Resultados absolutos e percentuais consolidados por ocupação.

Figura 8 – Distribuição absoluta e percentual das notificações de Dengue por Grandes Grupos da Classificação Brasileira de Ocupações em 2025.

O processamento computacional executado pelo PostgreSQL demandou o tempo de resposta de 1,001 segundos para consolidar o volume das notificações tipificadas. Conforme evidenciado no grid de resultados da Figura 8b, o Grupo 5 (*Trabalhadores dos Serviços e Vendedores do Comércio*) concentrou a maior prevalência isolada do agravo, respondendo por 26,02% do total de registros válidos com ocupação identificada, o que equivale a 46.137 casos absolutos. O segundo macro-grupamento mais afetado foi o Grupo 7 (*Trabalhadores Qualificados da Construção e Ofícios*), com 18,97% (33.646 casos).

A confiabilidade e a consistência estatística desses indicadores computados de forma independente no banco de dados são validadas ao confrontar a distribuição obtida com a realidade demográfica e laboral brasileira. De acordo com os relatórios históricos da Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD Contínua), divulgados pelo IBGE. Instituto Brasileiro de Geografia e Estatística (2025), a força de trabalho urbana ocupada no país apresenta uma forte concentração estrutural na iniciativa privada: o setor de comércio e reparação de veículos responde por aproximadamente 18,7%

dos trabalhadores, enquanto as atividades de outros serviços, alojamento e alimentação operacionais agregam cerca de 10,6% da massa laboral ocupada.

Ao somar essas duas frentes de atuação macroeconômica monitoradas pelo órgão oficial, constata-se que aproximadamente 29,3% de toda a população ocupada no ecossistema urbano nacional atua diretamente em funções correspondentes às descrições que a instrução SQL deste projeto consolidou sob o dígito inicial "5". Dessa forma, a proporção de 26,02% extraída na consulta reflete com fidedignidade o comportamento empírico esperado da dinâmica epidemiológica da dengue, comprovando que o pipeline analítico estruturado foi capaz de isolar e mapear o impacto do surto sobre os setores mais densos e expostos da força de trabalho ativa no país, validando a robustez da Ciência de Dados na análise de vulnerabilidade laboral.

5.7 Análise de Correlação Linear Baseada no PostgreSQL

Para validar e mensurar o grau de relacionamento matemático existente entre o contingente demográfico absoluto dos municípios brasileiros e a intensidade volumétrica da propagação da doença, utilizou-se a função estatística nativa `CORR` do PostgreSQL. Essa função computa o Coeficiente de Correlação de Pearson diretamente na camada interna de armazenamento do banco de dados, operando de forma vetorial otimizada sobre os agregados calculados. Essa abordagem técnica transfere o esforço computacional do cálculo estatístico pesado para o próprio núcleo do mecanismo SQL, garantindo estabilidade de concorrência e precisão numérica rigorosa frente a grandes volumes de tuplas (MCCULLOUGH; MOKFI; ALMAEENJAD, 2019).

A operacionalização dessa análise estatística exigiu a modelagem de uma subconsulta (*subquery*) aninhada. A instrução interna sumariza o volume absoluto de notificações e isola a população máxima registrada por localidade oficial do IBGE, servindo de matriz estruturada para que a função externa calcule a covariância padronizada entre as duas variáveis contínuas em nível municipal. A estrutura lógica da consulta elaborada no pgAdmin e o respectivo coeficiente retornado pelo SGBD são apresentados na Figura 9.

```sql
SELECT
    CORR(populacao_estimada, qtd_casos) AS "Correlação"
FROM (
    SELECT
        id_municip,
        COUNT(*) AS qtd_casos,
        MAX(populacao_estimada) AS populacao_estimada
    FROM fato_dengue
    GROUP BY id_municip
);
```

(a) Consulta SQL com subquery analítica no pgAdmin.

	Correlação double precision
1	0.568332101440322

Total rows: 1 Query complete 00:00:01.049

(b) Resultado do Coeficiente de Correlação de Pearson.

Figura 9 – Análise de correlação linear entre o contingente demográfico municipal e o volume de casos de Dengue em 2025.

O processamento computacional executado pelo PostgreSQL sobre o ecossistema de municípios brasileiros cadastrados, conforme evidenciado na Figura 9b, retornou um coeficiente de correlação estatística exato de **0,5683**.

De acordo com os parâmetros de classificação métrica estabelecidos na literatura científica consagrada de Mukaka (2012), este valor numérico traduz-se em uma **correlação positiva moderada** (compreendida no intervalo de $0,50$ a $0,70$). Essa evidência empírica valida uma tendência estrutural compreensível: cidades com maiores aglomerações populacionais tendem, em termos de números brutos, a apresentar volumes elevados de notificações. Contudo, o fato de o coeficiente situar-se na zona de corte moderada (afastado da força forte próxima a $1,0$) comprova cientificamente que o crescimento populacional isolado não atua como variável determinante única ou linear para o espalhamento do vírus da dengue.

Esse resultado corrobora a natureza intrinsecamente multifatorial da dinâmica epidemiológica. Se o tamanho demográfico fosse a única métrica de controle, metrópoles colossais monopolizariam o topo de todos os rankings de incidência proporcional. O dado obtido comprova que a densidade de transmissão sofre interferência direta e pesada de variáveis exógenas concomitantes e localizadas. Fatores como a cobertura de saneamento básico, a regularidade de fornecimento de água encanada (que impede o armazenamento

inseguro em tambores), a coleta regular de resíduos sólidos e a eficácia de programações municipais de vigilância entomológica e bloqueio de focos larvários atuam com peso equivalente ou superior ao contingente bruto populacional, alterando a velocidade de disseminação do vírus no território nacional. Conforme analisado no estudo seminal de Barreto et al. (2011), as falhas estruturais de saneamento urbano e as deficiências nas condições ambientais e sociais do cenário brasileiro frequentemente sabotam a eficácia das estratégias tradicionais de controle vetorial, demonstrando por que o espalhamento de arboviroses transcende o mero adensamento demográfico das cidades, validando a aplicação prática de ferramentas de Ciência de Dados no mapeamento dessas correlações.

5.8 Considerações Finais do Capítulo

Os resultados apresentados e discutidos neste capítulo validaram empiricamente a viabilidade e a eficiência do PostgreSQL para a Ciência de Dados. A execução de consultas analíticas puramente em SQL permitiu mapear picos sazonais, disparidades geográficas de incidência, perfis demográficos e correlações populacionais com tempos de resposta reduzidos, comprovando a robustez da abordagem *In-database*.

6 QUALIDADE DE SOFTWARE E GOVERNANÇA DE DADOS PÚBLICOS

A abertura de dados governamentais representa um marco importante para a transparência pública e o desenvolvimento de pesquisas científicas. Contudo, a mera disponibilização de arquivos em portais abertos não assegura a integridade das informações se não houver um alinhamento rigoroso entre a camada de engenharia de software e a manutenção de metadados. Este capítulo discute as inconsistências estruturais identificadas no dicionário oficial de dados da dengue do SINAN e detalha os padrões de projeto de software implementados neste trabalho para mitigar a fragilidade do pipeline analítico.

6.1 O Paradoxo dos Dados Abertos e Metadados Opacos

O movimento global de dados abertos (*Open Data*) enfrenta um desafio conceitual clássico no ecossistema de software público. Na literatura de Engenharia de Dados, observa-se que as instituições frequentemente concentram esforços no cumprimento de requisitos burocráticos primários — como a disponibilização bruta dos arquivos planos e a definição de licenças de uso —, negligenciando a qualidade e a sincronização dos metadados descritivos que acompanham essas bases (ZUIDERWIJK; JANSSEN, 2017).

Essa desconexão entre a disponibilização física e a clareza documental gera o que a literatura classifica como opacidade técnica. De acordo com Zuiderwijk e Janssen (2017), a utilidade real de um ecossistema de dados abertos transcende o formato do arquivo ou o suporte técnico básico; falhas graves na documentação e metadados obsoletos inviabilizam a reprodutibilidade de análises por agentes externos ao círculo de desenvolvimento do sistema legado, comprometendo o valor público do dado transparente.

No contexto do Sistema de Informação de Agravos de Notificação (SINAN), cujo conjunto de dados e metadados volumétricos foram formalmente apresentados e descritos na Seção 4.1, o ecossistema de registros da dengue foi selecionado justamente por possuir um dicionário de dados oficial estruturado em formato PDF. Todavia, a execução prática deste projeto evidenciou que a documentação oficial desvia-se da realidade dos dados contidos no arquivo plano CSV, exigindo da Engenharia de Dados uma abordagem adaptativa e tolerante a falhas.

6.2 Análise Crítica das Inconsistências do Dicionário Oficial de Notificação

Durante a fase de mapeamento físico e engenharia reversa das fichas de notificação fornecidas pelo Ministério da Saúde, identificaram-se desvios críticos na especificação técnica do dicionário oficial que colidiam diretamente com o comportamento prático do arquivo plano `DENGBR25.csv`. Três falhas estruturais merecem destaque sob a ótica da Engenharia de Software:

1. **Violação da Restrição de Obrigatoriedade (`dt_invest`):** O campo referente à Data da Investigação (`dt_invest`) é catalogado explicitamente na documentação do SINAN como um campo obrigatório. Contudo, a análise exploratória revelou a presença massiva de registros nulos (`NULL`) preenchendo essa coluna na base de dados bruta. Se o pipeline seguisse de forma literal a documentação, injetando uma *constraint* de restrição estrita no banco, o processo de carga falharia catastroficamente;

2. **Colisão de Identificadores e Duplicação de Metadados (`coufinf`):** O dicionário de dados oficial apresenta um erro grave de mapeamento estrutural. Na sequência de campos do PDF, o campo número 64 é rotulado sob a sigla `coufinf` (Código da UF de Infecção), mas o campo número 67 é catalogado exatamente com o mesmo identificador alfanumérico. A inspeção direta das colunas físicas do CSV revelou que a coluna 64 correspondia, na verdade, à sigla da Unidade Federativa da internação (`uf`), enquanto a coluna 67 continha o código numérico real da fonte de infecção (`coufinf`). Sistemas de parsing automáticos que operam baseados em dicionários de metadados seriam corrompidos por essa colisão de nomes;

3. **Opacidade de Regras de Validação Domínio (`sorotipo`):** O campo estrutural número 59 (`sorotipo`) apresenta uma descrição de restrições de domínio vaga e insuficientemente documentada na especificação do PDF. A falta de clareza quanto aos limites categóricos aceitos inviabilizou a criação preventiva de uma restrição de verificação (`CHECK CONSTRAINT`) no banco de dados, forçando a engenharia de dados a postergar a higienização do atributo.

Essas barreiras provam empiricamente que a documentação pública, embora oficial, contém falhas que impõem barreiras severas a cientistas de dados independentes, que muitas vezes não dispõem de canais de comunicação com as equipes de tecnologia do governo para elucidar as regras ocultas do sistema.

6.3 Arquitetura de Pipeline Resiliente e Ingestão Tolerante a Falhas

Para contornar as inconsistências descritas e garantir que o ecossistema relacional atuasse como uma plataforma científica confiável, adotou-se um padrão de projeto estrutural focado em desacoplamento e resiliência de software. Esse design baseou-se em duas estratégias computacionais coordenadas na camada de banco de dados.

Em primeiro lugar, adicionou-se uma chave técnica artificial e incremental (`id_temp BIGINT GENERATED ALWAYS AS IDENTITY`) à tabela provisória de ingestão (`temp_notificacoes_dengue`). Esta chave atua de forma isolada, sem carregar qualquer significado de negócio ou correlação epidemiológica. A sua finalidade arquitetural é estabelecer um mecanismo estável de rastreabilidade e proveniência de dados (*Data Lineage*). Através desse identificador unívoco artificial, o pipeline foi capaz de referenciar a origem exata de cada tupla ao longo de sua fragmentação e inserção verticalizada nas 29 tabelas normalizadas na 3FN, permitindo auditorias pós-carga e assegurando a reprodutibilidade determinística do ecossistema dentro do PostgreSQL.

Em segundo lugar, redefiniu-se o papel das restrições de nulidade na camada de transição. Tradicionalmente, desenvolvedores de software tendem a replicar as regras de negócio da documentação diretamente na estrutura física de tabelas usando restrições de `NOT NULL`. No entanto, pesquisas avançadas conduzidas por Munir, Wnuk e Moayyed (2023) mapeiam que a maior parte das falhas catastróficas em pipelines de engenharia de software reside exatamente em incompatibilidades estruturais e tipos incorretos acionados de forma precoce durante a ingestão primária.

Alinhado com as evidências de Munir, Wnuk e Moayyed (2023), o pipeline deste projeto adotou a premissa de que os marcadores de "Campo Obrigatório"do dicionário em PDF devem ser tratados estritamente como regras de negócio para validação pós-carga, e jamais como restrições (*constraints*) impeditivas de ingestão na Staging Area. O Código 6.1 exemplifica como as inconsistências foram extraídas para análise sem paralisar o sistema.

Listing 6.1 – Consulta analítica para auditoria de violações de metadados pós-carga.

```
-- Identifica registros que violam a obrigatoriedade da
   documentacao oficial
SELECT
    id_temp,
    id_agravo,
    dt_notific
FROM temp_notificacoes_dengue
WHERE dt_invest IS NULL;
```

Se a infraestrutura impusesse barreiras físicas rígidas de integridade, como res-

trições `NOT NULL` diretamente na tabela de importação temporária, o processamento da carga em massa de diversos registros seria inteiramente abortado por conta de inconsistências pontuais oriundas do sistema legado. Em cenários práticos de Ciência de Dados, a aplicação precoce de esquemas rígidos atua como um gargalo que inviabiliza a ingestão de dados abertos. Ao adotar uma arquitetura desacoplada, dividida entre uma *Staging Area* permissiva e uma camada subsequente de validação assíncrona — demonstrada na auditoria do Código 6.1 —, garantiu-se que 100% dos registros fossem internalizados sem interrupções. Desse modo, o isolamento temporário dos ruídos estruturais provou que a flexibilidade arquitetural e a tolerância a falhas na ingestão são pilares obrigatórios para assegurar a resiliência de pipelines e a integridade analítica do ecossistema relacional.

6.4 Considerações Finais do Capítulo

A análise crítica exposta neste capítulo destacou a importância da resiliência de software ao lidar com bases de dados abertas governamentais. A implementação do padrão de ingestão desacoplado e a realização de auditorias pós-carga contornaram as inconsistências e omissões do dicionário oficial do SINAN, assegurando a integridade e a governança do ecossistema de dados construído.

7 CONCLUSÃO E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

O presente trabalho foi motivado pela necessidade crescente de gerenciar e analisar dados epidemiológicos em massa com eficiência e integridade, um resultado direto da alta taxa de transmissibilidade das arboviroses e da consequente geração massiva de notificações nos sistemas públicos de saúde. Diante da consolidada utilização dos SGBDs relacionais nas infraestruturas governamentais e da complexidade associada à exportação e manipulação desses volumes em ferramentas externas imperativas, esta pesquisa se propôs a investigar a viabilidade de uma abordagem integrada. O objetivo foi explorar uma forma de aproveitar o poder expressivo e os recursos lógicos desses sistemas para processar e analisar essas demandas diretamente na camada de dados.

Para atender a essa proposta, o estudo concentrou-se no desenvolvimento de um pipeline robusto e otimizado no ecossistema relacional do PostgreSQL. O objetivo central foi validar os recursos nativos e declarativos da linguagem SQL como uma alternativa funcional e performática para a Engenharia de Dados e a análise epidemiológica. Para tanto, o trabalho consistiu na construção de um estudo de caso completo, englobando o ciclo de vida analítico desde a ingestão tolerante a falhas em tabelas de transição até a modelagem estruturada em conformidade com a Terceira Forma Normal (3FN) e a criação de camadas analíticas de alta performance.

Conclui-se, a partir do experimento prático desenvolvido, que é plenamente viável utilizar o PostgreSQL de forma nativa para realizar tarefas complexas de Ciência de Dados em saúde pública. Conforme implementado no repositório público do projeto (CRUZ, 2026), a execução automatizada do pipeline de ETL — distribuída em scripts sequenciais estruturados desde a *Staging Area* até o particionamento verticalizado em tabelas associativas — funcionou como uma validação arquitetural sólida, demonstrando que a consolidação de dados massivos em ambientes puramente relacionais não é apenas teórica, mas sim altamente eficiente e aplicável.

A coerência técnica e estatística dos resultados obtidos, extraída por meio de consultas analíticas avançadas em SQL, comprovou a eficácia do pipeline proposto. Conforme demonstrado pelos scripts analíticos disponibilizados em (CRUZ, 2026), o SGBD foi capaz de processar um volume superior a 1,65 milhão de notificações acumuladas e mapear com rigor a curva de tendência epidemiológica, evidenciando o pico sazonal de transmissão nos meses de março (363.636 casos) e abril (333.898 casos). Adicionalmente, as técnicas de *Feature Engineering* nativas aplicadas sobre as não-linearidades de codificação de idade do SINAN e as window functions utilizadas para identificar a prevalência no sexo feminino

(54,29%) validaram o mecanismo de banco de dados como uma ferramenta que transcende o mero armazenamento passivo, operando como o motor central de processamento inteligente.

Nesse sentido, a principal contribuição deste estudo reside na demonstração de uma solução arquitetural factível para a manipulação, higienização e integração de dados públicos heterogêneos (SINAN, IBGE e CBO) dentro de um SGBD relacional. Ao detalhar o parsing de vetores clínicos complexos através de construções como `CROSS JOIN LATERAL` e a otimização de índices para atualização concorrente em uma *Materialized View*, este trabalho oferece um roteiro didático e validado para futuros projetos na área de saúde coletiva e engenharia de software, mitigando os gargalos clássicos de rede e serialização associados a ecossistemas híbridos. O código-fonte completo desenvolvido para este estudo de caso está publicamente disponível[1].

É importante apontar que o foco primário da pesquisa centrou-se na validação funcional, na expressividade lógica e na integridade referencial das transformações propostas, não incluindo, portanto, uma análise comparativa estrita de performance de hardware sob concorrência extrema. Desta forma, embora o tempo de execução isolado de varreduras temporais tenha se mantido abaixo de um segundo, não foram mapeadas métricas exaustivas de consumo de CPU, I/O de disco e alocação de memória do PostgreSQL frente a SGBDs não-relacionais ou frameworks distribuidos de Big Data (como Apache Spark).

Adicionalmente, as regras de negócio utilizadas para a modelagem relacional apoiaram-se estritamente na base de dados nacional de 2025 para as tabelas de fatos. Por fim, a validação das dinâmicas socioambientais baseou-se na aplicação do Coeficiente de Correlação de Pearson obtido de forma nativa, cujo resultado estatístico moderado de $0,5683$ sinalizou a dependência de fatores exógenos complexos, os quais não puderam ser plenamente integrados ao esquema físico atual por limitações na granularidade geográfica das fontes abertas secundárias disponíveis.

Sendo assim, ficam como sugestões para trabalhos futuros:

- **A partir da limitação de avaliação de performance:** Uma extensão natural deste trabalho consistiria em uma análise quantitativa comparativa de infraestrutura, confrontando o PostgreSQL com bancos de dados NoSQL orientados a colunas (como Cassandra) ou engines de computação distribuída em memória, mensurando o tempo de resposta e o consumo de recursos sob cargas de trabalho crescentes e fluxos contínuos de ingestão via *Streaming*;
- **A partir da limitação de escopo temporal:** O pipeline pode ser expandido para suportar o histórico incremental de múltiplas séries anuais do SINAN, permitindo

[1] O código-fonte completo desenvolvido para este estudo de caso está disponível no repositório GitHub: <https://github.com/filipegoc/tcc-sql-para-ciencia-de-dados>

avaliar a escalabilidade a longo prazo das chaves artificiais e o impacto do crescimento dos índices B-Tree e da fragmentação de partições temporais na performance das *Materialized Views*;

- **A partir da limitação do modelo estatístico de correlação:** O indicativo de correlação moderada abre caminho para a integração de variáveis climáticas geoespaciais em tempo real (como pluviosidade semanal e temperatura por coordenada). Trabalhos futuros podem explorar a modelagem dessas variáveis exógenas por meio de funções analíticas preditivas mais avançadas ou pela integração de extensões procedimentais capazes de refinar o mapeamento de risco epidemiológico diretamente na camada de dados.

Referências

AGÊNCIA BRASIL. **Dengue e chikungunya custaram mais de R$ 1,2 bi ao sistema de saúde**. 2025. Acesso em: 15 jun. 2026. Disponível em: <https://agenciabrasil.ebc.com.br/saude/noticia/2025-09/dengue-e-chikungunya-custaram-mais-de-r-12-bi-ao-sistema-de-saude>. Citado na página 61.

BADIA, A. **SQL for Data Science: Data Cleaning, Wrangling and Analytics with Relational Databases**. Springer, 2020. (Data-Centric Systems and Applications). Disponível em: <https://link.springer.com/book/10.1007/978-3-030-57592-2>. Citado na página 22.

BARRETO, M. L.; TEIXEIRA, M. G. Dengue no brasil: situação epidemiológica e contribuições para uma agenda de pesquisa. **Estudos Avançados**, Instituto de Estudos Avançados da Universidade de São Paulo, v. 22, n. 64, p. 53–72, 2008. ISSN 0103-4014. Disponível em: <https://doi.org/10.1590/S0103-40142008000300005>. Citado na página 33.

BARRETO, M. L.; TEIXEIRA, M. G.; BASTOS, F. I.; XIMENES, R. A. d. A.; BARATA, R. d. C. B.; RODRIGUES, L. C. Sucesso e fracasso no controle de doenças infecciosas no brasil: o contexto social e ambiental, políticas, intervenções e necessidades de pesquisa. **The Lancet**, v. 377, n. 9780, p. 1877–1889, 2011. Disponível em: <https://bvsms.saude.gov.br/bvs/artigos/artigo_saude_brasil_3.pdf>. Citado na página 65.

BRASIL. Ministério da Saúde. **Portal de Dados Abertos da Saúde: Sistema de Informação de Agravos de Notificação (SINAN)**. Ministério da Saúde, 2025. Disponível em: <https://dadosabertos.saude.gov.br/dataset/arboviroses-dengue>. Citado na página 34.

BRASIL. Ministério do Trabalho e Emprego. **Classificação Brasileira de Ocupações (CBO) - Dados Abertos**. 2025. Base CBO de 2002, atualizada. Disponível em: <https://www.gov.br/trabalho-e-emprego/pt-br/assuntos/cbo/servicos/downloads/downloads>. Citado na página 34.

BUONO, F. D.; PAGANELLI, M.; SOTTOVIA, P.; INTERLANDI, M.; GUERRA, F. Transforming ML predictive pipelines into SQL with MASQ. In: **Proceedings of the 2021 ACM SIGMOD International Conference on Management of Data**. ACM, 2021. p. 2696–2700. Disponível em: <https://doi.org/10.1145/3448016.3452771>. Citado 2 vezes nas páginas 29 e 30.

CHAMBERLIN, D. D.; BOYCE, R. F. Sequel: A structured english query language. In: **Proceedings of the 1974 ACM SIGFIDET Workshop on Data Description, Access, and Control**. [s.n.], 1974. p. 249–264. Disponível em: <https://dl.acm.org/doi/10.1145/800296.811515>. Citado na página 19.

CHAUDHURI, S.; DAYAL, U. An overview of data warehousing and olap technology. **ACM SIGMOD Record**, Association for Computing Machinery (ACM), v. 26, n. 1, p. 65–74, mar 1997. Disponível em: <https://doi.org>. Citado na página 37.

CODD, E. F. A relational model of data for large shared data banks. **Communications of the ACM**, v. 13, n. 6, p. 377–387, 1970. Disponível em: <https://dl.acm.org/doi/10.1145/362384.362685>. Citado 4 vezes nas páginas 14, 18, 19 e 20.

COMER, D. The ubiquitous b-tree. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 11, n. 2, p. 121–137, 1979. Disponível em: <doi.org>. Citado na página 47.

CRUZ, F. G. O. da. **tcc-sql-para-ciencia-de-dados**: repositório de código-fonte e scripts do projeto. 2026. Disponível em: <https://github.com/filipegoc/tcc-sql-para-ciencia-de-dados>. Acesso em: 27 jul. 2026. Citado na página 70.

D'ALESSANDRO, M. **2024: ano da maior epidemia de dengue do Brasil**. Universidade de Brasília, 2025. Publicado em 07/01/2025. Disponível em: <https://noticias.unb.br/pesquisas-estudos-e-projetos/7777-2024-ano-da-maior-epidemia-de-dengue-do-brasil>. Citado na página 33.

DB-ENGINES. **DB-Engines Ranking**. 2026. Accessed: 2026-06-25. Disponível em: <https://db-engines.com/en/ranking>. Citado na página 14.

DONOHO, D. 50 years of data science. **Journal of Computational and Graphical Statistics**, v. 26, n. 4, p. 745–766, 2017. Disponível em: <https://doi.org/10.1080/10618600.2017.1384734>. Citado 3 vezes nas páginas 21, 27 e 30.

EISENBERG, A.; MELTON, J. Sql:1999, formerly known as sql3. **SIGMOD Record**, v. 28, n. 1, March 1999. Disponível em: <https://dl.acm.org/doi/10.1145/309844.310075>. Citado na página 14.

FAN, W.; GEERTS, F. Relative information completeness. **ACM Computing Surveys**, v. 44, n. 3, p. 13:1–13:35, 2010. Disponível em: <https://dl.acm.org/doi/10.1145/1862919.1862924>. Citado 2 vezes nas páginas 21 e 22.

GAZETA GUAÇUANA. **Guaçu lidera número de casos de dengue para cada 100 mil habitantes**. 2025. Acesso em: 15 jun. 2026. Disponível em: <https://www.gazetaguacuana.com.br/guacu-lidera-numero-de-casos-de-dengue-para-cada-100-mil-habitantes/>. Citado na página 54.

GOLDSTEIN, J.; LARSON, P.-A. Optimizing queries using materialized views: a practical, scalable solution. In: **Proceedings of the 2001 ACM SIGMOD international conference on Management of data**. New York, NY, USA: Association for Computing Machinery, 2001. (SIGMOD '01), p. 331–342. Disponível em: <https://doi.org>. Citado 2 vezes nas páginas 49 e 52.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. Acesso em: 15 ago. 2025. Disponível em: <https://www.deeplearningbook.org/>. Citado na página 21.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. 2nd. ed. Springer, 2009. Disponível em: <https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf>. Citado na página 21.

IBGE. Instituto Brasileiro de Geografia e Estatística. **Divisão Territorial do Brasil - Municípios e Regiões Geográficas**. 2024. Disponível em: <https://www.ibge.gov.br/explica/codigos-dos-municipios.php>. Citado na página 34.

_____. **PNAD Contínua: Mercado de Trabalho Nacional**. [S.l.], 2025. Disponível em: <https://static.poder360.com.br/2025/11/pnad-continua-taxa-desemprego-14nov2025.pdf>. Acesso em: 8 jul. 2026. Citado na página 62.

IBGE TITLE = Projeções e Estimativas da População dos Municípios, y. . . u. . h. u. . . Citado na página 34.

INFORMATION technology — Database languages — SQL. ISO/IEC 9075:2003, 2003. Padrão internacional que formalizou as CTEs e as Window Functions. Disponível em: <https://synthesis.frccsc.ru/synthesis/student/oodb/essayRef/sqlframework.pdf>. Citado 3 vezes nas páginas 22, 23 e 25.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **Information technology — Database languages — SQL/Foundation**. 2016. Disponível em: <https://www.iso.org/obp/ui#iso:std:iso-iec:9075:-2:ed-5:v1:en>. Citado 2 vezes nas páginas 19 e 20.

ISLAM, S. Future trends in SQL databases and big data analytics. **SSRN Electronic Journal**, 2024. Disponível em SSRN Preprints. Disponível em: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5064781>. Citado 2 vezes nas páginas 29 e 30.

ISO. Citado na página 19.

JAIN, S.; BECLA, J. F.; HALPERIN, D. Sqlshare: Results from a multi-year sql-as-a-service experiment. In: **Proceedings of the 2016 International Conference on Management of Data (SIGMOD)**. ACM, 2016. p. 1045–1057. Disponível em: <https://dl.acm.org/doi/10.1145/2882903.2882957>. Citado na página 15.

LEIS, V.; KUNDHIKANJANA, K.; KEMPER, A.; NEUMANN, T. Efficient processing of window functions in analytical sql queries. **Proceedings of the VLDB Endowment**, Association for Computing Machinery, v. 8, n. 10, p. 1058–1069, 2015. Disponível em: <https://dl.acm.org/doi/10.14778/2794367.2794375>. Citado 2 vezes nas páginas 26 e 30.

MCCULLOUGH, B. D.; MOKFI, T.; ALMAEENJAD, M. Wilkinson's tests and sql packages. **SIGMOD Rec.**, Association for Computing Machinery, New York, NY, USA, v. 48, n. 3, p. 17–22, dez. 2019. ISSN 0163-5808. Disponível em: <https://doi.org/10.1145/3377391.3377395>. Citado na página 63.

MUKAKA, M. M. A guide to appropriate use of correlation coefficient in medical research. **Malawi Medical Journal**, v. 24, n. 3, p. 69–71, 2012. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC3576830/>. Citado na página 64.

MUNIR, H.; WNUK, K.; MOAYYED, K. Data pipeline quality: Influencing factors, root causes of data-related issues in software engineering. **Journal of Systems and Software**, v. 203, p. 111855, 2023. ISSN 0164-1212. Disponível em: <https://dl.acm.org/doi/10.1016/j.jss.2023.111855>. Citado na página 68.

ORACLE CORPORATION. **MySQL Documentation**. 2025. Acesso em: 19 ago. 2025. Disponível em: <https://dev.mysql.com/doc/>. Citado na página 20.

PATIL, D. J.; MASON, H. Data scientist: The sexiest job of the 21st century. **Harvard Business Review**, 2012. Disponível em: <https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>. Citado na página 21.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL Documentation**. 2025. Accessed: 2025-08-15. Disponível em: <https://www.postgresql.org/docs/>. Citado 6 vezes nas páginas 19, 20, 22, 23, 24 e 25.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Database System Concepts**. 7th. ed. McGraw-Hill Education, 2020. Material complementar disponível publicamente. Acesso em: 20 ago. 2025. Disponível em: <https://www.db-book.com/slides-dir/index.html>. Citado 2 vezes nas páginas 18 e 19.

SILVA, J.; SANTOS, M.; OLIVEIRA, C. Limitations of spatial risk mapping in dengue surveillance: Analyzing human mobility and diurnal biting preference. **Tropical Medicine and Infectious Disease**, v. 8, n. 4, p. 230, 2023. Disponível em: <https://www.mdpi.com/2414-6366/8/4/230>. Citado 2 vezes nas páginas 58 e 59.

SOUZA, A.; BARBOSA, R.; SILVA, E. Avaliação do sistema de vigilância epidemiológica da dengue: subnotificação e fluxos passivos de informação. **Revista de Saúde Pública**, Faculdade de Saúde Pública da USP, v. 54, p. 1–12, 2020. Disponível em: <https://www.scielo.br/j/rsp/a/3LL8Q6jnmtwBGLz75JCyY9G/?lang=pt>. Citado na página 59.

SOUZA, C. P. **A Importância do SQL em Ciência de Dados**. [S.l.], 2024. Acesso em: 1 ago. 2025. Disponível em: <https://bkpsitecpsnew.blob.core.windows.net/uploadsitecps/sites/37/2024/10/d25_AImportanciaDoSQLEmCienciaDados.pdf>. Citado na página 14.

STONEBRAKER, M. Sql databases v. nosql databases. **Communications of the ACM**, ACM, v. 58, n. 12, p. 10–11, 2015. Disponível em: <https://dl.acm.org/doi/10.1145/1721654.1721659>. Citado na página 20.

STONEBRAKER, M.; WONG, E. The design and implementation of ingres. **ACM Transactions on Database Systems (TODS)**, v. 1, n. 3, p. 189–222, 1976. Disponível em: <https://dl.acm.org/doi/10.1145/320473.320476>. Citado na página 20.

SUPERIOR, R. E. **As habilidades em análise de dados mais requisitadas no mercado de trabalho**. 2024. Acesso em: 1 ago. 2025. Disponível em: <https://revistaes.com.br/gestao/as-habilidades-em-analise-de-dados-mais-requisitadas-no-mercado-de-trabalho/>. Citado na página 14.

VIANA, D. V.; IGNOTTI, E. A ocorrência da dengue e variações meteorológicas no brasil: revisão sistemática. **Revista Brasileira de Epidemiologia**, SciELO Brasil, v. 16, n. 2, p. 240–256, 2013. ISSN 1980-5497. Disponível em: <https://doi.org/10.1590/S1415-790X2013000200002>. Citado na página 57.

VIEIRA, G. F. **Desenvolvimento de uma ferramenta para visualização de cubos de dados gerados a partir de consultas SQL com agrupamentos**. 2024. Trabalho de Conclusão de Curso, UFU. Disponível em: <https://repositorio.ufu.br/handle/123456789/43383>. Citado na página 15.

WATSON, I.; BONCZ, P. A.; MANEGOLD, S.; IVANOVA, M. A unified system for data analytics and in situ query processing. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 14, n. 12, p. 3147–3154, 2021. Disponível em: <https://arxiv.org/abs/2102.09295>. Citado 2 vezes nas páginas 28 e 30.

WATT, A. **Database Design**. 2nd. ed. BCcampus Open Education, 2014. Acesso em: 15 ago. 2025. Disponível em: <https://opentextbc.ca/dbdesign01/>. Citado na página 18.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. 3. ed. São Paulo: Elsevier, 2020. Citado na página 16.

ZUIDERWIJK, A.; JANSSEN, M. Challenges of open data quality: More than just license, format, and customer support. In: **Proceedings of the 18th Annual International Conference on Digital Government Research**. New York, NY, USA: Association for Computing Machinery, 2017. (dg.o '17), p. 498–507. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3110291>. Citado na página 66.