

# **EPISAFE**

**BÁRBARA BATISTTELLA DE OLIVEIRA**

**LEANDRO BERTOCCHI**

**LEONARDO AGOSTINI COSTA**

**IRAÊ ERVIN GRUBER DA SILVA**

**JEFFERSON ALAN SCHMIDT LUDWIG**

**UNOESC - SÃO MIGUEL DO OESTE**

**2024**

**BÁRBARA BATISTTELLA DE OLIVEIRA**

**LEANDRO BERTOCCHI**

**LEONARDO AGOSTINI COSTA**

**IRAÊ ERVIN GRUBER DA SILVA**

**JEFFERSON ALAN SCHMIDT LUDWIG**

# **EPISAFE**

**Trabalho de trabalho acadêmico  
apresentado ao Curso de Engenharia de  
Software I, Programação II e Banco de  
Dados II**

**Orientador: Prof. ROBERSON JUNIOR FERNANDES ALVES**

**Orientador: Prof. OTILIA DONATO BARBOSA**

**UNOESC - SÃO MIGUEL DO OESTE**

**2024**

BÁRBARA BATISTTELLA DE OLIVEIRA

LEANDRO BERTOCCHI

LEONARDO AGOSTINI COSTA

IRAÊ ERVIN GRUBER DA SILVA

JEFFERSON ALAN SCHMIDT LUDWIG

## **EPISAFE**

Dissertação apresentada ao Programa de  
Graduação, Universidade UNOESC Área de  
Ciências da Computação

Aprovada em

**BANCA EXAMINADORA**

---

**Prof. ROBERSON JUNIOR FERNANDES ALVES**

**Universidade Unoesc - São Miguel do Oeste**

---

**Prof. OTILIA DONATO BARBOSA**

**Universidade Unoesc - São Miguel do Oeste**

## Agradecimentos

Gostaríamos de agradecer a todos os envolvidos, professores e colegas que nos auxiliaram durante o processo de desenvolvimento e contribuíram para a finalização deste trabalho.

## Epígrafe

"A prevenção é a base da saúde pública, mas a verdadeira força de uma sociedade está em sua capacidade de agir com rapidez, ciência e solidariedade diante das crises."

## RESUMO

EPISAFE é um programa para aprimorar a prevenção e o controle de doenças como a dengue, monitorando áreas de risco e implementando ações direcionadas. Além disso, o programa é uma ferramenta valiosa para a realização de levantamentos censitários e estudos socioeconômicos, auxiliando na identificação de demandas prioritárias em diferentes regiões da cidade.

EPISAFE is a program to improve the prevention and control of diseases such as dengue fever by monitoring risk areas and implementing targeted actions. In addition, the program is a valuable tool for conducting census surveys and socioeconomic studies, helping to identify priority demands in different regions of the city.

## Lista de Ilustrações

1. <b>Figura 1</b> – Consultas SQL .....	7
2. <b>Figura 2</b> – Consultas SQL .....	8
3. <b>Figura 3</b> – Criação Tabela Endereço .....	14
4. <b>Figura 4</b> – Criação Tabela Área de atuação .....	14
5. <b>Figura 5</b> – Criação Tabela Área de atuação com endereços .....	15
6. <b>Figura 6</b> – Criação Tabela Residência .....	15
7. <b>Figura 7</b> – Criação Tabela Usuário .....	15
8. <b>Figura 8</b> – Criação Tabela Agente .....	15
9. <b>Figura 9</b> – Criação Tabela Ciclo .....	15
10. <b>Figura 10</b> – Criação Tabela Visita .....	15
11. <b>Figura 11</b> – Criação Tabela Armadilha .....	15
12. <b>Figura 12</b> – Criação Tabela Caso de dengue .....	16
13. <b>Figura 13</b> – Criação Tabela Cadastro de notificação .....	16
14. <b>Figura 14</b> – Diagrama .....	16

## Lista de Abreviaturas e Siglas

- **ABNT**: Associação Brasileira de Normas Técnicas
- **DBMS**: Database Management System (Sistema de Gerenciamento de Banco de Dados)
- **DBeaver**: Ferramenta de gerenciamento de banco de dados
- **EPISAFE**: Sistema de monitoramento e controle de epidemias
- **SGDB**: Sistema Gerenciador de Banco de Dados
- **SQL**: Structured Query Language (Linguagem de Consulta Estruturada)
- **UNOESC**: Universidade do Oeste de Santa Catarina

## Lista de Símbolos

- ●: Indicador de itens em listas (marcadores de requisitos e funcionalidades)
- →: Representação de direcionamento ou relacionamento em diagramas e modelos relacionais
- $\Sigma$ : Somatório de dados (se usado em relatórios ou análises quantitativas)
- $\leftrightarrow$ : Relações bidirecionais em diagramas ou fluxos de dados

## **Sumário**

1. Introdução
  - 1.1 Contextualização
  - 1.2 Objetivo do Artigo
2. Levantamento de Requisitos
  - 2.1 Brainstorming e Análise de Outros Projetos
  - 2.2 Requisitos Coletados
    - 2.2.1 Requisitos Funcionais
    - 2.2.2 Requisitos Não Funcionais
3. Modelo Relacional
  - 3.1 Introdução ao Modelo Relacional
  - 3.2 Construção do Modelo Relacional
4. Banco de Dados
  - 4.1 Geração de Scripts
  - 4.2 Comandos de Consulta (Relatórios)
5. Conclusão ou Considerações Finais
6. Referências
7. Glossário

# 1 INTRODUÇÃO

A coleta e o gerenciamento eficiente de dados são fundamentais para o planejamento e a execução de políticas públicas que atendam às necessidades da população. Nesse contexto, o EPISAFE surge como uma iniciativa estratégica da administração municipal, com o objetivo de coletar e armazenar informações dos moradores das cidades. O programa desempenha um papel crucial no suporte à tomada de decisões em diversas áreas, como a prevenção de doenças e a organização de levantamentos censitários.

Entre as principais aplicações do EPISAFE, destaca-se a prevenção à dengue, uma doença que representa um grave problema de saúde pública em diversas regiões do Brasil. Por meio do monitoramento dos dados coletados, é possível mapear áreas de risco e implementar ações proativas de combate ao mosquito transmissor. Além disso, o programa oferece uma base confiável para a realização de censos demográficos, permitindo à gestão municipal compreender melhor as características da população e direcionar recursos de forma mais eficiente.

Este artigo tem como objetivo analisar o funcionamento do EPISAFE, explorando suas contribuições para a gestão pública e destacando os benefícios de uma abordagem baseada em dados no enfrentamento de desafios urbanos. A discussão aborda os aspectos técnicos, as potencialidades e as limitações do programa, além de refletir sobre a importância da tecnologia no desenvolvimento de políticas públicas mais eficazes.

## **Episafe (desenvolvimento)**

O sistema EPISAFE, como descrito no trabalho, é um exemplo concreto de como a tecnologia pode contribuir de maneira significativa para a saúde pública e gestão urbana. Sua concepção envolveu um rigoroso levantamento de requisitos, incluindo análises de projetos similares e sessões de brainstorming, que ajudaram a identificar as principais necessidades dos usuários, como a capacidade de monitoramento geoespacial e a geração de relatórios detalhados.

Na fase de modelagem, a utilização de ferramentas como o Visual Paradigm e o Whimsical possibilitou a criação de um modelo relacional robusto e bem estruturado. Esse modelo foi fundamental para garantir a integridade e a eficiência no armazenamento e na consulta de dados. A implementação técnica utilizou a

ferramenta DBeaver para a execução de scripts SQL, assegurando que as operações no banco de dados fossem realizadas de forma ágil e precisa.

Os resultados obtidos com o EPISAFE destacam-se pela sua aplicabilidade prática, permitindo aos gestores de saúde monitorar focos de dengue, identificar padrões epidemiológicos e tomar decisões baseadas em dados confiáveis. Além disso, a escalabilidade e as funcionalidades de segurança implementadas asseguram que o sistema seja adequado para uso em diferentes contextos, tanto urbanos quanto rurais.

## **2 LEVANTAMENTO DE REQUISITOS**

### **2.1 Brainstorming e Análise de outros projetos**

O processo de levantamento de requisitos para o desenvolvimento do sistema EPISAFE envolveu a exploração e análise de projetos relacionados a bancos de dados voltados para o monitoramento de epidemias. Como referência, foi utilizado um sistema desenvolvido em São Miguel do Oeste, que forneceu uma base sólida para o planejamento inicial.

Os projetos analisados detalharam todo o fluxo de trabalho utilizado por outros acadêmicos na criação de seus softwares, incluindo representações visuais de modelos conceituais. Essas imagens foram especialmente úteis durante sessões de brainstorming realizadas entre os desenvolvedores do EPISAFE, permitindo um alinhamento mais preciso das ideias.

Com base nos exemplos observados, foi possível identificar eventos fundamentais no cotidiano de um Fiscal da Dengue, o que contribuiu significativamente para a formulação dos últimos detalhes do levantamento de requisitos. Esse processo garantiu uma compreensão aprofundada das necessidades práticas do sistema, facilitando o desenvolvimento de uma solução eficiente e alinhada à realidade operacional.



Figura 1 – Consultas SQL

```
1  -- 1) Relacionar todos as pessoas com idades entre 20 e 30 anos e do sexo feminino.
2  -- Ordenar o relatório pelo nome dos clientes em ordem decendente;
3  select
4      nome_pessoa,
5      DATE_PART ('year', AGE (data_nascimento)) as idade
6  from
7      caso_de_dengue cdd
8  where
9      sexo = 'F'
10     and data_nascimento between CURRENT_DATE - INTERVAL '30 years' and CURRENT_DATE - INTERVAL '20 years'
11  order by
12      nome_pessoa desc;
13
14  -- 2) Relacionar os casos registradas em meses ímpares de 2023 de clientes das cidades de São Miguel do Oeste e Descanso.
15  -- Ordene o relatório pela data do caso de forma ascendente;
16  select
17      *
18  from
19      caso_de_dengue cdd
20      join visita v on v.id = cdd.visita_id
21      join residencia r on r.id = v.id
22      join endereco e on e.id = r.endereco_id
23  where
24      e.cidade = 'São Miguel do Oeste'
25      or e.cidade = 'Descanso'
26  -- 3) Relacionar todas as residências das cidades de Maravilha, Descanso, Itapiranga e Guaraciaba que foram visitadas em 2024.
27  -- Ordene o relatório da cidade com mais visitas para a cidade com menos visitas;
28  select
29      Endereco.cidade AS cidade,
30      COUNT(*) AS total_visitas
31  from
```

Fonte: os autores.

Figura 2 – Consultas SQL

```
32  Visita
33  JOIN Residencia ON Visita.residencia_id = Residencia.id
34  JOIN Endereco ON Residencia.endereco_id = Endereco.id
35  where
36      date_part ('year', Visita.data_visita) = 2024
37      and Endereco.cidade IN (
38          'Maravilha',
39          'Descanso',
40          'Itapiranga',
41          'Guaraciaba'
42      )
43  group by
44      Endereco.cidade
45  order by
46      total_visitas DESC;
47
48  -- 4) Relacionar o nome da cidade, o total de casos por faixa etária. Relacionar da faixa etária com mais casos para a faixa com menos casos.
```

Fonte: os autores.

## 2.2 Requisitos Coletados

### Requisitos Funcionais:

- O sistema deve ser capaz de gerar relatórios sobre a incidência de casos de dengue por período, localização, idade, sexo, e gravidade.
- Deve permitir análises geoespaciais para identificar áreas de alta incidência e focos de mosquitos.
- Deve ser possível registrar e gerenciar focos de mosquitos, incluindo localização, data de detecção, e medidas de controle aplicadas.
- O sistema deve permitir a atualização e exclusão de registros de focos.
- O sistema deve permitir o cadastro de locais afetados por epidemias.
- Cada local deve possuir informações como nome, endereço, coordenadas geográficas e tipo de área (urbana/rural).

- Deve ser possível registrar casos de dengue, incluindo data de diagnóstico, local, idade do paciente, sexo, e gravidade do caso.
- O sistema deve permitir a atualização e exclusão de registros de casos
- O banco de dados deve armazenar informações sobre os pacientes, incluindo nome, idade, sexo, endereço e histórico de doenças.
- Deve ser possível associar pacientes a casos específicos de dengue.
- O sistema deve enviar notificações e alertas para autoridades de saúde sobre novos focos de mosquitos e aumentos significativos no número de casos.
- Deve permitir a configuração de critérios para disparo de alertas.
- O banco de dados deve ter mecanismos de autenticação e autorização para garantir que apenas usuários autorizados possam acessar e modificar os dados.

#### **Requisitos Não Funcionais:**

- O sistema deve ser capaz de processar e responder a consultas em tempo real ou com baixa latência, especialmente para operações de leitura frequentes.
- O banco de dados deve ser otimizado para suportar grandes volumes de dados sem degradação significativa de desempenho.
- O sistema deve ser capaz de escalar horizontalmente (adicionando mais servidores) e verticalmente (aumentando os recursos de hardware) conforme a quantidade de dados e o número de usuários cresce.
- Deve suportar a replicação de dados para balanceamento de carga e alta disponibilidade.
- Deve haver políticas de backup e recuperação para proteger contra perda de dados.
- O sistema deve ser fácil de manter e atualizar, com uma arquitetura modular que permita modificações sem impactar significativamente outras partes do sistema.
- Deve haver logs de auditoria para rastrear todas as operações realizadas no sistema.

## **3 MODELO RELACIONAL**

### **3.1 Introdução ao Modelo Relacional**

Com as informações necessárias em mãos para implementar o projeto EPISAFE, avançamos para a criação do modelo relacional. Esse modelo é responsável por processar os requisitos levantados e convertê-los em um diagrama estrutural que define a organização da nossa base de dados.

Tradicionalmente, o modelo relacional é desenvolvido após a construção de um modelo conceitual, que é uma representação mais abstrata e de alto nível do banco de dados. No entanto, optamos por combinar esses dois processos em uma única etapa para agilizar o desenvolvimento do projeto.

A criação do modelo relacional consiste em projetar visualmente a estrutura do banco de dados, o que inclui:

- **Definir tabelas** que representarão as principais entidades do sistema;
- **Estabelecer relacionamentos** entre as tabelas para garantir a integridade dos dados;
- **Especificar colunas** dentro de cada tabela, com base nos atributos das entidades;
- **Atribuir tipos de dados** adequados a cada coluna, garantindo precisão e eficiência no armazenamento das informações.

Esse processo é essencial para transformar os requisitos levantados em uma base de dados funcional e bem estruturada, que servirá de alicerce para o funcionamento eficiente do EPISAFE.

### 3.2 Construção do Modelo Relacional

Através do uso da ferramenta Visual Paradigm, fomos capazes de colocar nossas ideias em prática e de arquitetar o nosso modelo (além do Brainstorming com o Whimsical). Basicamente seguimos e priorizamos as 4 etapas abaixo para a criação do banco de dados.

1. Criação da tabela e sua colunas;
2. Tipagem correta das colunas;
3. Definição de Chaves Primárias;
4. Ligações da tabela;

## 4 BANCO DE DADOS

### 4.1 Geração de Scripts

A próxima etapa de desenvolvimento envolve a geração de scripts para construção do banco de dados físico. Os scripts de que estamos falando são comandos escritos em SQL (Structured Query Language) que informam ao SGDB (Database Management System) como criar e manipular nossos bancos de dados, tabelas e dados. Para realizar esse processo, utilizamos uma das funcionalidades da ferramenta Visual Paradigm, que é gerar um script de criação baseado no modelo relacional criado. Exportamos os comandos gerados para o Notepad e os copiamos para o DBeaver (a ferramenta que usamos para manipular o banco de dados).

### 4.2 Comandos de Consulta (Relatórios)

- 1) Relacionar todos as pessoas com idades entre 20 e 30 anos e do sexo feminino. Ordenar o relatório pelo nome dos clientes em ordem descendente;
- 2) Relacionar os casos registrados em meses ímpares de 2023 de clientes das cidades de São Miguel do Oeste e Descanso. Ordene o relatório pela data do caso de forma ascendente;

3) Relacionar todas as residências das cidades de Maravilha, Descanso, Itapiranga e Guaraciaba que foram visitadas em 2024. Ordene o relatório da cidade com mais visitas para a cidade com menos visitas;

4) Relacionar o nome da cidade, o total de casos por faixa etária. Relacionar da faixa etária com mais casos para a faixa com menos casos.

Figura 3 – Criação Tabela Endereço

```
1  CREATE table
2  |   Endereco (
3  |       id serial primary key,
4  |       cidade VARCHAR(50),
5  |       bairro VARCHAR(50),
6  |       logradouro VARCHAR(50),
7  |       observacao text
8  |   );
9
```

Fonte: os autores.

Figura 4 – Criação Tabela Área de atuação

```
10 CREATE table
11 |   Area_de_atuacao (
12 |       id serial primary key,
13 |       apelido VARCHAR(100) not null,
14 |       observacao TEXT
15 |   );
16
```

Fonte: os autores.

Figura 5 – Criação Tabela Área de atuação com endereços

```
17 create table
18 |   Area_de_atuacao_com_enderecos (
19 |       id serial primary key,
20 |       endereco_id integer not null references Endereco (id),
21 |       area_de_atuacao_id integer not null references Area_de_atuacao (id)
22 |   );
23
```

Fonte: os autores.

Figura 6 – Criação Tabela Residência

```
24 v create table
25 v   Residencia (
26     id serial primary key,
27     tipo_residencia INTEGER check (tipo_residencia IN (1, 2, 3)),
28     numero VARCHAR(10) not null,
29     ponto_de_referencia text,
30     endereco_id INTEGER references Endereco (id)
31 );
32
```

Fonte: os autores.

Figura 7 – Criação Tabela Usuário

```
33 v create table
34 v   Usuario (
35     id SERIAL primary key,
36     login VARCHAR(50) unique not null,
37     senha VARCHAR(100) not null
38 );
39
```

Fonte: os autores.

Figura 8 – Criação Tabela Agente

```
40 v create table
41 v   Agente (
42     id serial primary key,
43     nome_completo VARCHAR(100) not null,
44     data_nasc DATE not null,
45     sexo CHAR(1) check (sexo in ('F', 'M')),
46     telefone VARCHAR(15) not null,
47     cpf CHAR(11) UNIQUE not null,
48     endereco_id INTEGER references Endereco (id),
49     usuario_id INTEGER references Usuario (id)
50 );
51
```

Fonte: os autores.

Figura 9 – Criação Tabela Ciclo

```
52 create table
53     Ciclo (
54         id serial primary key,
55         nome VARCHAR(100) not null,
56         descricao TEXT,
57         data_inicio DATE not null,
58         data_fim DATE not null
59     );
60
```

Fonte: os autores.

Figura 10 – Criação Tabela Visita

```
61 v create table
62 v     Visita (
63         id SERIAL primary key,
64         residencia_id INTEGER references Residencia (id),
65         ciclo_id INTEGER references Ciclo (id),
66         descricao TEXT,
67         agente_id INTEGER not null references Agente (id),
68         data_visita DATE not null
69     );
70
```

Fonte: os autores.

Figura 11 – Criação Tabela Armadilha

```
71 v create table
72 v     Armadilha (
73         id serial primary key,
74         area_de_atuacao_id INTEGER references area_de_atuacao (id),
75         agente_id INTEGER not null references Agente (id),
76         tipo_armadilha VARCHAR(50),
77         data DATE not null,
78         descricao TEXT
79     );
80
```

Fonte: os autores.

Figura 12 – Criação Tabela Caso de dengue

```
81  ✓ create table
82  ✓      Caso_de_dengue (
83      |      id serial primary key,
84      |      nome_pessoa text not null,
85      |      sexo CHAR(1) check (sexo in ('F', 'M')),
86      |      cpf CHAR(11) unique not null,
87      |      data_nascimento DATE not null,
88      |      visita_id INTEGER references Visita (id)
89      |  );
90
```

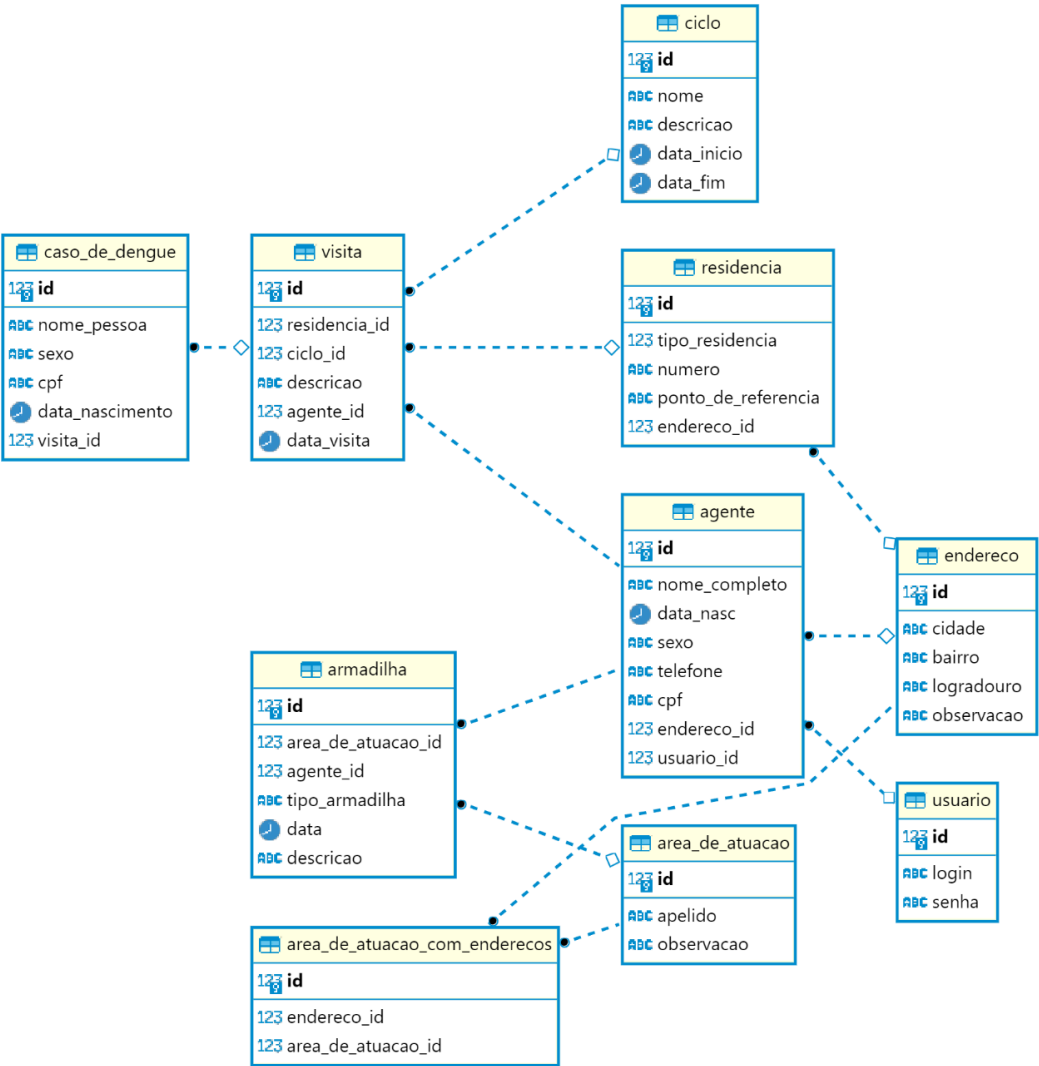
Fonte: os autores.

Figura 13 – Criação Tabela Cadastro de notificação

```
91  ✓ create table
92  ✓      Cadastro_de_notificacao (
93      |      id serial primary key,
94      |      visita_id INTEGER references Visita (id),
95      |      data_retorno DATE not null,
96      |      foco_dengue BOOLEAN not null,
97      |      descricao TEXT
98      |  );
```

Fonte: os autores.

Figura 14 – Diagrama



Fonte: os autores.



## CONCLUSÃO OU CONSIDERAÇÕES FINAIS

O projeto EPISAFE demonstra o potencial da tecnologia como aliada na gestão pública e no enfrentamento de desafios relacionados à saúde e bem-estar da população. Desde o levantamento de requisitos até a construção do modelo relacional e a implementação do banco de dados, cada etapa foi orientada para atender às demandas práticas do controle de epidemias, especialmente no combate à dengue.

A abordagem adotada permitiu a criação de um sistema robusto, capaz de coletar, organizar e analisar dados de forma eficiente, oferecendo insights valiosos para a tomada de decisões estratégicas. A integração de ferramentas como Visual Paradigm, Whimsical e DBeaver foi essencial para garantir a qualidade do desenvolvimento e a precisão na geração de relatórios personalizados, que podem subsidiar ações direcionadas por autoridades de saúde e gestores municipais.

Além disso, o EPISAFE reafirma a importância do uso de metodologias ágeis e colaborativas no desenvolvimento de soluções tecnológicas. A análise de projetos similares e as sessões de brainstorming contribuíram para a inovação e a adequação do sistema às necessidades reais do cotidiano de Fiscais da Dengue.

Com isso, o EPISAFE não apenas atende aos objetivos propostos, mas também estabelece uma base para futuras expansões e aprimoramentos, demonstrando o valor de sistemas de informação como pilares para a construção de políticas públicas mais eficazes e sustentáveis.

## Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6023: Informação e documentação - Referências - Elaboração. Rio de Janeiro: ABNT, 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 14724: Informação e documentação - Trabalhos acadêmicos - Apresentação. Rio de Janeiro: ABNT, 2011.

BÁRBARA BATISTTELLA DE OLIVEIRA; LEANDRO BERTOCCHI; LEONARDO AGOSTINI COSTA; IRAÊ ERVIN GRUBER DA SILVA; JEFFERSON ALAN SCHMIDT LUDWIG. EPISAFE. Trabalho acadêmico apresentado ao Curso de Engenharia de Software I, Programação II e Banco de Dados II, UNOESC - São Miguel do Oeste, 2024. Disponível em: [https://github.com/BarbaraBatisttella/EpiSafe-controle-de-epidemias/blob/370a59c46/episafe\\_relatorios.sql](https://github.com/BarbaraBatisttella/EpiSafe-controle-de-epidemias/blob/370a59c46/episafe_relatorios.sql). Acesso em: 21 nov. 2024.

VISUAL PARADIGM. Ferramenta para modelagem UML e banco de dados. Disponível em: <https://www.visual-paradigm.com/>. Acesso em: 21 nov. 2024.

DBeaver. Gerenciador de banco de dados. Disponível em: <https://dbeaver.io/>. Acesso em: 21 nov. 2024.

GITHUB. Repositório EpiSafe. Disponível em: <https://github.com/BarbaraBatisttella/EpiSafe-controle-de-epidemias>. Acesso em: 21 nov. 2024.

## **Glossário**

- **ABNT (Associação Brasileira de Normas Técnicas):** Organização responsável por normatizar documentos técnicos e acadêmicos no Brasil, garantindo padronização e qualidade.
- **Banco de Dados:** Estrutura organizada para armazenamento e gerenciamento de informações, permitindo fácil acesso, consulta e manipulação de dados.
- **Brainstorming:** Técnica de geração de ideias em grupo para levantamento de requisitos e soluções criativas.
- **DBeaver:** Ferramenta de gerenciamento de banco de dados usada para manipulação e consulta de dados.
- **DBMS (Database Management System):** Sistema responsável por gerenciar, organizar e controlar o acesso a bancos de dados.
- **EPISAFE:** Sistema de monitoramento e controle de epidemias, com foco na coleta de dados para prevenção de doenças como a dengue.
- **Modelo Relacional:** Estrutura lógica de organização de dados em tabelas interligadas, utilizada em sistemas de bancos de dados.
- **Requisitos Funcionais:** Conjunto de funções específicas que um sistema deve realizar, como geração de relatórios e registro de informações.
- **Requisitos Não Funcionais:** Características de qualidade do sistema, como desempenho, escalabilidade e segurança.
- **SQL (Structured Query Language):** Linguagem de consulta estruturada usada para criar, manipular e consultar dados em bancos de dados.
- **Visual Paradigm:** Ferramenta de modelagem usada para criar diagramas e estruturar sistemas, incluindo modelos relacionais.
- **Whimsical:** Ferramenta de brainstorming e criação de diagramas usada no desenvolvimento inicial do projeto.
- **UNOESC (Universidade do Oeste de Santa Catarina):** Instituição de ensino superior onde o projeto foi desenvolvido.

## **Índice**

- 1. Introdução**
  - 1.1 Contextualização
  - 1.2 Objetivo do Artigo
- 2. Levantamento de Requisitos**
  - 2.1 Brainstorming e Análise de Outros Projetos
  - 2.2 Requisitos Coletados
    - 2.2.1 Requisitos Funcionais
    - 2.2.2 Requisitos Não Funcionais
- 3. Modelo Relacional**
  - 3.1 Introdução ao Modelo Relacional
  - 3.2 Construção do Modelo Relacional
- 4. Banco de Dados**
  - 4.1 Geração de Scripts
  - 4.2 Comandos de Consulta (Relatórios)
- 5. Conclusão ou Considerações Finais**
- 6. Referências**