



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA BIOMÉDICA  
CURSO DE ENGENHARIA BIOMÉDICA

JOSÉ FILIPE SILVA DE ANDRADE

**PROPOSTA DE SISTEMA DE SUPORTE PARA APERFEIÇOAMENTO DO  
ATENDIMENTO A CASOS DE ENVENENAMENTO POR ANIMAIS  
PEÇONHENTOS EM PERNAMBUCO**

Recife  
2024

JOSÉ FILIPE SILVA DE ANDRADE

**PROPOSTA DE SISTEMA DE SUPORTE PARA APERFEIÇOAMENTO DO  
ATENDIMENTO A CASOS DE ENVENENAMENTO POR ANIMAIS  
PEÇONHENTOS EM PERNAMBUCO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Biomédica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Engenheira Biomédica.

**Orientadora:** Prof.<sup>a</sup> Dr.<sup>a</sup> Cristine Martins Gomes de Gusmão

Recife

2024

Ficha de identificação da obra elaborada pelo autor,  
através do programa de geração automática do SIB/UFPE

Andrade, José Filipe Silva de.

Proposta de sistema de suporte para aperfeiçoamento do atendimento a casos de envenamento por animais peçonhentos em Pernambuco / José Filipe Silva de Andrade. - Recife, 2024.

71 p.

Orientador(a): Cristine Martins Gomes de Gusmão

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Biomédica - Bacharelado, 2024.

1. Acidentes por animais peçonhentos. 2. Saúde pública. 3. Soroterapia. 4. Pernambuco. 5. Informática em saúde. I. Gusmão, Cristine Martins Gomes de. (Orientação). II. Título.

620 CDD (22.ed.)

JOSÉ FILIPE SILVA DE ANDRADE

**PROPOSTA DE SISTEMA DE SUPORTE PARA APERFEIÇOAMENTO DO  
ATENDIMENTO A CASOS DE ENVENENAMENTO POR ANIMAIS  
PEÇONHENTOS EM PERNAMBUCO**

Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia  
Biomédica da Universidade Federal de  
Pernambuco, como requisito parcial para  
obtenção do título de Engenheira  
Biomédica.

Aprovado em: 30/10/2024

**BANCA EXAMINADORA**

---

Prof.<sup>a</sup> Dr.<sup>a</sup> Cristine Martins Gomes de Gusmão (Orientadora)  
Universidade Federal de Pernambuco

---

Prof. Dr. Josenildo Costa da Silva (Examinador externo)  
Instituto Federal do Maranhão

# Agradecimentos

A Deus, família, amigos, professores e ao meu cachorro.

# Resumo

Os acidentes por animais peçonhentos (AAPs) representam uma preocupação significativa de saúde pública em áreas rurais e regiões tropicais, onde o acesso ao tratamento é limitado. De acordo com dados do SINITOX, os AAPs são a segunda maior causa de intoxicação humana no país, sendo superados apenas por intoxicações medicamentosas. Entre 2018 e 2022, o Nordeste registrou mais de 478 mil casos de AAPs, com destaque para picadas de escorpiões. Em Pernambuco, entre 2019 e 2023, foram registrados 106 mil casos de acidentes por animais peçonhentos, dos quais menos de 3% receberam tratamento por soroterapia. Ademais, a subnotificação e a dificuldade de acesso à soroterapia continuam a agravar a situação. Este trabalho propõe o desenvolvimento de um sistema de suporte para otimizar o atendimento a vítimas de envenenamento por animais peçonhentos no estado de Pernambuco. A solução foi construída em *Python* e hospedada no *Streamlit Cloud*, com a finalidade de facilitar a localização dos centros de referência em soroterapia mais próximos, considerando o tipo de soro necessário e a cidade de origem do paciente. A aplicação do sistema é uma demonstração de como a engenharia biomédica pode ser empregada para melhorar a eficiência dos serviços de saúde, especialmente em áreas distantes dos grandes centros urbanos.

**Palavras-chave:** acidentes por animais peçonhentos; saúde pública; soroterapia; Pernambuco; informática em saúde.

# Abstract

Envenomations by venomous animals represent a significant public health concern in rural and tropical regions, where access to treatment is limited. According to SINITOX data, envenomations are the second leading cause of human poisoning in the country, surpassed only by drug intoxications. Between 2018 and 2022, the Northeast region recorded over 478,000 cases of envenomations, with scorpion stings being the most prominent. In Pernambuco, between 2019 and 2023, 106,000 cases of envenomations were registered, yet less than 3% received antivenom treatment. Additionally, underreporting and the difficulty of accessing antivenom treatment continue to worsen the situation. This work proposes the development of a support system to optimize the care of envenomation victims in the state of Pernambuco. The solution was built using Python and hosted on Streamlit Cloud, aiming to facilitate the identification of the nearest antivenom centers, based on the type of antivenom needed and the patient's city of origin. The system's application demonstrates how biomedical engineering can be employed to improve the efficiency of healthcare services, particularly in underdeveloped areas far from major urban centers.

**Keywords:** envenomations; public health; antivenom; Pernambuco; health informatics.

# Lista de Figuras

Figura 2.1 – Animais peçonhentos	18
Figura 2.2 – Processo de produção de soro antiofídico	21
Figura 2.2 – Linha do tempo do SUS	23
Figura 2.3 – Regiões de saúde em Pernambuco, 2013	24
Figura 2.4 – Distribuição de notificações por estado do nordeste	25
Figura 2.5 – Taxa de oferta de soroterapias por estado do nordeste	26
Figura 2.6 – Representação da latitude e longitude	32
Figura 3.1 – Opções de downloads no IBGE	35
Figura 3.2 – Tabela BR_localidades_2010_v1 no Microsoft Access	36
Figura 3.3 – Tabela latitude-longitude-cidades no Google Colab	37
Figura 3.4 – Importação de bibliotecas	37
Figura 3.5 – Carregamento dos arquivos	38
Figura 3.6 – Cálculo das distâncias e armazenamento de resultados	39
Figura 3.7 – Criação do DataFrame final	40
Figura 3.8 – Criação do DataFrame final	40
Figura 3.9 – Informações do DataFrame final	41
Figura 3.10 – Caixas de seleções de animal e soro	42
Figura 3.10 – Filtro no DataFrame	43
Figura 3.11 – Plotagem do mapa com Folium	43
Figura 3.11 – Mensagem de hibernação do Streamlit Cloud	44
Figura 3.12 – Mapa online de tratamento por soroterapia de São Paulo	46
Figura 3.13 – Geoprocessamentos realizados pelo QGIS no Rio Grande do Sul	47
Figura 4.1 – Tela inicial da aplicação em um monitor	49
Figura 4.2 – Filtros de busca	49
Figura 4.3 – Exemplo de resultado	50
Figura 4.4 – Representação da aplicação no celular	51
Figura 4.5 – Representação da aplicação no smartwatch	51
Figura C.1 – Ficha de notificações de AAP - página 1	61
Figura D.1 – Tabela de hospitais com soro antibotrópico	63
Figura D.2 – Tabela de hospitais com soro anticrotálico	63
Figura D.3 – Tabela de hospitais com soro antielapídico	64
Figura D.4 – Tabela de hospitais com soro antiescorpiônico	64

Figura D.5 – Tabela de hospitais com soro antiaracnídico	65
Figura D.6 – Tabela de hospitais com soro antilaquético	65
Figura D.7 – Tabela de hospitais com soro antiloxoscélico	65

# Lista de Tabelas

Tabela 2.1 – Relação entre animal, tipo de veneno e soro	21
Tabela 3.1 – Comandos e descrições do Streamlit	30
Tabela A.1 – Neglected tropical diseases	59
Tabela B.1 – Hospitais de Referência de Atendimento - Pernambuco	60
Tabela F.1 – Tabela de teste de GUI	70

# Lista de Abreviaturas e Siglas

AAPs	Acidentes por Animais Peçonhentos
API	<i>Application Programming Interface</i>
CIATox-PE	Centro de Informação e Assistência Toxicológica de Pernambuco
CLI	Interface de Linha de Comando
CSV	<i>Comma-Separated Values</i>
DATASUS	Departamento de Informática do Sistema Único de Saúde
DTNs	Doenças Tropicais Negligenciadas
GERES	Gerências Regionais de Saúde
GPU	Unidade de Processamento Gráfico
GPS	<i>Global Positioning System</i>
GUI	Interface Gráfica de Usuário
HTML	<i>Hypertext Markup Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
KML	<i>Keyhole Markup Language</i>
LNC	Lista de Notificação Compulsória
MDB	<i>Microsoft Database</i>
NS	Número de soroterapias
OMS	Organização Mundial da Saúde
PDF	<i>Portable Document Format</i>
RAS	Rede de Atenção à Saúde
SHP	<i>Shapefile Format</i>
SINAN	Sistema de Informação de Agravos de Notificação
SINITOX	Sistema Nacional de Informações Tóxico-Farmacológicas
SUS	Sistema Único de Saúde
TN	Total de notificações
TOS	Taxa de Oferta de Soroterapia
UBS	Unidade Básica de Saúde
VSCoDe	<i>Visual Studio Code</i>
WGS-84	<i>World Geodetic System 1984</i>

# Sumário

<b>1. Introdução</b>	<b>14</b>
1.1. Objetivos	15
1.2. Estrutura do trabalho	16
<b>2. Fundamentação teórica</b>	<b>17</b>
2.1. Animais Peçonhentos	17
2.1.1. Envenenamento e Antídoto	18
2.2. Sistema Único de Saúde (SUS)	22
2.2.1. Centros de Referência em Soroterapia em Pernambuco	23
2.2.1.1. Demanda em Pernambuco	24
2.2.1.2 Centro de Informação e Assistência Toxicológica de Pernambuco (CIATox-PE)	26
2.3. Tecnologias do Sistema	27
2.3.1. Google Colab	27
2.3.2. Linguagem de Programação Python	28
2.3.3. Streamlit	29
2.3.4. Controle de Versão com Git e GitHub	30
2.3.5. Hospedagem e Deploy	31
2.4. Coordenadas Geográficas	31
2.4.1. Geodesic do Geopy: Cálculo da Distância Geodésica	32
<b>3. Metodologia</b>	<b>34</b>
3.1. Análise de Dados	34
3.1.1. Coleta de Dados	34
3.1.1.1. Dados dos Hospitais com Soroterapia	34
3.1.1.2. Dados de Localização dos Municípios Pernambucanos	35
3.1.2. Análise e Processamento de Dados	37
3.2. Sistema de Mapeamento	41
3.2.1. Visão do Código-Fonte	42
3.2.2. Deploy do Sistema	44
3.3. Avaliação do Sistema	45
3.3.1. Casos de Testes	45
3.3.2. Sistemas Análogos	45
<b>4. Resultados</b>	<b>48</b>
4.1. Interface do Usuário	48
4.2. Resultado dos Testes	51
<b>5. Conclusão</b>	<b>52</b>
<b>Referências</b>	<b>54</b>
<b>Anexo A - Doenças Tropicais Negligenciadas</b>	<b>58</b>

<b>Anexo B - Hospitais de Referência para AAPs em Pernambuco</b>	<b>59</b>
<b>Anexo C - Ficha de Notificação para Acidentes por Animais Peçonhentos</b>	<b>60</b>
<b>Anexo D - Tabelas de Hospitais por Soro</b>	<b>62</b>
<b>Anexo E - Código-Fonte do Sistema de Mapeamento</b>	<b>65</b>
<b>Anexo F - Modelo de Teste de GUI</b>	<b>69</b>

# 1. Introdução

Os acidentes por animais peçonhentos (AAPs) representam um grave problema de saúde pública nos países tropicais, incluindo o Brasil, onde a fauna é diversificada e bastante distribuída. Os AAPs envolvem interações com diversas espécies de animais venenosos, como serpentes, aranhas, escorpiões, abelhas e lagartas (BRASIL, 2001). De acordo com dados do Sistema Nacional de Informações Tóxico-Farmacológicas (SINITOX) os AAPs são a segunda maior causa de intoxicação humana no Brasil, ficando atrás apenas dos casos de intoxicação por medicamentos (MACHADO, 2016). Essa incidência se relaciona diretamente com as condições socioeconômicas, ambientais e a qualidade de vida das populações mais vulneráveis.

A gravidade dos AAPs varia de acordo com a espécie envolvida, a quantidade de veneno inoculada, o local da picada e a rapidez com que o tratamento é administrado (BRASIL, 2001). No entanto, os impactos desses acidentes vão além das lesões imediatas, podendo resultar em sequelas permanentes, como amputações, comprometimento de funções motoras e até levar ao óbito (MATOS et al., 2020). Segundo Cunha (2024), com base em dados obtidos por meio da ferramenta *Tabnet* do Departamento de Informática do Sistema Único de Saúde (DATASUS), entre 2018 e 2022, foram registrados 478.793 casos de acidentes com animais peçonhentos no Nordeste, com uma média de 39.899 casos por ano. A maioria dos casos foi causada por picadas de escorpião. O Nordeste foi a segunda região com mais casos, ficando atrás apenas do Sudeste, que registrou 522.169 casos. Além disso, muitas vítimas, especialmente em áreas subdesenvolvidas, não conseguem ter acesso rápido ao tratamento adequado, agravando ainda mais as complicações dos acidentes.

Um dos desafios enfrentados pelos profissionais de saúde é a subnotificação de casos. Em muitas regiões, a verdadeira magnitude dos acidentes é mascarada por falhas no preenchimento adequado das fichas de notificação e pela falta de registro de dados epidemiológicos (MACHADO, 2016). A alta prevalência desses acidentes em países tropicais, combinada com as dificuldades no acesso ao tratamento, levou a Organização Mundial da Saúde (OMS) a incluir esses acidentes, especialmente os acidentes ofídicos (cobras), na lista de Doenças Tropicais

Negligenciadas (DTNs) em 2009 (MACHADO, 2016) (ver Anexo A). Além disso, o elevado número de notificações registradas no Sistema de Informação de Agravos de Notificação (SINAN) resultou na inclusão desse agravo na Lista de Notificação Compulsória (LNC) do Brasil em agosto de 2010 (BRASIL, 2016).

Ademais, outro fator que agrava a situação é a dificuldade que muitas vítimas enfrentam para localizar os hospitais que dispõem de soroterapia, o único tratamento eficaz para esses envenenamentos. Entre 2019 e 2023, o estado de Pernambuco registrou 106 mil casos de acidentes por animais peçonhentos, mas menos de 3% desses casos receberam tratamento por soroterapia (CAMPOS et al., 2024). A falta de informação clara e acessível sobre onde encontrar o antiveneno adequado é um obstáculo, especialmente para populações longe dos centros urbanos. Em muitos casos, as vítimas não sabem quais unidades de saúde estão equipadas para fornecer o tratamento, o que resulta em atrasos que podem ser fatais. Esse cenário é agravado pela má distribuição de centros de referência para soroterapia, que estão concentrados em regiões urbanas, enquanto as áreas mais afetadas por envenenamentos estão frequentemente distantes desses locais (CAMPOS et al., 2024).

Ainda segundo Campos et al. (2024), a análise da distribuição geográfica dos casos de AAPs no Nordeste, realizada nos últimos anos (2019-2023), revelou uma significativa concentração nos estados da Bahia, Pernambuco e Alagoas. Essa concentração pode estar relacionada a fatores populacionais e ambientais específicos desses estados, como a densidade populacional, o clima e a presença de habitats que favorecem esses animais. Este trabalho se concentra no estado de Pernambuco e propõe um sistema de suporte para aprimorar o atendimento a casos de envenenamento por AAPs. O sistema é apresentado na forma de um *website*, construído na linguagem *Python* e hospedado no *Streamlit Cloud*<sup>1</sup>, que exibe os Centros de Referência em Soroterapia em Pernambuco, indicando qual é o centro mais próximo de cada município para cada tipo de soro.

---

<sup>1</sup> Endereço do sistema: <https://sorosantivenompe.streamlit.app/>

## 1.1. Objetivos

O objetivo principal é auxiliar o aperfeiçoamento do atendimento a casos de envenenamento por animais peçonhentos em Pernambuco, a partir do desenvolvimento de um sistema de suporte baseado em localização de centros de referência que oferecem soroterapia mais próximo da localização de origem do paciente, para cada tipo de soro. Os objetivos incluem:

- Coleta de dados relevantes para aplicação;
- Construção de uma base de dados contendo informações sobre os centros de referência que disponibilizam soros antiveneno mais próximos de cada município de Pernambuco (para cada tipo de soro);
- Desenvolver uma plataforma interativa que permita aos usuários identificar o hospital mais próximo que oferece o soro adequado ao tipo de acidente;
- Implementar funcionalidades de filtragem para facilitar o uso do sistema por pelo usuário;
- Aplicar responsividade da aplicação entre diferentes dispositivos;
- Fornecer base e sugestão de ideias para trabalhos futuros nesta área.

## 1.2. Estrutura do trabalho

O trabalho está dividido em cinco capítulos. No Capítulo 1, foi apresentado o contexto no qual o estudo está inserido, incluindo a relevância dos acidentes por animais peçonhentos em Pernambuco, a justificativa para a criação de um sistema de suporte ao atendimento e os objetivos gerais e específicos do trabalho.

No Capítulo 2, é apresentada a fundamentação teórica relacionada às ferramentas e conceitos utilizados no estudo, como informações sobre animais peçonhentos, o Sistema Único de Saúde (SUS), além das tecnologias utilizadas como Python para o desenvolvimento do sistema e conceitos de coordenadas geográficas.

O Capítulo 3 aborda a metodologia do estudo, detalhando o processo de coleta e organização de dados, o desenvolvimento do sistema de mapeamento dos hospitais e como foi realizado a avaliação do sistema.

Os resultados são apresentados no Capítulo 4, onde se discute a eficiência do sistema e apresenta a aplicação final. No Capítulo 5, são feitas as considerações finais e sugestões para futuros estudos que possam aprimorar o sistema e assim contribuir para o avanço de pesquisas na área.

## 2. Fundamentação teórica

Neste capítulo, serão abordados conceitos técnicos que ajudarão na compreensão do projeto.

### 2.1. Animais Peçonhentos

Em Brasil (2024), há uma introdução da definição de animais peçonhentos, diferenciando-os de animais venenosos. Segundo ele, lê-se:

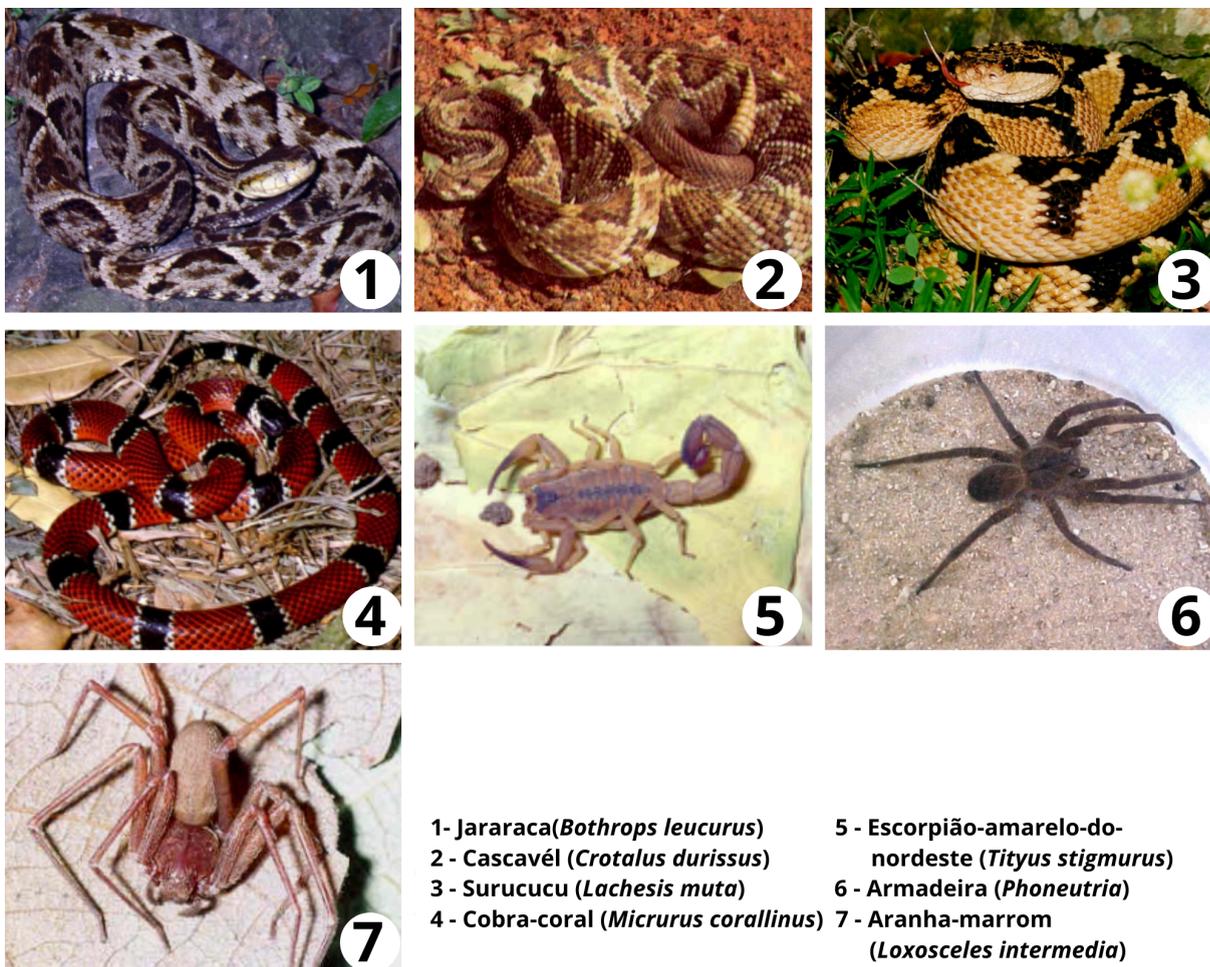
Os animais venenosos podem armazenar os componentes tóxicos obtidos do ambiente ou de outros organismos, e utilizam tais toxinas como mecanismo de defesa contra predação. Essas toxinas somente desempenham sua função ao serem ingeridas pelo organismo predador. Já os animais peçonhentos têm, adicionalmente, a capacidade de injetar ativamente esses compostos, tanto em presas (para predação) quanto em predadores (para defesa). Para essa injeção ativa das toxinas, os animais peçonhentos utilizam aparelhos inoculadores, que podem ser dentes especializados (presas), ferrões, quelíceras, cerdas urticantes, esporões etc (BRASIL, 2024).

Compreender essa distinção é essencial para a notificação adequada de acidentes no SINAN. Animais como carrapatos, mosquitos e morcegos, apesar de tóxicos, não são classificados como peçonhentos, o que influencia nas estratégias de vigilância e saúde pública.

Entre os principais animais envolvidos nos AAPs estão serpentes como jararacas (gêneros *Bothrops* e *Bothrocophias*), cascavéis (*Crotalus durissus*), surucucu-pico-de-jaca (*Lachesis muta*) e corais-verdadeiras (gêneros *Micrurus* e *Leptomicrurus*); escorpiões como o escorpião-amarelo (*Tityus serrulatus*), o escorpião-marrom (*Tityus bahiensis*), o escorpião-amarelo-do-nordeste (*Tityus stigmurus*) e o escorpião-preto-da-amazônia (*Tityus obscurus*); aranhas como a aranha-marrom (*Loxosceles*), a armadeira (*Phoneutria*) e a viúva-negra (*Latrodectus*); além das abelhas africanizadas (*Apis*) e lagartas como a taturana

(*Lonomia*). Outros animais, como lacraias, arraias e bagres, também podem causar acidentes graves (BRASIL, 2024).

Figura 2.1 – Animais peçonhentos



Fonte: (MEDEIROS, 2002) adaptado pelo próprio autor

### 2.1.1. Envenenamento e Antídoto

As toxinas dos animais peçonhentos variam em seus mecanismos de ação e são classificadas, segundo Brasil (2001), principalmente como:

- **Hemotoxinas:** Atacam as células sanguíneas e vasos, levando a hemorragias, necrose tecidual e inflamação. Essas toxinas são comuns em serpentes do gênero *Bothrops* (jararaca), *Crotalus* (cascavel) e *Lachesis* (surucucu).

- **Neurotoxinas:** Interferem no sistema nervoso, bloqueando a transmissão de impulsos nervosos, o que pode causar paralisia muscular e respiratória. Essas toxinas são produzidas por serpentes como a *Crotalus* (cascavel), *Micrurus* (coral), a aranha *Phoneutria* (armadeira) e os escorpiões do gênero *Tityus*.
- **Miotoxinas:** Afetam diretamente os músculos, causando destruição muscular (rabdomiólise) e, em casos graves, insuficiência renal. As miotoxinas são encontradas no veneno de serpentes do gênero *Lachesis* (surucucu).
- **Necrótico-hemolíticos:** Toxinas que provocam destruição de tecidos (necrose) e hemólise, rompendo as células vermelhas do sangue. Essas toxinas são características do veneno da aranha *Loxosceles* (aranha-marrom).

O processo de produção dos soros antiveneno segue um protocolo comum, independentemente do tipo de veneno, porém é ajustado para cada tipo de toxina. Com base em Silva (2008), o procedimento básico foi descrito em um fluxo de 4 etapas, sendo elas respectivamente:

1. **Imunização de Cavalos:** Pequenas doses de veneno são injetadas em cavalos de maneira controlada, provocando uma resposta imunológica que gera a produção de anticorpos específicos para neutralizar o veneno. Cada cavalo é imunizado com o veneno específico de serpentes, aranhas ou escorpiões. Ao longo de várias semanas, os animais desenvolvem imunidade a essas toxinas.
2. **Coleta de Plasma:** Após a fase de imunização, o sangue dos cavalos é colhido e, por meio da centrifugação, o plasma — a parte líquida do sangue onde se encontram os anticorpos — é separado dos outros componentes sanguíneos. A centrifugação envolve a rotação rápida do sangue em tubos especiais, aplicando uma força centrífuga que separa o plasma dos glóbulos vermelhos e outros componentes mais pesados. Essa etapa é crítica para garantir que o soro final seja rico em anticorpos específicos.
3. **Purificação:** O plasma coletado passa por um processo de purificação, onde técnicas como a separação magnética podem ser usadas para isolar as proteínas desejadas, ou seja, os anticorpos que combatem as toxinas.

Durante esse processo, impurezas e proteínas não desejadas são removidas, garantindo que o produto final seja seguro e eficaz para uso humano. Esse tratamento também diminui o risco de reações alérgicas quando o soro é administrado.

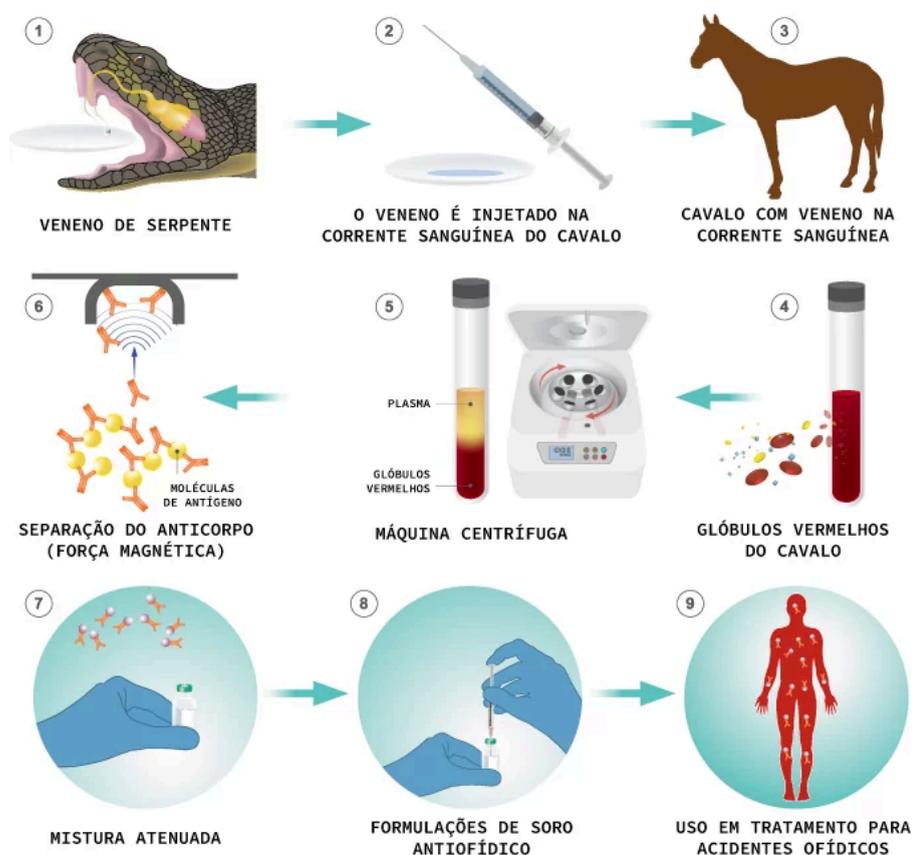
4. **Neutralização das Toxinas:** Cada tipo de soro antiveneno é projetado para neutralizar os efeitos específicos das toxinas que foram utilizadas no processo de imunização. Assim, temos:

- **Antibotrópico** para o veneno da jararaca (hemotoxina).
- **Anticrotálico** para o veneno da cascavel (hemotoxina e neurotoxina).
- **Antilaquéutico** para o veneno da surucucu (hemotoxina e miotoxina).
- **Antielapídico** para o veneno da cobra coral (neurotoxina).
- **Antiloxoscélico** para o veneno da aranha-marrom (necrótico-hemolítico).
- **Antiaracnídico** para o veneno da armadeira (neurotóxico) e dos escorpiões *Tityus spp.* (neurotóxico).

Ainda há o antiescorpiônico que é utilizado para tratar, especialmente, envenenamentos por algumas espécies de escorpiões, como os do gênero *Tityus*, que possuem venenos neurotóxicos.

A Figura 2.1 apresenta o processo de produção de soro antiofídico para exemplificar o que foi descrito. Enquanto a Tabela 2.1 apresenta a relação entre os animais peçonhentos com seus respectivos tipos de veneno e soro indicado.

Figura 2.2 – Processo de produção de soro antiofídico



Fonte: (IMUNOLOGIA, 2023)

Tabela 2.1 – Relação entre animal, tipo de veneno e soro

Animal Peçonhento	Tipo de Veneno	Soro Indicado
Serpente ( <i>Bothrops</i> - Jararaca)	Hemotoxina	Antibotrópico
Serpente ( <i>Crotalus</i> - Cascavel)	Hemotoxina e neurotoxina	Anticrotálico
Serpente ( <i>Lachesis</i> - Surucucu)	Hemotoxina e miotoxina	Antilaquético
Serpente ( <i>Micrurus</i> - Coral)	Neurotoxina	Antielapídico
Aranha ( <i>Loxosceles</i> - Aranha-marrom)	Necrótico-hemolítico	Antiloxoscélico

Aranha ( <i>Phoneutria</i> - Aranha-armadeira)	Neurotóxico	Antiaracnídico
Escorpião ( <i>Tityus spp.</i> )	Neurotóxico	Antiaracnídico / Antiescorpiônico

Fonte: Próprio autor

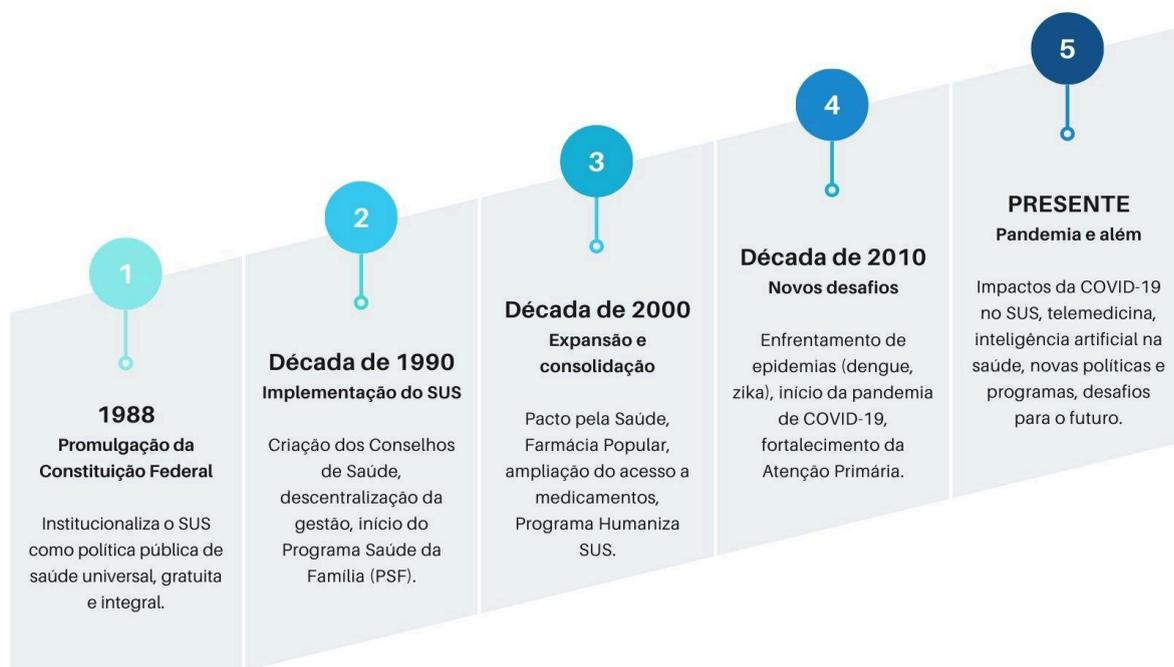
## 2.2. Sistema Único de Saúde (SUS)

O Sistema Único de Saúde (SUS) é o sistema público de saúde do Brasil, criado pela Constituição Federal de 1988, com o objetivo de garantir o acesso universal, integral e igualitário à saúde para toda a população. Conforme o artigo 196 da Constituição Federal, "a saúde é direito de todos e dever do Estado [...]" (BRASIL, 1988, art. 196). O SUS é responsável pela gestão de diversos serviços de saúde, que vão desde a atenção primária até a alta complexidade, além de ser o principal fornecedor de tratamentos especializados, como a soroterapia para AAPs (PAIM et al., 2015).

A implantação do SUS começou no início da década de 1990, após a promulgação da Lei Orgânica da Saúde (Lei n. 8.080, de 19 de setembro de 1990), complementada pela Lei n. 8.142, de 28 de dezembro de 1990. Essas leis definem a organização do sistema, sua estrutura, financiamento e as responsabilidades dos entes federados na prestação de serviços de saúde à população (PAIM et al., 2015).

O SUS é gerido de forma descentralizada pelos três níveis de governo: federal, estadual e municipal. Essa estrutura organizacional permite uma gestão mais próxima da realidade local, facilitando a adaptação das políticas de saúde às necessidades da população. Os princípios e diretrizes que orientam o funcionamento do SUS são fundamentados na equidade, universalidade e integralidade, assegurando que todos os cidadãos tenham acesso aos serviços de saúde de maneira justa e completa. O financiamento do SUS provém de diversas fontes, sendo o Fundo Nacional de Saúde uma das principais, composto por impostos e contribuições sociais, além de emendas parlamentares, o que reforça a necessidade de um sistema eficiente para atender a demanda da saúde pública no Brasil (PAIM et al., 2015).

Figura 2.2 – Linha do tempo do SUS



Fonte: Próprio autor

### 2.2.1. Centros de Referência em Soroterapia em Pernambuco

Em Pernambuco, existem 14 unidades de saúde que são referência para o atendimento de acidentes por animais peçonhentos (BRASIL, 2023). As informações destas unidades podem ser consultadas no Anexo B. Esses centros de referência são responsáveis por fornecer atendimento especializado e disponibilizar soros antivenenos para o tratamento dos casos graves de envenenamentos causados por cobras, escorpiões e aranhas. A distribuição dos soros é coordenada pela Secretaria de Saúde de Pernambuco.

A regionalização em Pernambuco é fundamentada no Decreto nº 7.508, de 28 de junho de 2011, que estabelece diretrizes para a organização da Rede de Atenção à Saúde (RAS) no Brasil. Esse decreto reforça a importância de uma rede regionalizada e hierarquizada, na qual os serviços de saúde são organizados em níveis de complexidade (BRASIL, 2011). As RAS são responsáveis por realizar o fluxo de referência e contrarreferência entre as unidades de saúde dentro de seus

limites geográficos, permitindo que os pacientes sejam acompanhados desde as Unidades Básicas de Saúde (UBS), a porta preferencial de entrada do sistema, até serviços de alta complexidade em outros entes federativos (SOARES et. al., 2019). No caso, para tratamento por soroterapia, as UBS podem encaminhar pacientes para os centros de referência especializados.

A rede de atendimento em Pernambuco é estruturada de forma a garantir que cada uma das 12 Gerências Regionais de Saúde (GERES) do estado tenha pelo menos um hospital de referência para o atendimento de acidentes por animais peçonhentos (BRASIL, 2023). Isso garante que a população tenha acesso rápido ao tratamento adequado independentemente da localização geográfica. A Figura 2.3 ilustra as regiões de saúde em Pernambuco, sendo as regiões Recife I e Petrolina VII as únicas com mais de um hospital de referência (ver Anexo B). Vale destacar que o município do arquipélago de Fernando de Noronha pertence à região Recife I.

*Figura 2.3 – Regiões de saúde em Pernambuco, 2013*

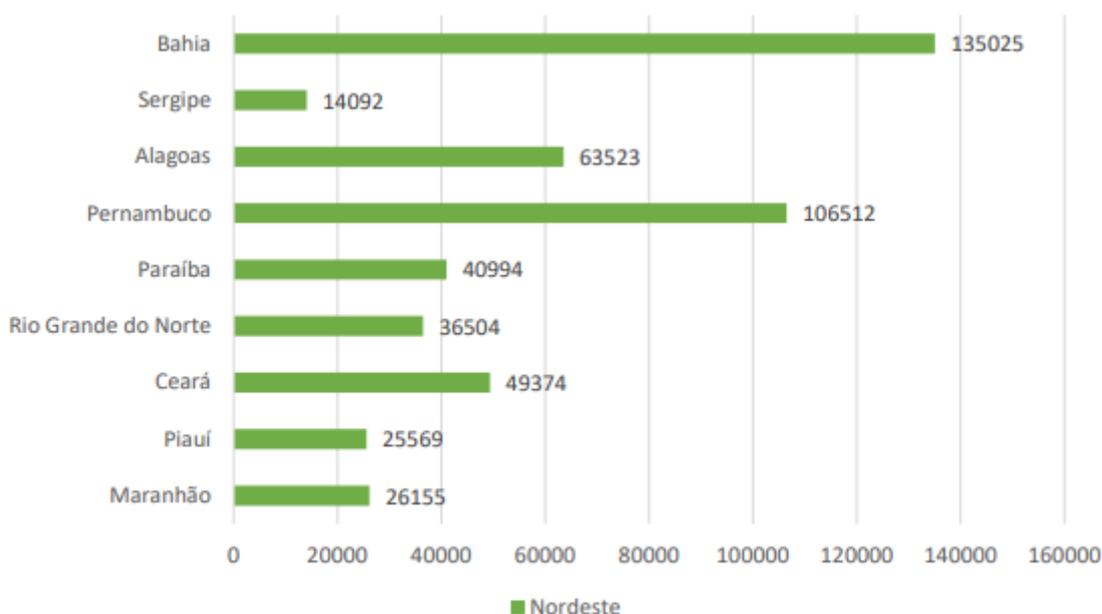


Fonte: (SOUZA et al., 2015)

### **2.2.1.1. Demanda em Pernambuco**

Um estudo feito por Campos et al. (2024) analisou dados referentes a acidentes com animais peçonhentos ocorridos entre 2019 e 2023 na região Nordeste, obtidos através do SINAN. O Anexo C apresenta a ficha de notificação utilizada para coleta de dados nas unidades de saúde. Entre as diversas análises por ele realizadas, focaremos na administração de soroterapia.

Figura 2.4 – Distribuição de notificações por estado do nordeste



Fonte: (CAMPOS et al., 2024)

A Figura 2.4 apresenta os números absolutos de notificações registradas nos estados do nordeste. Os resultados revelaram que o estado da Bahia registrou o maior número de casos, com 135.025 notificações, seguido por Pernambuco, com 106.512, e Alagoas, com 63.523 acidentes. Os três estados, juntos, representam aproximadamente 61% dos AAPs em todo o Nordeste. Essa concentração pode ser atribuída a estudos anteriores que indicam que regiões com maior densidade populacional e atividades ao ar livre, como a agricultura, estão associadas a um maior número de acidentes com peçonhentos (PARISE, 2016; ZEMERO et al., 2022; CAMPOS et al., 2024).

O estudo, também, apresentou a Taxa de Oferta de Soroterapia (TOS) que foi calculada dividindo o número de soroterapias oferecidas pelo total de notificações de acidentes e multiplicando o resultado por 100:

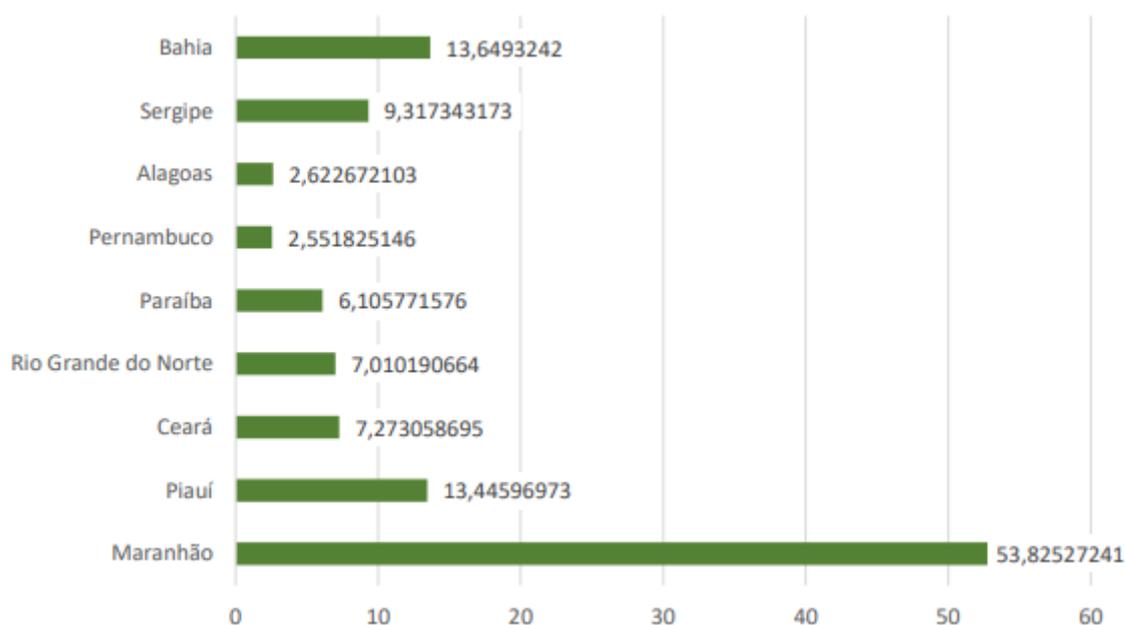
$$TOS = \frac{NS}{TN} \times 100$$

- **NS:** Número de soroterapias
- **TN:** Total de notificações

O TOS sugere uma maior eficiência proporcional no tratamento de acidentes com animais peçonhentos. Os resultados estão na Figura 2.5. Logo, o Maranhão, embora esteja entre os estados com menor número total de acidentes (Figura 2.4), apresentou a maior TOS, com 53,82%.

Ao analisar a situação de Pernambuco, observa-se que o estado possui a menor TOS entre todos os estados, com apenas 2,55%. Embora Pernambuco tenha o segundo maior número de casos no Nordeste, a busca por tratamento com soroterapia é insuficiente. Entre as possíveis causas, a falta de informação sobre a disponibilidade de soros é uma das questões que podem ser consideradas.

*Figura 2.5 – Taxa de oferta de soroterapias por estado do nordeste*



Fonte: (CAMPOS et al., 2024)

### **2.2.1.2 Centro de Informação e Assistência Toxicológica de Pernambuco (CIATox-PE)**

O Centro de Informação e Assistência Toxicológica de Pernambuco (CIATox-PE) foi fundado em 1992, no Hospital da Restauração, e atualmente se localiza no bairro da Boa Vista, em Recife. O perfil do órgão foi ampliado para atuar como uma central de informações, orientando tanto a população quanto profissionais

de saúde sobre intoxicações exógenas e acidentes com animais peçonhentos (PERNAMBUCO, 2024a).

O CIAtox-PE oferece atendimento 24 horas por dia, disponível por meio de telefone, com equipes especializadas compostas por médicos, biólogos, enfermeiros e acadêmicos (ALBUQUERQUE et. al., 2022). O serviço é essencial para descentralizar o tratamento de pacientes intoxicados, que pode ocorrer em diversas unidades de saúde, como as Unidades Básicas de Saúde (UBS), policlínicas, hospitais, etc. Nos casos mais graves que requerem soroterapia, os pacientes são encaminhados para hospitais regionais (PERNAMBUCO, 2024a).

Após o contato telefônico inicial, a equipe do CIAtox-PE cria um prontuário eletrônico para o paciente, no qual é realizada a notificação do caso. Além disso, os dados dos pacientes, coletados via telefone, são registrados no banco de dados do SINITOX (ALBUQUERQUE et. al., 2022). A equipe acompanha remotamente a evolução do paciente até o desfecho final. Além da assistência direta, o centro também se dedica a atividades de ensino e pesquisa, colaborando com universidades e instituições de pesquisa, contribuindo para o avanço do conhecimento sobre intoxicações e AAP (PERNAMBUCO, 2024a).

## 2.3. Tecnologias do Sistema

O sistema desenvolvido constitui um mapeamento de centros de soroterapia em Pernambuco e envolveu uma arquitetura composta por várias ferramentas e bibliotecas tecnológicas. As principais tecnologias envolvidas foram *Python* e suas bibliotecas (tanto de análise de dados como o *framework*<sup>2</sup> *Streamlit* para a construção da interface *web*). A seguir, são descritas as características de cada uma das ferramentas usadas.

### 2.3.1. Google Colab

O *Google Colab* foi utilizado durante o estágio de criação dos *DataFrames*<sup>3</sup> que armazenam os dados das coordenadas dos hospitais e cidades. O *Google*

---

<sup>2</sup> Um *framework* é uma estrutura de software que oferece ferramentas e padrões para facilitar o desenvolvimento de aplicações.

<sup>3</sup> *DataFrames* são estruturas de dados em forma de tabela, usadas para armazenar e manipular dados organizados em linhas e colunas.

*Colab* (abreviação para *Google Colaboratory*) é uma plataforma gratuita usada, principalmente, para análise de dados. Esta ferramenta permite o uso de Unidades de Processamento Gráfico (GPUs) e oferece armazenamento temporário em nuvem para executar *scripts*<sup>4</sup> em *Python*. “É um ambiente digital de acesso aberto e disponibilizado pela *Google* para aplicação de conhecimentos de programação na linguagem *Python*” (SILVA, 2020, apud CARNEIRO et al., 2018).

### 2.3.2. Linguagem de Programação *Python*

Segundo Oliveira (2023), a linguagem de programação *Python*, criada por Guido van Rossum no final da década de 1980, tem experimentado um forte crescimento na popularidade nos últimos anos. É bastante reconhecida por sua sintaxe clara e legível, projetada para facilitar a compreensão e a escrita de código. *Python* é uma linguagem de alto nível, o que significa que oferece um maior nível de abstração em comparação com linguagens de baixo nível, como *C* ou *Assembly*. Essa característica torna *Python* uma escolha atraente tanto para iniciantes em programação quanto para profissionais que desejam prototipar rapidamente novos softwares ou sistemas.

Além disso, o *Python* é uma linguagem de programação interpretada, o que significa que o código é executado linha a linha, em vez de ser compilado de uma vez e isso permite que os programadores testem e corrijam seu código em tempo real (OLIVEIRA, 2023).

Neste trabalho, a linguagem *Python* foi escolhida tanto para a criação do *DataFrame* no *Google Colab* quanto para o desenvolvimento da Interface Gráfica do Usuário (GUI), que permite o uso de filtros de busca e a visualização das informações em um mapa. Para a implementação do código da interface gráfica, foi utilizado o editor *Visual Studio Code* (VSCode), que oferece algumas vantagens, como o recurso *IntelliSense* para autocompletar comandos em *Python*, além da Extensão *Python*, que faz uma integração aprimorada com a linguagem e sua integração nativa com o *Git* (GETTING STARTED WITH PYTHON IN VS CODE, 2024).

As principais bibliotecas e *frameworks* utilizados foram:

---

<sup>4</sup> Arquivos de código escrito.

- **Pandas:** Utilizada para manipulação e análise dos dados, facilitando operações como leitura, filtragem e transformação de diversas extensões de arquivos, como CSV (*Comma-Separated Values*).
- **Geopy:** Uma biblioteca para trabalhar com geocodificação e cálculos de distância entre coordenadas geográficas. Ela facilita a conversão de endereços em coordenadas e o cálculo de distâncias entre pontos no mapa.
- **Folium:** Biblioteca principal para geração de mapas interativos. Ela permite a integração direta com o *Leaflet.js*<sup>5</sup> e facilita a renderização de mapas, adição de marcadores e visualização de trajetos.
- **Streamlit:** *Framework* usado para a construção do *frontend* interativo. Ele oferece uma interface simples e eficiente para a criação de aplicações *web* com visualizações dinâmicas e seleção de dados.
- **Streamlit-Folium:** Um *wrapper*<sup>6</sup> para a integração perfeita entre *Folium* e *Streamlit*, permitindo que os mapas sejam exibidos diretamente na interface do usuário.
- **Streamlit\_js\_eval:** Biblioteca usada para permitir a execução de *JavaScript* diretamente dentro da aplicação *Streamlit*, agregando funcionalidades de *frontend* avançadas.

### 2.3.3. Streamlit

O *Streamlit* é um *framework* de código aberto em *Python* que facilita a criação de aplicações *web* interativas, voltadas principalmente para análise de dados, visualização e exploração de modelos de *machine learning*, além de outros projetos relacionados a dados (FERREIRA et. al., 2024). Sua estrutura de código é de fácil implementação, permitindo o desenvolvimento de aplicações em pouco tempo. Além disso, o *Streamlit* é altamente compatível com diversas bibliotecas, como *Streamlit-Folium* e *Streamlit\_js\_eval*, que expandem suas funcionalidades, facilitando a integração de mapas interativos e a execução de *JavaScript* diretamente na aplicação (FERREIRA et. al., 2024).

No contexto do sistema de mapeamento de soroterapia, o *Streamlit* foi essencial para a construção da GUI, a integração dos dados processados e a visualização em tempo real dos centros de soroterapia.

---

<sup>5</sup> Biblioteca na linguagem *JavaScript* leve e de código aberto para criar mapas em aplicações *web*.

<sup>6</sup> Uma camada de código que encapsula e simplifica a interação com uma biblioteca.

A seguir, uma tabela com os principais comandos do *Streamlit* utilizados no desenvolvimento do sistema:

Tabela 3.1 – Comandos e descrições do *Streamlit*

Comando	Descrição
<code>st.title()</code>	Exibe o título principal da aplicação.
<code>st.image()</code>	Carrega e exibe imagens na aplicação, como logos ou ícones.
<code>st.markdown()</code>	Renderiza <i>Hypertext Markup Language</i> (HTML) e <i>Markdown</i> para formatação personalizada de texto e interfaces.
<code>st.selectbox()</code>	Cria um menu suspenso, permitindo ao usuário selecionar uma opção de uma lista (ex: animal e soro).
<code>st.container()</code>	Cria um <i>container</i> para agrupar elementos de interface e controles.
<code>st.columns()</code>	Organiza o <i>layout</i> da interface em colunas, facilitando a disposição de diferentes elementos na tela.
<code>st.subheader()</code>	Adiciona subtítulos na aplicação, organizando sessões de maneira hierárquica.
<code>st.write()</code>	Exibe texto e variáveis formatadas na aplicação.
<code>st.tabs()</code>	Cria abas interativas para organizar e exibir diferentes conteúdos em uma mesma página.
<code>st.set_page_config()</code>	Define configurações da página como título, ícone e layout.
<code>folium_static()</code> (via <code>st_folium</code> )	Renderiza mapas interativos <i>Folium</i> diretamente no aplicativo <i>Streamlit</i> .

Fonte: Próprio Autor

#### 2.3.4. Controle de Versão com *Git* e *GitHub*

O *Git* é um sistema de controle de versão distribuído mais utilizado atualmente. Constitui uma relação composta pelo o servidor que armazena e gerencia o código e clientes por Interface de Linha de Comando (CLI) e a GUI. “Características como armazenar revisão, criar linhas do tempo alternativas em clientes locais e no servidor e a possibilidade de fundir estas linhas estão entre as principais tarefas do *Git*” (REIS, 2024).

O *GitHub*, por sua vez, é uma plataforma baseada em Git que hospeda repositórios de código e oferece funcionalidades adicionais como automação e colaboração em repositórios de código na nuvem (REIS, 2024).

No desenvolvimento deste projeto, o código foi versionado usando *Git*, com o *GitHub* armazenando o repositório central.

### 2.3.5. Hospedagem e Deploy<sup>7</sup>

A aplicação foi hospedada na nuvem utilizando o *Streamlit Cloud*, que facilita a integração direta com repositórios do *GitHub*. Ao criar um repositório público no *GitHub* contendo o código e as dependências do projeto, os desenvolvedores podem realizar o *deploy* de seus aplicativos de forma simples utilizando o *Streamlit Community Cloud*. O processo envolve autenticar-se com a conta do *GitHub*, selecionar o repositório desejado, definir o *branch*<sup>8</sup> a ser monitorado, e indicar o arquivo principal da aplicação. Após a configuração, o *deploy* é realizado automaticamente. Além disso, qualquer alteração no código-fonte resulta em uma atualização automática do aplicativo, garantindo que a versão mais recente esteja sempre em produção (SOARES, 2024).

## 2.4. Coordenadas Geográficas

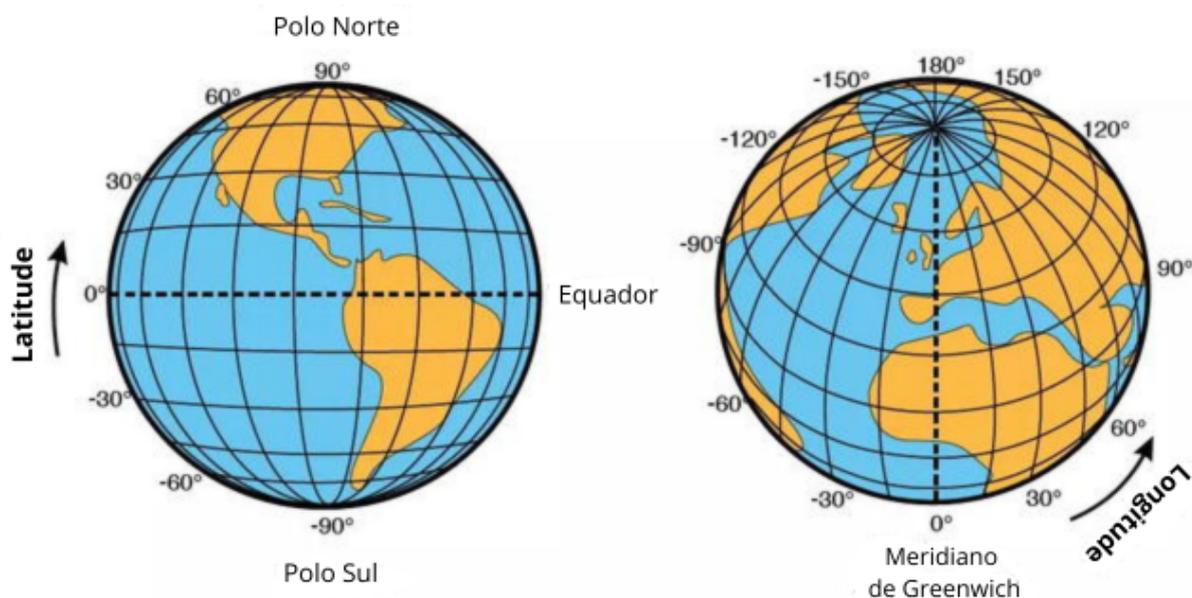
As coordenadas geográficas são um sistema de localização que define qualquer ponto na superfície terrestre a partir de dois valores angulares: latitude e longitude. Esse sistema utiliza como referência eixos imaginários, onde a posição de um ponto é determinada pela interseção de linhas horizontais (latitude) com linhas verticais (longitude), paralelas à linha do Equador e ao Meridiano de *Greenwich*, respectivamente. Devido à forma esférica da Terra, essas coordenadas são expressas em medidas angulares — graus, minutos e segundos — em vez de lineares. A latitude varia entre  $-90^\circ$  a  $90^\circ$  em relação ao equador, enquanto a longitude, em relação ao meridiano de *Greenwich*, varia entre  $-180^\circ$  e  $180^\circ$  (ARAUJO, 2024).

---

<sup>7</sup> *Deploy* é o processo de disponibilizar uma aplicação em um ambiente de produção.

<sup>8</sup> Uma ramificação em um sistema de controle de versão.

Figura 2.6 – Representação da latitude e longitude



Fonte: (VALQUI, 2024) adaptado pelo próprio autor

#### 2.4.1. Geodesic do Geopy: Cálculo da Distância Geodésica

O cálculo da distância geodésica é a medição da distância mais curta entre dois pontos na superfície da Terra, levando em consideração a curvatura do planeta. A função *Geodesic* da biblioteca *Geopy* em *Python* é uma ferramenta que realiza o cálculo de distâncias geodésicas. Essa função utiliza modelos elipsoidais, como o WGS-84 (*World Geodetic System 1984*), que é o padrão utilizado por GPS (*Global Positioning System*). O método de cálculo é baseado no algoritmo de *Vincenty* e o algoritmo de *Karney* (ESMUKOV et. al., 2024).

A função *Geodesic* realiza o cálculo da distância geodésica de forma numérica baseada na Fórmula de *Vicenty*. Para encontrar a distância entre dois pontos, se calcula através do método inverso.

Segundo Kettle (2017), primeiramente, o cálculo da distância geodésica entre dois pontos  $(\phi_1, \lambda_1)$  e  $(\phi_2, \lambda_2)$  ocorre, expressando as coordenadas em radianos, onde  $\phi$  representa a latitude e  $\lambda$  representa a longitude.

Para o modelo WGS-84, estão definidos os valores do semieixo maior  $a$  e o achatamento  $f$  do elipsóide.

De forma resumida, alguns parâmetros precisam ser definidos para serem usados nas expressões. Assim, a diferença de longitude é calculada como:

$$L = \lambda_2 - \lambda_1$$

Os ângulos  $U_1$  e  $U_2$  são definidos como:

$$U_1 = \arctan((1 - f) \cdot \tan(\phi_1))$$

$$U_2 = \arctan((1 - f) \cdot \tan(\phi_2))$$

O  $\sigma$  é o ângulo entre os dois pontos medidos no centro do elipsóide:

$$\sigma = \arctan\left(\frac{\sqrt{((\cos U_2 \cdot \sin \lambda)^2 + (\cos U_1 \cdot \sin U_2 - \sin U_1 \cdot \cos U_2 \cdot \cos L)^2)}}{\sin U_1 \cdot \sin U_2 + \cos U_1 \cdot \cos U_2 \cdot \cos L}\right)$$

O valor de  $\lambda$  é inicialmente definido como  $L$  e é ajustado iterativamente até atingir uma convergência, que seria uma diferença desprezível:  $|\lambda_{novo} - \lambda_{anterior}| < \epsilon$ , onde  $\epsilon$  é essa tolerância (em geral  $1 \times 10^{-12}$ ). Quando não atingida a convergência, ocorre a interatividade, onde é necessário calcular o Ângulo de Azimute ( $\alpha$ ), realizar a correção do achatamento elipsoidal ( $C$ ) e calcular o ângulo médio ( $\sigma_m$ ). Assim, aplicando na expressão abaixo, tem o novo  $\lambda_{novo}$  que pode ser comparado na convergência novamente:

$$\lambda = L + (1 - C) \cdot f \cdot \sin \alpha (\sigma + C \cdot \sin \sigma (\cos^2 \sigma_m + C \cdot \cos \sigma (-1 + 2 \cdot \cos^2 2\sigma_m)))$$

Atingido a convergência, e tendo o  $\sigma$  ideal, a distância geodésica  $s$  é calculada com a fórmula:

$$s = b \cdot A \cdot \sigma$$

Onde  $b = a \cdot (1 - f)$  é o semieixo menor e  $A$  é uma série de termos que dependem de  $U_1$  e  $U_2$ .

A função *Geodesic*, em *Python*, possui vários tipos de entradas. Quando usada assim: *geodesic((latitude1, longitude1), (latitude2, longitude 2))*, o retorno padrão será um objeto de distância que suporta vários métodos para converter sua grandeza, como *meters* (metros) ou *miles* (milhas) (ESMUKOV et. al., 2024).

## 3. Metodologia

### 3.1. Análise de Dados

O objetivo da análise de dados foi criar um *DataFrame* que contivesse a localização dos hospitais de soroterapia em Pernambuco e identificar o hospital mais próximo de cada um dos 185 municípios pernambucanos, por tipo de soro.

#### 3.1.1. Coleta de Dados

##### 3.1.1.1. Dados dos Hospitais com Soroterapia

A primeira etapa focou na obtenção de dados dos hospitais que disponibilizam tratamento por soroterapia em Pernambuco. Esses dados foram extraídos de arquivo em formato PDF (*Portable Document Format*) disponibilizado pelo Ministério da Saúde (ver Anexo 2). A partir dele, foi criado um arquivo CSV chamado `hospitais_23.csv`, contendo as mesmas informações de seu formato em PDF. Após isso, duas novas colunas foram adicionadas ao CSV: latitude e longitude. Como o conjunto de hospitais incluía apenas 14 unidades, as coordenadas geográficas foram obtidas manualmente utilizando a pesquisa no *Google Maps*, cujo um dos retornos é a latitude e longitude do objeto de busca.

A coluna `ATENDIMENTOS_DISPONÍVEIS` do arquivo `hospitais_23.csv` contém os tipos de tratamento oferecidos por cada hospital, incluindo as seguintes categorias: Botrópico, Crotálico, Elapídico, Escorpiônico, Fonêtrico, Loxoscélico e Laquélico. Para atingir o objetivo da análise de dados proposta, foi necessário separar essa coluna em duas novas colunas:

1. **SORO**: Especifica o tipo de soro disponível no hospital.
2. **ANIMAL**: Identifica o tipo de animal peçonhento relacionado ao tratamento oferecido.

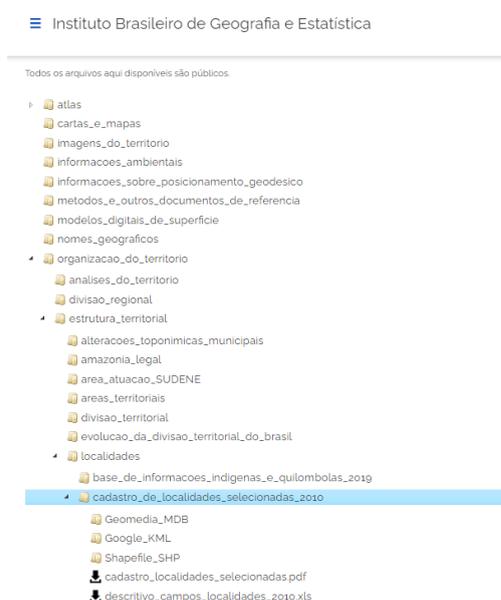
Essa separação é necessária para que o sistema possa filtrar os hospitais com base em dois critérios: tipo de soro disponível e tipo de animal responsável pelo envenenamento.

Para cada combinação de SORO e ANIMAL, novos arquivos CSV foram gerados de forma manual. Assim, foram criadas sete novas tabelas CSV, cada uma contendo os hospitais que oferecem um tipo específico de soro. Por exemplo: Um arquivo CSV com apenas os hospitais que fornecem soro antiescorpiônico; outro arquivo CSV para os hospitais que fornecem soro antibotrópico, e assim por diante.. As sete tabelas geradas podem ser encontradas no Anexo D, e foram fundamentais para a construção do *DataFrame* .

### 3.1.1.2. Dados de Localização dos Municípios Pernambucanos

A segunda etapa, foi a busca da localização dos municípios de Pernambuco. Para identificar as coordenadas destes municípios, foram utilizados dados fornecidos pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Existem diferentes formas de acessar esses dados, incluindo *Application Programming Interface* (APIs) e opções de *download* direto pelo portal do IBGE (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA – IBGE, 2024a; IBGE, 2024b). A fonte de dados utilizada foi uma das opções de *downloads*, com o arquivo disponível no diretório: *cadastro\_de\_localidades\_selecionadas\_2010*, Figura 3.1, que inclui dados geográficos de localidades com base no censo de 2010 (uma vez que a publicação dos dados do censo de 2022 está sendo feita gradualmente (IBGE, 2024a)).

Figura 3.1 – Opções de *downloads* no IBGE



Fonte: Próprio autor

Os dados estão disponíveis em diferentes formatos, como Geomedia MDB (*Microsoft Database*), Google KML (*Keyhole Markup Language*) e Shapefile SHP (*Shapefile Format*), contendo localidades como cidades, vilas e distritos. A opção escolhida foi a *Geomedia MDB* que pode ser reproduzida pelo *Microsoft Access*, ferramenta de banco de dados que se destaca pelos comandos intuitivos para fazer análise de dados (CORREA et. al., 2023). Ao abrir o arquivo, a tabela BR\_localidades\_2010\_v1, dentro do formato MDB na Figura 3.2, foi filtrada pela coluna NM\_CATEGORIA para selecionar apenas as localidades categorizadas como "CIDADE", e pela coluna NM\_UF para isolar as cidades do estado de Pernambuco, filtrando por "PERNAMBUCO".

Figura 3.2 – Tabela BR\_localidades\_2010\_v1 no Microsoft Access

NM_MUNIC	NM_MICRO	NM_MESO	NM_UF	CD_NIVEL	CD_CATEGO	NM_CATEGORIA	NM_LOCALI	LONG	LAT	ALT
ABREU E LIMA	RECIFE	METROPOLITA	PERNAMBUCO	1	05	CIDADE	ABREU E LIMA	-34,89913058	-7,90444899	27,0605
AFOGADOS DA PAJEÚ	SERTÃO PERNA	PERNAMBUCO	PERNAMBUCO	1	05	CIDADE	AFOGADOS DA	-37,63768602	-7,748327586	515,3462
AFRÂNIO	PETROLINA	SÃO FRANCISC	PERNAMBUCO	1	05	CIDADE	AFRÂNIO	-41,00579734	-8,511091284	515,8429
AGRESTINA	BREJO PERNA	AGRESTE PERN	PERNAMBUCO	1	05	CIDADE	AGRESTINA	-35,94677684	-8,455973	437,154
ÁGUA PRETA	MATA MERIDIC	MATA PERNA	PERNAMBUCO	1	05	CIDADE	ÁGUA PRETA	-35,51855735	-8,709122552	92,7346
ÁGUAS BELAS	VALE DO IPOJUL	AGRESTE PERN	PERNAMBUCO	1	05	CIDADE	ÁGUAS BELAS	-37,12195717	-9,1145895	373,3229
ALAGOINHA	VALE DO IPOJUL	AGRESTE PERN	PERNAMBUCO	1	05	CIDADE	ALAGOINHA	-36,77545601	-8,466270523	731,3152
ALIANÇA	MATA SETENT	MATA PERNA	PERNAMBUCO	1	05	CIDADE	ALIANÇA	-35,2285015	-7,604729808	112,1885
ALTINHO	BREJO PERNA	AGRESTE PERN	PERNAMBUCO	1	05	CIDADE	ALTINHO	-36,05969976	-8,4895925	456,2457
AMARAJO	MATA MERIDIC	MATA PERNA	PERNAMBUCO	1	05	CIDADE	AMARAJO	-35,45077292	-8,378487	291,6230
ANGELIM	GARANHUNS	AGRESTE PERN	PERNAMBUCO	1	05	CIDADE	ANGELIM	-36,28481436	-8,890374208	639,9271
ARAÇOIABA	ITAMARACÁ	METROPOLITA	PERNAMBUCO	1	05	CIDADE	ARAÇOIABA	-35,09396653	-7,785641873	161,8505
ARARIPINA	ARARIPINA	SERTÃO PERNA	PERNAMBUCO	1	05	CIDADE	ARARIPINA	-40,50246057	-7,5781415	639,7811
ARCOVERDE	SERTÃO DO M	SERTÃO PERNA	PERNAMBUCO	1	05	CIDADE	ARCOVERDE	-37,05650556	-8,4239405	671,5833
BARRA DE GUA	BREJO PERNA	AGRESTE PERN	PERNAMBUCO	1	05	CIDADE	BARRA DE GUA	-35,66281889	-8,420935038	486,8551
BARREIROS	MATA MERIDIC	MATA PERNA	PERNAMBUCO	1	05	CIDADE	BARREIROS	-35,19908599	-8,814137990	22,0897
BELÉM DE MAR	MATA MERIDIC	MATA PERNA	PERNAMBUCO	1	05	CIDADE	BELÉM DE MAR	-35,83992409	-8,621201911	273,534
BELÉM DO SÃ	ITAPARICA	SÃO FRANCISC	PERNAMBUCO	1	05	CIDADE	BELÉM DO SÃ	-38,96159956	-8,751462780	316,3497
BELO JARDIM	VALE DO IPOJUL	AGRESTE PERN	PERNAMBUCO	1	05	CIDADE	BELO JARDIM	-36,42003508	-8,335295439	616,7049
DETA	SERTÃO DO M	SERTÃO PERNA	PERNAMBUCO	1	05	CIDADE	DETA	-38,2822814	-8,28694670	450,1094

Fonte: Próprio autor

Esses dados foram exportados e armazenados em um arquivo CSV chamado: latitude-longitude-cidades.csv. As colunas contidas no arquivo final foram: id\_municipio: Índice do município, uf: Unidade Federativa (UF), municipio: Nome do município, longitude: Coordenada de longitude e latitude: Coordenada de latitude. A Figura 3.3 mostra essa tabela no *Google Colab*.

Figura 3.3 – Tabela latitude-longitude-cidades no *Google Colab*

```
[ ] cidades = cidades[cidades['uf'] == 'PE']
cidades
```

	id_municipio	uf	municipio	longitudo	latitude
	2801	2767	PE	Abreu e Lima	-34.898389 -7.900719
	2802	2768	PE	Afogados da Ingazeira	-37.630991 -7.743121
	2803	2769	PE	Afrânio	-41.009544 -8.511358
	2804	2770	PE	Agrestina	-35.944725 -8.459657
	2805	2773	PE	Alagoinha	-36.778767 -8.466504
	...	...	...	...	...
	2981	2949	PE	Vicência	-35.313950 -7.656549
	2982	2950	PE	Vitória de Santo Antão	-35.297620 -8.128191
	2983	2951	PE	Xexéu	-35.621222 -8.804601
	2984	2771	PE	Água Preta	-35.526311 -8.706095
	2985	2772	PE	Águas Belas	-37.122619 -9.111255

185 rows x 5 columns

Fonte: Próprio autor

### 3.1.2. Análise e Processamento de Dados

Para a análise e manipulação dos dados, foi utilizado o ambiente *Google Colab*, com a linguagem *Python*. No início, as bibliotecas *Pandas* e *Geopy* foram importadas como mostrado na Figura 3.4.

Figura 3.4 – Importação de bibliotecas

```
[ ] import pandas as pd
from geopy.distance import geodesic
```

Fonte: Próprio autor

No menu “Arquivos” do *Colab*, foram feitos os *uploads* dos arquivos coletados. A Figura 3.5 mostra que os arquivos de formato CSV dos hospitais de soroterapia e da localização dos municípios coletados foram carregados para o ambiente de trabalho usando o método *read\_csv* do *Pandas*. O arquivo *hospitais\_SALox\_23.csv* representa uma das sete tabelas que foram criadas

anteriormente (especificamente a tabela da Figura D.7 do Anexo D). A variável `hospitais_23`, carregou todas as sete tabelas, uma por vez. O programa desenvolvido no *Colab* foi preparado para ser rodado sete vezes, uma para cada tipo de soro (para cada tabela), permitindo a análise individual dos dados de cada tabela e seu relacionamento com os dados de cada cidade.

Figura 3.5 – Carregamento dos arquivos

```
[ ] # Carregar os arquivos CSV
hospitais_23 = pd.read_csv('hospitais_SALox_23.csv', encoding='latin1')
cidades = pd.read_csv('latitude-longitude-cidades.csv')
```

Fonte: Próprio autor

A variável `cidades` ainda foi filtrada para conter apenas as cidades de Pernambuco, como mostra a Figura 3.3.

O cálculo da distância entre cada cidade e cada hospital foi realizado em um *loop*. Conforme ilustrado na Figura 3.6, para cada cidade e hospital, são coletadas as coordenadas geográficas, que estão localizadas nas colunas de latitude e longitude. Essas informações são armazenadas, respectivamente, nas variáveis `cidade_coords` e `hospital_coords`. A biblioteca *Geopy* é utilizada para calcular a distância geodésica por meio do método *geodesic*. O cálculo da distância ocorre em um *loop*, onde, para cada cidade, a distância é calculada em relação a cada hospital.

Figura 3.6 – Cálculo das distâncias e armazenamento de resultados

```
[ ] # Calcular a distância entre cada cidade e cada hospital
for i, cidade in cidades.iterrows():
    cidade_coords = (cidade['latitude'], cidade['longitude'])
    hospital_mais_proximo = None
    distancia_mais_proxima = float('inf')

    for j, hospital in hospitais_23.iterrows():
        hospital_coords = (hospital['LATITUDE'], hospital['LONGITUDE'])
        distancia = geodesic(cidade_coords, hospital_coords).meters

        # Se encontrar uma distância menor, atualizar o hospital mais próximo
        if distancia < distancia_mais_proxima:
            distancia_mais_proxima = distancia
            hospital_mais_proximo = hospital

    # Adicionar os resultados para a cidade atual
    resultados.append([
        cidade['municipio'],
        cidade['latitude'],
        cidade['longitude'],
        hospital_mais_proximo['UNIDADE_DE_SAUDE'],
        hospital_mais_proximo['MUNICIPIO'],
        hospital_mais_proximo['LATITUDE'],
        hospital_mais_proximo['LONGITUDE'],
        hospital_mais_proximo['TELEFONES'],
        hospital_mais_proximo['ENDERECO'],
        round(distancia_mais_proxima/1000, 3),
        hospital_mais_proximo['SORO'],
        hospital_mais_proximo['ANIMAL']
    ])
```

Fonte: Próprio autor

Primeiramente, o código itera sobre as cidades utilizando o método *iterrows()*, que percorre cada linha do *DataFrame* *cidades*. Para cada cidade, a variável *cidade\_coords* armazena a latitude e a longitude. A variável *hospital\_mais\_proximo* é inicializada como *None*, e *distancia\_mais\_proxima* é definida como infinito (*float('inf')*), garantindo que qualquer distância encontrada inicialmente será menor.

Em seguida, o código percorre cada hospital no *DataFrame* *hospitais\_23*, extraíndo suas coordenadas. A distância entre a cidade e o hospital é calculada utilizando *geodesic(cidade\_coords, hospital\_coords).meters*, que retorna a distância em metros.

Dentro do loop que itera sobre os hospitais, se a distância calculada (*distancia*) for menor do que a *distancia\_mais\_proxima* registrada, o hospital atual

torna-se o `hospital_mais_proximo` e a `distancia_mais_proxima` é atualizada. Esse processo assegura que, ao final da iteração sobre os hospitais, teremos o hospital mais próximo da cidade atual.

Após calcular a distância de uma cidade a todos os hospitais, os resultados são armazenados em uma lista, que será convertida em um *DataFrame*, como mostra a Figura 3.7.

Figura 3.7 – Criação do *DataFrame* final

```
# Criar um novo DataFrame com os resultados
colunas = ['cidade_origem', 'latitude_origem', 'longitude_origem', 'hospital', 'cidade_hospital', 'latitude_hospital', 'longitude_hospital', 'telefone', 'endereco', 'distancia_km', 'soro', 'animal']
resultados_df = pd.DataFrame(resultados, columns=colunas)
```

Fonte: Próprio autor

Este *DataFrame* conterá informações sobre a cidade de origem, suas coordenadas, os detalhes do hospital mais próximo e a distância correspondente em quilômetros.

Este programa é executado sete vezes, uma para cada uma das tabelas de hospitais por tipo de soro (Anexo D), garantindo a interação de todas as cidades com todos os hospitais disponíveis e avaliando, dentre eles, qual o mais próximo para cada tipo de soro. O *DataFrame* final foi plotado na Figura 3.8, e já se torna um documento relevante, visto que ele já orienta o hospital mais próximo para cada cidade. Posteriormente, ele foi baixado e intitulado: `hospital_mais_proximo.csv`.

Figura 3.8 – Criação do *DataFrame* final

	cidade_origem	latitude_origem	longitude_origem	hospital	cidade_hospital	latitude_hospital	longitude_hospital	telefone	endereco	distancia_km	soro	animal
0	Abreu e Lima	-7.900719	-34.898389	Hospital da Restauração	Recife	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	Avenida Gov. Agamenon Magalhães s/n - Derby	16.881	Soro Antibiótico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
1	Afogados da Ingazeira	-7.743121	-37.630991	Hospital Regional Professor Agamenon Magalhães	Serra Talhada	-7.988751	-38.298864	(87) 3831-9600	Rua Comandante Superior 935 - Nossa Senhora da...	78.502	Soro Antibiótico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
2	Afrânio	-8.511358	-41.009544	Hospital Universitário de Petrolina (HU-UNIVASF)	Petrolina	-9.392355	-40.495607	(87) 2181-6501/6526	Avenida José de Sá Maniçoba s/n - Centro	112.644	Soro Antibiótico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
3	Agestina	-8.459657	-35.944725	Hospital Mestre Vitalino	Caruaru	-8.245172	-35.973795	(81) 3725-7750	Avenida Amazonas 175 - Bairro Universitário	23.937	Soro Antibiótico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
4	Alagoinha	-8.466504	-36.778767	Hospital e Policlínica Rui de Barros Correia	Arcoverde	-8.422456	-37.054930	(87) 3821-8296	Avenida Agamenon Magalhães s/n - São Miguel	30.799	Soro Antibiótico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
...	...	...	...	...	...	...	...	...	...	...	...	...
1290	Vicência	-7.656549	-35.313950	Hospital da Restauração	Recife	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	Avenida Gov. Agamenon Magalhães s/n - Derby	63.472	Soro Antiloxosélico (SALox) - aranha marrom	Aranha
1291	Vitória de Santo Antão	-8.128191	-35.297620	Hospital da Restauração	Recife	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	Avenida Gov. Agamenon Magalhães s/n - Derby	44.801	Soro Antiloxosélico (SALox) - aranha marrom	Aranha
1292	Xexéu	-8.804601	-35.621222	Hospital Mestre Vitalino	Caruaru	-8.245172	-35.973795	(81) 3725-7750	Avenida Amazonas 175 - Bairro Universitário	73.041	Soro Antiloxosélico (SALox) - aranha marrom	Aranha
1293	Água Preta	-8.706095	-35.526311	Hospital Mestre Vitalino	Caruaru	-8.245172	-35.973795	(81) 3725-7750	Avenida Amazonas 175 - Bairro Universitário	70.898	Soro Antiloxosélico (SALox) - aranha marrom	Aranha
1294	Águas Belas	-9.111255	-37.122619	Hospital e Policlínica Rui de Barros Correia	Arcoverde	-8.422456	-37.054930	(87) 3821-8296	Avenida Agamenon Magalhães s/n - São Miguel	76.544	Soro Antiloxosélico (SALox) - aranha marrom	Aranha

Fonte: Próprio autor

A Figura 3.9 mostra informações sobre o DataFrame, utilizando o método `info()`. As colunas `cidade_origem`, `latitude_origem` e `longitude_origem` são referentes a cada cidade de Pernambuco. A coluna `distancia_km` informa a distância entre o item da coluna `hospital` e a `cidade_origem`. As demais colunas são informações referente a cada hospital.

Figura 3.9 – Informações do DataFrame final

```
[ ] resultados_df.info()
↳ <class 'pandas.core.frame.DataFrame'>
  RangeIndex: 1295 entries, 0 to 1294
  Data columns (total 12 columns):
   #   Column              Non-Null Count  Dtype
  ---  ---
   0   cidade_origem        1295 non-null   object
   1   latitude_origem      1295 non-null   float64
   2   longitude_origem     1295 non-null   float64
   3   hospital             1295 non-null   object
   4   cidade_hospital      1295 non-null   object
   5   latitude_hospital    1295 non-null   float64
   6   longitude_hospital   1295 non-null   float64
   7   telefone             1295 non-null   object
   8   endereco             1295 non-null   object
   9   distancia_km         1295 non-null   float64
  10   soro                 1295 non-null   object
  11   animal               1295 non-null   object
dtypes: float64(5), object(7)
memory usage: 121.5+ KB
```

Fonte: Próprio autor

## 3.2. Sistema de Mapeamento

Esta seção descreve a implementação técnica do sistema de mapeamento de centros de soroterapia em Pernambuco. A aplicação foi desenvolvida em *Python* utilizando o *framework Streamlit*, com a IDE *VSCode* para escrita do código-fonte. Foram utilizados dois arquivos CSV, coletados, como fonte de dados: `hospital_mais_proximo.csv` e `latitude-longitude-cidades.csv`. O primeiro contém informações sobre os hospitais mais próximos, com soroterapia disponível, para cada cidade, por tipo de soro. O segundo, com as coordenadas geográficas (latitude e longitude) das cidades.

O objetivo desta aplicação é desenvolver uma interface gráfica (GUI) que gere um mapa interativo mostrando os hospitais em Pernambuco que oferecem soroterapia. A interface permite que o usuário selecione o animal envolvido no envenenamento, o tipo de soro necessário e a cidade de origem, para então identificar o hospital mais próximo. Além disso, a aplicação exibe informações relevantes sobre o hospital de destino, como endereço, distância e telefone. A interface é projetada para ser intuitiva e proporcionar respostas rápidas.

### 3.2.1. Visão do Código-Fonte

O código-fonte da aplicação, disponível no Anexo E, foi estruturado utilizando diversas bibliotecas de Python e construído em arquitetura monolítica devido sua simplicidade. Dentre elas, a biblioteca *Pandas* desempenha é responsável pelo processamento e manipulação dos dados. Os arquivos CSV (*hospital\_mais\_proximo.csv* e *latitude-longitude-cidades.csv*) são lidos e transformados em *DataFrames* através da função *pd.read\_csv()*. Trabalhar com *DataFrame* possibilita a fácil filtragem e seleção das informações relevantes, como o município de origem do paciente, o tipo de soro necessário e o hospital mais próximo. A biblioteca permite que o usuário selecione, por meio de filtros, o animal responsável pelo acidente e o soro correspondente. O código para a seleção do animal e do soro é implementado assim:

Figura 3.10 – Caixas de seleções de animal e soro

```
animal = st.selectbox("Qual tipo de animal causou o acidente?", dados_geral['animal'].unique(), index=None, placeholder="Selecione o animal")
soro = st.selectbox('Soro Antiveneno', dados_geral[dados_geral['animal']==animal]['soro'].unique(), index=None, placeholder="Selecione o Soro Antiveneno")
```

Fonte: Próprio autor

Os comandos do *Streamlit* ajudam na construção da interface da aplicação, fornecendo opções para interagir com o usuário. No caso acima, a função *st.selectbox()* cria opções de caixas de seleções com os tipos de animal e soro disponíveis no *DataFrame*. O mesmo ocorre para as cidades pernambucanas, posteriormente.

Toda interface da aplicação é construída com *Streamlit*, dividindo a tela utilizando as funções de layout de colunas e abas do *Streamlit* para organizar as informações, deixando o visual claro e conciso (Tabela 3.1). A tela inicial é dividida

em três colunas, onde a coluna central contém o título e o conteúdo principal, enquanto as laterais são usadas para imagens e espaçamentos.

Dentro da aba "Buscador de Soros", se encontram os filtros criados, há uma lógica de filtragem onde o usuário pode selecionar o tipo de animal, soro e a cidade de origem nas opções das caixas de seleções disponíveis. Esses filtros são aplicados diretamente ao *DataFrame* que contém as informações dos hospitais, resultando na seleção do hospital mais próximo com o soro necessário. O código para essa lógica de filtragem é o seguinte:

Figura 3.10 – Filtro no *DataFrame*

```
#Filtro destino
filtro = (dados_geral['soro'] == soro)&(dados_geral['cidade_origem'] == mun_origem)
municipio_origem = dados_geral[filtro]
```

Fonte: Próprio autor

Em seguida, há uma integração com a biblioteca *Folium* que permite a exibição dos resultados no formato de mapas interativos. Após a aplicação dos filtros, a localização do município de origem e do hospital mais próximo é plotada no mapa, com marcadores diferenciados para indicar o ponto de origem (paciente) e o ponto de destino (hospital com o soro disponível):

Figura 3.11 – Plotagem do mapa com *Folium*

```
folium.Marker(
    location= [municipio_origem['latitude_origem'].values, municipio_origem['longitude_origem'].values],
    tooltip="Origem",
    popup="Você está aqui",
    icon=folium.Icon(color="green",icon="house",prefix="fa"),
).add_to(mapa)

folium.Marker(
    location= [municipio_destino['latitude_hospital'].values, municipio_destino['longitude_hospital'].values],
    tooltip="Destino",
    popup=f"O soro está aqui, na cidade de {municipio_destino['cidade_hospital'].values[0]}, {municipio_destino['hospital'].values[0]}",
    icon=folium.Icon(color="red", icon="hospital",prefix="fa"),
).add_to(mapa)
```

Fonte: Próprio autor

O mapa é centralizado entre as duas localizações e utiliza *tooltips* e *popups* para fornecer informações ao usuário. Ao lado é apresentado todas as informações sobre o hospital de destino disponíveis do *DataFrame*. Além disso, um mapa exibido com todos os hospitais cadastrados no sistema é apresentado caso não haja seleção alguma nos filtros, além de apresentar dados do CIATox-PE.

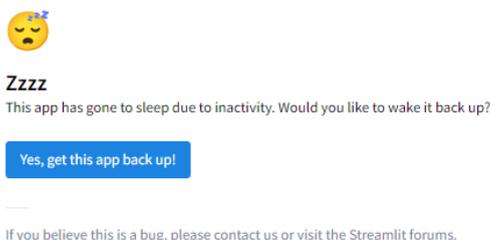
A explicação detalhada do código-fonte descrito nessa seção e apresentado no Anexo E pode ser encontrada na Documentação do Código-Fonte do Sistema de Mapeamento de Hospitais por Soroterapia em Pernambuco, versão 1.0, criado pelo autor da monografia e revisado pela respectiva orientadora da monografia.

### 3.2.2. *Deploy* do Sistema

Após o desenvolvimento da aplicação, o sistema foi versionado utilizando o *Git* e enviado para um repositório remoto no *GitHub*. A partir desse repositório, foi possível conectar o projeto diretamente ao *Streamlit Cloud* para realizar o *deploy*. Esse processo envolveu a criação de uma conta no *Streamlit Cloud* e a integração com o *GitHub*, permitindo que o código-fonte fosse hospedado a partir do endereço: <https://sorosantivenompe.streamlit.app/>. O *deploy* foi realizado de maneira simples, onde o *Streamlit Cloud* lê o repositório e automaticamente configura o ambiente necessário para executar a aplicação, utilizando o arquivo *requirements.txt* presente no projeto para instalar as dependências.

Entretanto, uma característica do *Streamlit Cloud*, para conservar recursos do servidor, a aplicação entra em estado de hibernação quando não está em uso por um determinado período. Todos os dias, ao acessar o endereço, o *Streamlit* exibe uma mensagem solicitando que o usuário clique em "*Yes, get this app back up!*" para reativar a aplicação (Figura 3.11). Esse processo de reativação garante que a aplicação seja carregada novamente. Este ciclo garante que qualquer nova atualização no código-fonte seja automaticamente refletida na aplicação hospedada.

Figura 3.11 – Mensagem de hibernação do *Streamlit Cloud*



Fonte: Próprio autor

### 3.3. Avaliação do Sistema

Como forma de garantir a eficiência do sistema, a aplicação foi avaliada sob dois principais métodos: casos de testes e comparações com sistemas já implementados.

#### 3.3.1. Casos de Testes

Segundo Leloudas (2023), o teste de *software* é uma etapa importante no desenvolvimento de sistemas, focado em identificar defeitos, erros e verificar se o sistema atende aos requisitos e funciona corretamente. Ele ainda diz que o teste assegura que o *software* seja confiável, eficiente e atenda às expectativas dos usuários. O processo de teste é contínuo, começando nas fases iniciais do desenvolvimento e se estendendo por todo o ciclo de vida do *software*. Existem diferentes níveis de testes, incluindo teste de unidade, integração, sistema e aceitação, cada um com um propósito específico em diferentes estágios do desenvolvimento.

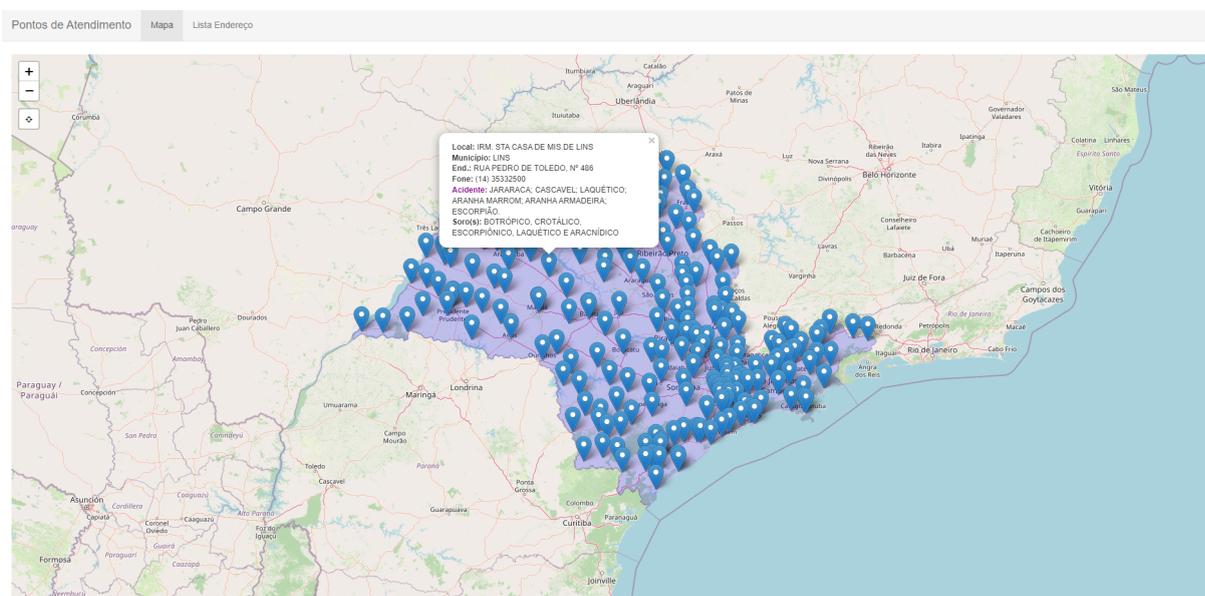
Assim, foi utilizado o teste de Interface Gráfica do Usuário (GUI) para testar o sistema construído, pois as funcionalidades da aplicação devem funcionar de acordo com o projetado. Para isso, foi construído um modelo de teste, ver Anexo F (Tabela F.1), que possui casos de testes, objetivos, procedimentos, resultados esperados e o resultado obtido. A aplicação foi testada e o resultado obtido foi preenchido por um membro do grupo de extensão.

#### 3.3.2. Sistemas Análogos

Sistemas com funcionalidades e tecnologias semelhantes já foram implementados e disponibilizados por secretarias de saúde para o uso da população. Um exemplo é o mapa online lançado pela Secretaria da Saúde do Estado de São Paulo, que permite a localização e identificação dos 220 pontos de atendimento soroterápico para vítimas de envenenamento por animais peçonhentos (FRAGA, 2024).

O mapa oferece uma visão completa de todos os pontos de atendimento, com a opção de geolocalização que mostra a posição do usuário e, conseqüentemente, a localização dos hospitais mais próximos. Assim, é de responsabilidade do usuário verificar qual hospital possui o soro específico necessário para o tratamento, já que a ferramenta não realiza essa filtragem automaticamente. A Figura 3.12 apresenta a página inicial da ferramenta (FRAGA, 2024).

**Figura 3.12 – Mapa online de tratamento por soroterapia de São Paulo**



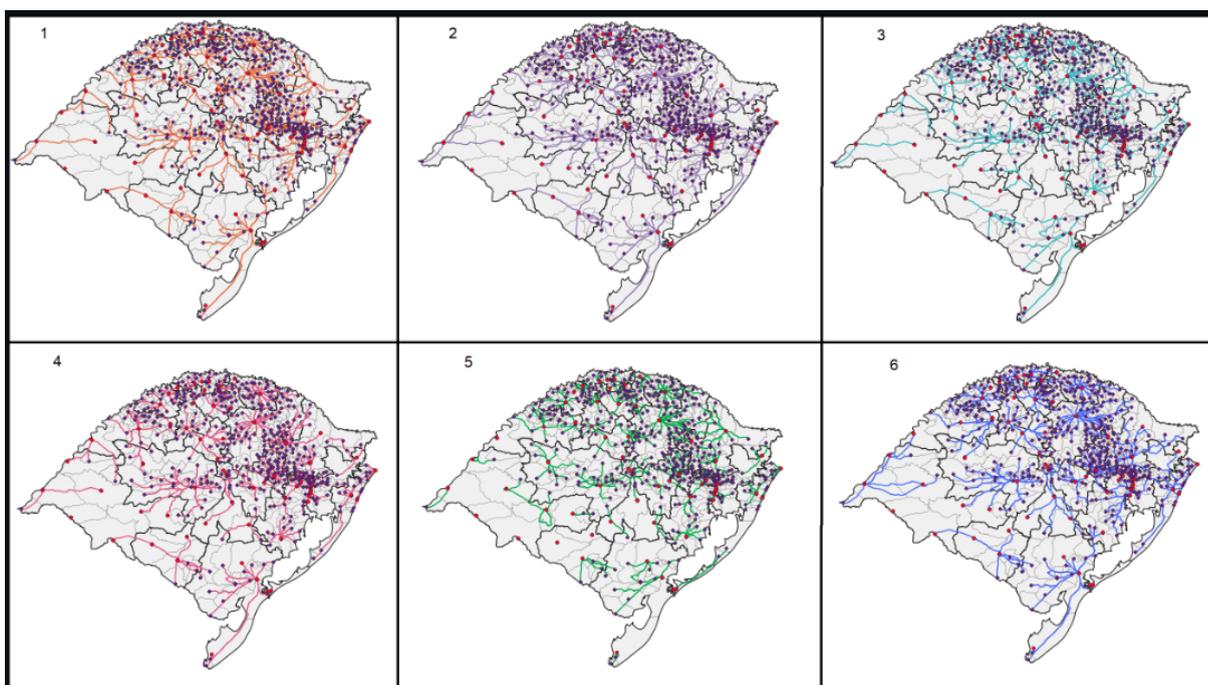
Fonte: (FRAGA, 2024)

Outro exemplo de sistema é o Buscador de Soros Antiveneno no Rio Grande do Sul, desenvolvido pela Secretaria Estadual de Saúde do estado. Assim como o sistema deste projeto, ele também está hospedado no *Streamlit Cloud* e tem a mesma função de indicar o hospital mais próximo com o soro adequado para o tipo de envenenamento, considerando a cidade de origem do usuário. Mas, sua metodologia foi desenvolvida no *software QGIS*, utilizando um processo chamado *Network Analysis*, com o objetivo de calcular as distâncias dos municípios ao ponto focal de soro antiveneno mais próximo. Para a idealização desse projeto, foram utilizados dados *shapefile* (linhas, pontos e polígonos), além de ferramentas e *plugins* específicos do *QGIS*. Cada hospital foi definido como ponto focal, mas, como nem todos os hospitais mantêm estoque dos seis tipos de soros antivenenos, o

geoprocessamento foi realizado seis vezes, uma para cada soro (JARENKOW, 2023).

Além disso, o sistema levou em consideração as rodovias do estado para traçar as distâncias de forma mais assertiva (JARENKOW, 2023). A Figura 3.13 ilustra os seis geoprocessamentos realizados, cada um representando a distribuição de um soro antiveneno específico.

*Figura 3.13 – Geoprocessamentos realizados pelo QGIS no Rio Grande do Sul*



Fonte: (JARENKOW, 2023)

## 4. Resultados

O sistema desenvolvido demonstra um bom desempenho em sua principal função: localizar os hospitais mais próximos que oferecem soroterapia.

Em comparação com o mapa online da Secretaria da Saúde do Estado de São Paulo, o sistema oferece uma vantagem significativa, pois identifica diretamente o hospital a ser procurado, além de fornecer suas informações detalhadas. Já a aplicação de São Paulo pede que o próprio usuário localize o hospital, fornecendo apenas os meios para fazê-lo.

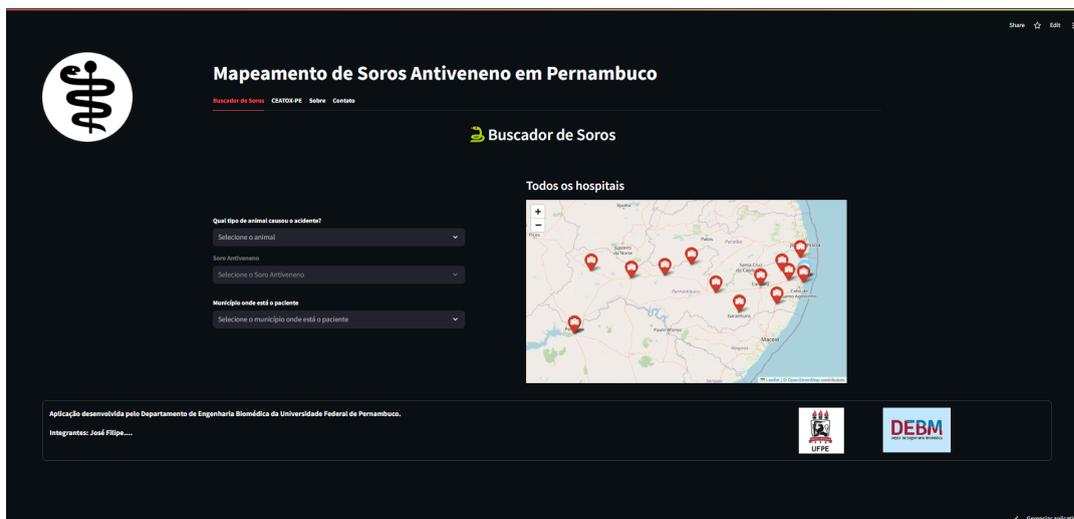
Com relação ao Buscador de Soros Antiveneno do Rio Grande do Sul, embora o presente sistema seja eficiente, o sistema gaúcho apresenta uma maior precisão no cálculo das distâncias, pois considera a malha rodoviária em seus cálculos. Porém, o presente sistema calcula a distância entre dois pontos sem levar em conta as rodovias.

A fim de facilitar futuras implementações, foi criada a primeira versão da Documentação do Código-Fonte da aplicação. Essa documentação inclui a configuração do ambiente de desenvolvimento, uma visão geral do código-fonte, casos de uso e testes realizados. O objetivo é proporcionar aos futuros desenvolvedores uma base detalhada sobre o funcionamento desta versão inicial do sistema, garantindo uma continuidade bem estruturada para aprimoramentos posteriores. Este documento serviu de base para sintetizar as informações da Seção 3.2.1 deste trabalho.

### 4.1. Interface do Usuário

A interface é composta por apenas uma tela (tela inicial) com três colunas, conforme mostrado na Figura 4.1. Na primeira coluna, é exibida a logomarca do projeto. A segunda coluna contém o título e as abas de navegação, enquanto a terceira, reservada inicialmente para uso futuro, será utilizada para outros conteúdos como informações adicionais ou permanecerá sendo apenas um espaçamento.

Figura 4.1 – Tela inicial da aplicação em um monitor



Fonte: Próprio autor

A tela contém quatro abas principais: Buscador de Soros, CIATOX-PE, Sobre, e Contato. A aba principal, Buscador de Soros, permite ao usuário selecionar o tipo de animal causador do acidente, o tipo de soro necessário, e a cidade onde o paciente está localizado. Essas opções acionam o filtro que busca o hospital mais próximo com o antiveneno requerido. As demais abas estão reservadas para implementações futuras.

O formulário de busca, na aba Buscador de Soros, oferece três campos principais: a seleção do animal, o tipo de soro, e o município de origem do paciente. Esses campos são apresentados ao usuário por meio de caixas de seleção (*selectboxes*), conforme demonstrado na Figura 4.2. Essa estrutura permite que o usuário filtre rapidamente os resultados com base nas opções selecionadas.

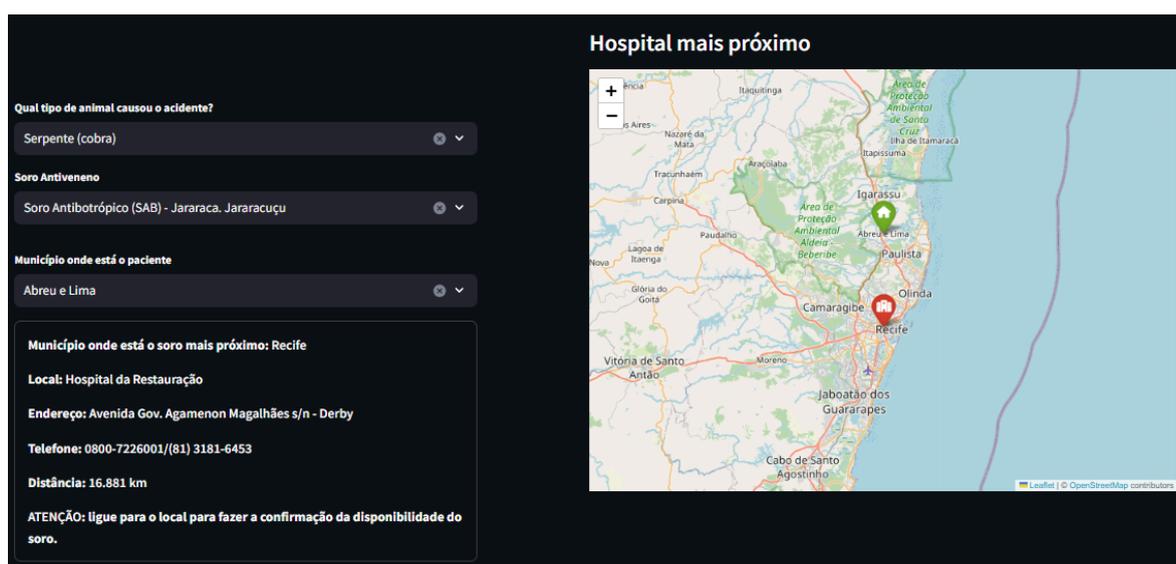
Figura 4.2 – Filtros de busca

Fonte: Próprio autor

O mapa interativo, gerado com a biblioteca *Folium*, exibe a localização dos hospitais mais próximos que possuem o soro necessário. O mapa, exibido à direita do formulário de busca, inclui marcadores de origem e destino, com detalhes sobre o hospital, como nome, endereço e telefone. O mapa pode ser ampliado e explorado pelos usuários, facilitando a visualização das informações geográficas. A tela inicial apresenta, por padrão, todos os hospitais de referências por soroterapia registrados no banco de dados, além de destacar o CIATox-PE com marcador azul (Figura 4.1).

A Figura 4.3 mostra um exemplo de resultado para os filtros selecionados: Serpente(cobra), Soro Antibotrópico e Abreu e Lima. O mapa é redimensionado para mostrar a região onde se encontram as cidades de origem e destino, além de um quadro, abaixo dos filtros, mostrar as informações relevantes para o hospital de destino.

Figura 4.3 – Exemplo de resultado

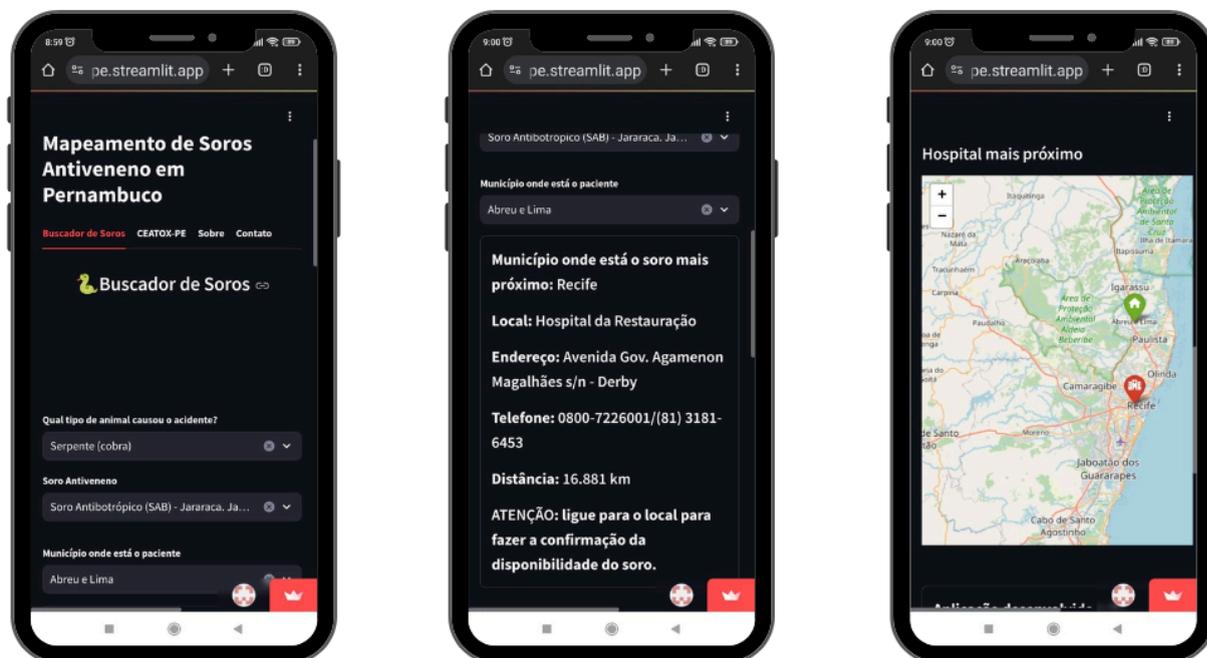


Fonte: Próprio autor

A interface foi projetada para ser responsiva, adaptando-se a diferentes tamanhos de tela, desde computadores até dispositivos móveis. A Figura 4.4 apresenta a exibição da aplicação em um dispositivo móvel e a Figura 4.4 mostra a exibição da aplicação em um *smartwatch*<sup>9</sup>. Essa responsividade é graças as ferramentas do *streamlit* que foi ideal para esta aplicação pois seu uso previsto é exatamente por dispositivos móveis.

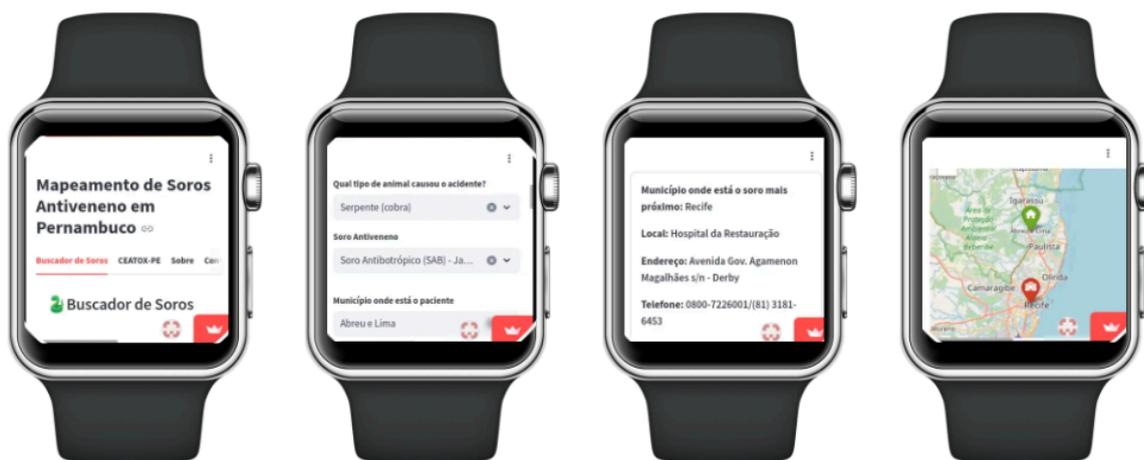
<sup>9</sup> Um relógio inteligente com recursos de conectividade e aplicativos.

Figura 4.4 – Representação da aplicação no celular



Fonte: Próprio autor

Figura 4.5 – Representação da aplicação no smartwatch



Fonte: Próprio autor

## 4.2. Resultado dos Testes

Todos os testes descritos no Tabela F.1 foram aprovados com sucesso. No entanto, algumas sugestões foram feitas para melhorar a experiência do usuário,

como a alteração do formato da distância exibida, reduzindo de três para duas casas decimais para facilitar a compreensão. O tempo de resposta ao carregar o mapa em dispositivos *desktop* foi de apenas 1 segundo, enquanto em dispositivos móveis foi de 2 segundos, ambos dentro de limites aceitáveis. Os filtros e a exibição de dados funcionaram perfeitamente, sem apresentar problemas durante os testes.

## 5. Conclusão

Este trabalho mostrou a viabilidade de um sistema que visa otimizar a resposta a acidentes com animais peçonhentos em Pernambuco, um estado onde estes incidentes são frequentes e pouco tratados. A aplicação desenvolvida cumpre seu papel ao permitir que usuários localizem rapidamente o hospital mais próximo com soroterapia disponível, levando em consideração o tipo de soro necessário e a cidade de origem da vítima. O sistema contribui para reduzir os riscos de complicações graves, especialmente em regiões afastadas dos grandes centros urbanos, onde o acesso aos serviços de saúde é mais limitado.

A implementação do sistema em *Python* e sua hospedagem na plataforma *Streamlit Cloud* mostra que as ferramentas tecnológicas modernas podem ser aplicadas de forma eficiente na saúde pública e de simples implementação.

Há diversas melhorias a serem implementadas no sistema, e é importante destacar algumas delas para o futuro. Primeiramente, uma refatoração do código-fonte é necessária para torná-lo mais legível, o que inclui a modificação dos nomes de variáveis e a remoção de linhas de código obsoletas. Uma adição da função de geolocalização, como a realizada pela aplicação de São Paulo, e a mudança do cálculo da distância para considerar os dados de malha rodoviária de Pernambuco, como é o caso do sistema gaúcho, deixaria o sistema mais funcional. Além disso, o processo de coleta de dados dos hospitais com soroterapia precisa ser automatizado e linear. Atualmente, a coleta é feita manualmente, o que pode se tornar inviável caso haja coleta em estados mais extensos e populosos. Para facilitar a expansão do sistema para outros estados e regiões, é essencial automatizar e padronizar esse processo. Outro ponto, é o fluxo de criação do *dataframe* final, que precisa ser melhor idealizado para minimizar os possíveis erros que o fluxo atual pode causar.

O sistema deve ser pensado para expansão, não apenas para ser suporte a assistência em outros estados, mas também para melhorar seu funcionamento em Pernambuco. Muitas cidades do estado estão mais próximas de centros de tratamento por soroterapia localizados em estados vizinhos. Por exemplo, algumas

idades da região do Sertão do São Francisco, como Jatobá, Tacaratu e Petrolândia, têm acesso mais rápido aos centros de soroterapia da cidade baiana de Paulo Afonso do que aos disponíveis em Pernambuco. Considerando que o Sistema Único de Saúde (SUS) é estruturado de forma integrada, já existem macrorregiões interestaduais de saúde, como a PEBA, que abrange cidades do sertão de Pernambuco e da Bahia (PERNAMBUCO, 2024b). Essa cooperação entre estados precisa ser intensificada no sistema, garantindo um atendimento mais eficiente nessas áreas.

A abordagem interdisciplinar, que combina engenharia biomédica e tecnologia da informação, mostra-se um exemplo de como soluções tecnológicas podem ser implementadas para melhorar a eficiência dos serviços de saúde. A expectativa é que a iniciativa inspire mais desenvolvimentos e colaborações no campo da saúde pública e do atendimento emergencial.

## Referências

ALBUQUERQUE, M. C. A. de; LYRA FILHO, C. R. do N.; AMORIM, M. L. P.; LINS, I. B. L.; LIMA, P. V. C. de; MELLO, M. J. G. de. ***Venomous animals in Pernambuco: children at risk***. Revista Brasileira de Saúde Materno Infantil, v. 22, n. 1, p. 29-36, jan.-mar. 2022. Disponível em: <https://doi.org/10.1590/1806-93042022000100003>. Acesso em: 27 set. 2024.

ARAUJO, N. F. **Plano cartesiano e coordenadas geográficas no ensino médio: uma proposta de sequência didática articulando matemática e geografia**. 2024. TCC – Pós-Graduação em Matemática, Instituto Federal da Paraíba, Cajazeiras, 2024.

BRASIL. **Constituição (1988)**. Constituição da República Federativa do Brasil. Brasília, DF: Senado Federal, 1988. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/constituicao/constituicao.htm](https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm). Acesso em: 15 set. 2024.

BRASIL. **Decreto nº 7.508, de 28 de junho de 2011**. Diário Oficial da União, Brasília, DF, 29 jun. 2011. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/\\_Ato2011-2014/2011/Decreto/D7508.htm](https://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2011/Decreto/D7508.htm). Acesso em: 15 set. 2024.

BRASIL. Ministério da Saúde. **Acidentes por animais peçonhentos**. 2016. Disponível em: <https://portalsinan.saude.gov.br/acidente-por-animais-peconhentos>. Acesso em: 20 set. 2024.

BRASIL. Ministério da Saúde. **Hospitais de referência para acidentes por animais peçonhentos**. 2023. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/a/animais-peconhentos/hospitais-de-referencia>. Acesso em: 20 set. 2024.

BRASIL. Ministério da Saúde. **Guia de animais peçonhentos do Brasil**. 1. ed. Brasília: Ministério da Saúde, 2024. Disponível em: <https://portaldeboaspraticas.iff.fiocruz.br/wp-content/uploads/2024/05/Guia-Animais-peconhentos-do-Brasil.pdf>. Acesso em: 20 set. 2024.

BRASIL. Ministério da Saúde. Fundação Nacional de Saúde. **Manual de diagnóstico e tratamento de acidentes por animais peçonhentos**. 2. ed. Brasília: Fundação Nacional de Saúde, 2001. Disponível em: <https://www.icict.fiocruz.br/sites/www.icict.fiocruz.br/files/Manual-de-Diagnostico-e-Tratamento-de-Acidentes-por-Animais-Pe--onhentos.pdf>. Acesso em: 10 set. 2024.

CAMPOS, C. L. M.; NERY, W. da S.; NASCIMENTO, N. G. S. do; et al. **Análise epidemiológica dos acidentes provocados por animais peçonhentos no**

**nordeste do Brasil entre 2019 e 2023.** Revista Brasileira de Medicina de Excelência, v. 2, n. 3, p. 1-15, jul./ago. 2024. Disponível em: <<https://sevenpublicacoes.com.br/REVMEDBRA/article/view/5371/9840>>. Acesso em: 15 set. 2024.

CARNEIRO, T.; NÓBREGA, R. V. M. da; NEPOMUCENO, T.; BIAN, G.; ALBUQUERQUE, V. H. C. de; FILHO, P. P. R. **Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications.** IEEE Access, v. 6, p. 61677–61685, 2018.

CORREA, G. C.; AVILA, N. A.; SILVA, J. G. de O.; CARR, C. N.; CARR, R. F. **Microsoft Access: tudo o que você precisa saber para gerenciar seu banco de dados de maneira eficaz.** Research, Society and Development, v. 12, n. 5, 2023. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/41324>>. Acesso em: 27 set. 2024.

CUNHA, F. M. S. **Um balanço geral dos acidentes por animais peçonhentos no nordeste brasileiro.** Não publicado, 2024.

ESMUKOV, K.; et. al. **geopy/geopy.** GITHUB. 2024. Disponível em: <<https://github.com/geopy/geopy>>. Acesso em: 2 out. 2024.

FERREIRA, L. O. de P.; LARA, H. S. **Desenvolvimento e aplicação do TURDUS: uma ferramenta interativa para análise de sinais de aceleração com Python.** OBSERVATÓRIO DE LA ECONOMÍA LATINOAMERICANA, [S. l.], v. 22, n. 9, p. 6779 , 2024

FRAGA, F. **SP lança mapa online com pontos de atendimento com soro antiveneno.** Agência Brasil. 2024. Disponível em: <<https://agenciabrasil.ebc.com.br/saude/noticia/2024-01/sp-lanca-mapa-online-com-pontos-de-atendimento-com-soro-antiveneno>>. Acesso em: 10 out. 2024.

GETTING STARTED WITH PYTHON IN VS CODE. **Visual Studio Code.** 2024. Disponível em: <<https://code.visualstudio.com/docs/python/python-tutorial>>. Acesso em: 21 set. 2024.

IMUNOLOGIA: visão geral. **Realize Educação.** 2023. Disponível em: <<https://realizeeducacao.com.br/wiki/imunologia-visao-geral/>>. Acesso em: 21 set. 2024.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA – IBGE. **Downloads Geociências.** Disponível em: <<https://www.ibge.gov.br/geociencias/downloads-geociencias.html>>. Acesso em: 27 set. 2024.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA – IBGE. **API de Serviço de Dados.** Disponível em: <<https://servicodados.ibge.gov.br/api/docs/>>. Acesso em: 27 set. 2024.

MEDEIROS, C. R.; MALAQUE, C. M. S.; WEN, F. H.; FRANÇA, F. O. S.; et. al. **Acidentes por animais peçonhentos.** Instituto Butantan. 2002. Disponível em:

<[https://www.saude.sp.gov.br/resources/cve-centro-de-vigilancia-epidemiologica/areas-de-vigilancia/doencas-de-transmissao-por-vetores-e-zoonoses/aula03\\_peconhentos.pdf](https://www.saude.sp.gov.br/resources/cve-centro-de-vigilancia-epidemiologica/areas-de-vigilancia/doencas-de-transmissao-por-vetores-e-zoonoses/aula03_peconhentos.pdf)>. Acesso em: 17 out. 2024.

JARENKOW, A. **andrejarenkow/Portfolio**. GITHUB. 2023. Disponível em: <<https://github.com/andrejarenkow/Portfolio>>. Acesso em: 2 out. 2024.

KETTLE, S. **Distance on an ellipsoid: Vincenty's Formulae**. *Environmental Systems Research Institute*. 2017. Disponível em: <<https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-an-ellipsoid-vincenty-s-formulae/ba-p/902053>>. Acesso em: 27 set. 2024.

LELOUDAS, P. **Introduction to software testing**. Berkeley, CA: Apress, 2023.

MACHADO, C. **Um panorama dos acidentes por animais peçonhentos no Brasil**. *Journal Health NPEPS*, v. 1, n. 1, p. 1-3, 2016.

MATOS, R. R.; IGNOTTI, E. **Incidência de acidentes ofídicos por gêneros de serpentes nos biomas brasileiros**. *Ciência & Saúde Coletiva*, v. 25, p. 2837-2846, 2020.

NEGLECTED TROPICAL DISEASES. **World Health Organization**. 2024. Disponível em: <[https://www.who.int/health-topics/neglected-tropical-diseases#tab=tab\\_1](https://www.who.int/health-topics/neglected-tropical-diseases#tab=tab_1)>. Acesso em: 27 set. 2024.

OLIVEIRA, V. A. **Desenvolvimento de uma interface de usuário em linguagem Python para monitoramento de grandezas elétricas em tempo real**. 2023. TCC – Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2023.

PAIM, J. S.; et al. **O que é o SUS**. 1. ed. e-book interativo. Rio de Janeiro: Editora Fiocruz, 2015. 93 p. Apoio: Faperj. ISBN 978-85-7541-453-8. Disponível em: <<https://editorafiocruz.com.br/o-que-e-o-sus-ebook-interativo>>. Acesso em: 20 set. 2024.

PARISE, E. V. **Vigilância e monitoramento dos acidentes por animais peçonhentos no município de Palmas, Tocantins, Brasil**. *Hygeia-Revista Brasileira de Geografia Médica e da Saúde*, v. 12, n. 22, p. 72-87, 2016.

PERNAMBUCO. **CIATox - Centro de Assistência Toxicológica de Pernambuco**. Secretaria Estadual de Saúde. 2024. Disponível em: <<https://portal.saude.pe.gov.br/ciatox/>>. Acesso em: 27 set. 2024.

PERNAMBUCO. **MS oficializa macrorregião interestadual de saúde entre Pernambuco e Bahia, a primeira do Brasil**. 2024. Disponível em: <<https://portal.saude.pe.gov.br/ms-oficializa-macrorregiao-interestadual-de-saude-entre-pernambuco-e-bahia-a-primeira-do-brasil/>>. Acesso em: 11 out. 2024.

REIS, L. Q. **Sistema de Automação de Infraestrutura para Aplicação de Edge Computing**. 2024. TCC – Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2024.

SILVA, F. S. Q. **Avaliação da pureza de soros antiofídicos brasileiros e desenvolvimento de nova metodologia para essa finalidade**. 2008. 186 f. Tese (Doutorado) — Programa de Pós-Graduação em Vigilância Sanitária, Instituto Nacional de Controle de Qualidade em Saúde, Fundação Oswaldo Cruz, Rio de Janeiro, 2008.

SILVA, M. D. **Aplicação da Ferramenta Google Colaboratory para o Ensino da Linguagem Python**. Escola Regional de Engenharia de Software (ERES, 2020); Evento Online. Anais... Sociedade Brasileira de Computação, 2020. p. 67–76.

SOARES, R. S., SANTOS, S. P., MARQUES, C. C., et al. **Implementação de um laboratório virtual de física médica**. Revista Brasileira de Ensino de Física, v. 46, Porto Alegre, 2024.

SOARES, J. R. H. S.; BEZERRA, A. C. V. .; SÁ, A. J. de . O. **O federalismo sanitário brasileiro e a regionalização da saúde no estado de Pernambuco**. Hygeia - Revista Brasileira de Geografia Médica e da Saúde, Uberlândia, v. 15, n. 34, p. 115–128, 2019.

SOUZA, B. P. A.; ALBUQUERQUE P. C. **Projeto Mais Médicos para o Brasil em Pernambuco**: uma abordagem inicial. Tempus Actas de Saúde Coletiva. 2015.

VALQUI, J. **Descubre qué continente se extiende por los 4 hemisferios y su vínculo con Perú**. La República. 2024. Disponível em: <<https://larepublica.pe/mundo/2024/01/27/el-unico-continente-ubicado-en-los-4-hemisferios-y-que-guarda-una-relacion-con-peru-kiribati-africa-sudamerica-europa-743202>>. Acesso em: 20 set. 2024.

ZEMERO, M. I. M.; CHRISTINO, M. G.; FRANCO, M. C. A. **Acidentes com animais peçonhentos na região Norte do Brasil na série histórica de 2007–2017**. Pará Research Medical Journal, v. 6, 2022.

## Anexo A - Doenças Tropicais Negligenciadas

Tabela A.1 – *Neglected tropical diseases*

<b>DTNs</b>
<i>Buruli ulcer</i>
<i>Chagas disease</i>
<i>Dengue and Chikungunya</i>
<i>Dracunculiasis</i>
<i>Echinococcosis</i>
<i>Foodborne trematodiasis</i>
<i>Human African trypanosomiasis</i>
<i>Leishmaniasis</i>
<i>Leprosy</i>
<i>Lymphatic filariasis</i>
<i>Mycetoma, chromoblastomycosis, and other deep mycoses</i>
<i>Noma</i>
<i>Onchocerciasis</i>
<i>Rabies</i>
<i>Scabies and other ectoparasitoses</i>
<i>Schistosomiasis</i>
<i>Soil-transmitted helminthiasis</i>
<i>Snakebite envenoming</i>
<i>Taeniasis/cysticercosis</i>
<i>Trachoma</i>
<i>Yaws</i>

Fonte: (NEGLECTED TROPICAL DISEASES, 2024)

## Anexo B - Hospitais de Referência para AAPs em Pernambuco

Tabela B.1 – Hospitais de Referência de Atendimento - Pernambuco

MUNICÍPIO	UNIDADE DE SAÚDE	ENDEREÇO	TELEFONES	CNES	ATENDIMENTOS DISPONÍVEIS
Algodões da Ingazeira	Hospital Regional Emília Câmara	Rua São Paulo, km 1 PE 320, s/n - Alto Bela Vista	(87) 3838-8845	2428385	Escorpíonico
Arcoverde	Hospital e Policlínica Rui de Barros Correia	Avenida Agamenon Magalhães, s/n - São Miguel	(87) 3821-8296	2551764	Botrópico, Cratálico, Elapídico, Escorpíonico, Fonelútrico, Lovoscélico
Caruaru	Hospital Mestre Vitalino	Avenida Amazonas, 175 - Bairro Universitário	(81) 3725-7750	7488810	Botrópico, Cratálico, Elapídico, Escorpíonico, Fonelútrico, Lovoscélico
Garanhuns	Hospital Dom Moura	Avenida Simões Gomes, s/n - Heliópolis	(87) 3761-6100	2702983	Escorpíonico
Goiânia	Hospital Belarmino Correia	Praça Correia Picango, s/n - Jardim Novo Mundo	(81) 3626-8639	2711885	Escorpíonico
Jaboatão dos Guararapes	Hospital Jaboatão Prazeres	Rua Recife, s/n - Cajueiro Seco	(81) 3184-4201	2711990	Escorpíonico
Limoeiro	Hospital Regional de Limoeiro	Rua Padre Nicolau Pimentel, 224 - Centro	(81) 3628-8800	2712032	Escorpíonico
Quircuri	Hospital Regional Fernando Bezerra	Rua Teobaldo Gomes Torres, 8 - Centro	(87) 3874-4844	2712040	Botrópico, Cratálico, Elapídico, Escorpíonico, Fonelútrico, Lovoscélico
Palmares	Hospital Regional Silvio Magalhães	RR 101, km 185, s/n - Engenho Quilombo dos Palmares	(81) 3661-8400	2438393	Escorpíonico
Petrolina	Hospital Universitário de Petrolina (HU-UNIVASF)	Avenida José de Sá Meneses, s/n - Centro	(87) 2101-6501/6526	6002414	Botrópico, Cratálico, Elapídico, Escorpíonico, Fonelútrico, Lovoscélico
Recife	Hospital Dom Melairi/MIIP	Avenida Joaquim Nabuco, s/n - Centro	(87) 3202-7000	2430711	Escorpíonico
Salgueiro	Hospital Regional Inácio de Sá	Rua Antonio de Alencar Sampaio, 346 - Pimalto	(800-7226007)/(81) 3181-6463	0006655	Botrópico, Cratálico, Elapídico, Escorpíonico, Fonelútrico, Lovoscélico, Laquélico
Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior, 955 - Nossa Senhora da Penha	(87) 3831-9600	2348489	Cratálico, Elapídico, Escorpíonico, Fonelútrico, Lovoscélico, Laquélico, Botrópico
Vitória de Santo Antão	Hospital João Murilo	Avenida Henrique de Holanda, 87 - Matriz	(81) 3526-8833	2712008	Escorpíonico

Fonte: (BRASIL, 2023)

# Anexo C - Ficha de Notificação para Acidentes por Animais Peçonhentos

Figura C.1 – Ficha de notificações de AAP - página 1

República Federativa do Brasil Ministério da Saúde		SINAN SISTEMA DE INFORMAÇÃO DE AGRAVOS DE NOTIFICAÇÃO FICHA DE INVESTIGAÇÃO		Nº
<b>ACIDENTES POR ANIMAIS PEÇONHENTOS</b>				
<b>CASO CONFIRMADO:</b> Paciente com evidências clínicas de envenenamento, específicas para cada tipo de animal independentemente do animal causador do acidente ter sido identificado ou não. Não há necessidade de preenchimento da ficha para casos suspeitos.				
Dados Gerais	1	Tipo de Notificação 2 - Individual		
	2	Agravado/doença <b>ACIDENTES POR ANIMAIS PEÇONHENTOS</b>		3
	4	5	Código (CID10) X 29	6
Dados de Residência	7	Município de Notificação Código (IBGE)		
	8	Unidade de Saúde (ou outra fonte notificadora) Código		9
	10	Data dos Primeiros Sintomas		
Notificação Individual	11	Nome do Paciente		
	12	Data de Nascimento		
	13	(ou) Idade 1 - Hora 2 - Dia 3 - Mês 4 - Ano		
	14	Sexo M - Masculino F - Feminino 1 - Ignorado		
Dados de Residência	15	Cidade		
	16	Estado		
	17	Município de Residência Código (IBGE)		
	18	Distrito		19
	20	Bairro		
	21	Logradouro (rua, avenida,...) Código		
	22	Número		23
	24	Complemento (apto., casa, ...)		
	25	Geo campo 2		26
	27	Geo campo 1		
Dados Complementares do Caso	28	Ponto de Referência		
	29	CEP		30
	31	DDD) Telefone		
	32	Zona 1 - Urbana 2 - Rural 3 - Periurbana 9 - Ignorado		
Dados Complementares do Caso	33	Pais (se residente fora do Brasil)		
	34	Data da Investigação		
	35	Ocupação		36
	37	Data do Acidente		
Dados Complementares do Caso	38	Município de Ocorrência do Acidente: Código (IBGE)		39
	40	Localidade de Ocorrência do Acidente:		
	41	Zona de Ocorrência 1 - Urbana 2 - Rural 3 - Periurbana 9 - Ignorado		
	42	Tempo Decorrido Picada/Atendimento 1) 0-1h 2) 1-3h 3) 3-6h 4) 6-12h 5) 12-24h 6) 24 e + h 9) Ignorado		
Dados Clínicos	43	Local da Picada 01 - Cabeça 02 - Braço 03 - Antebraço 04 - Mão 05 - Dedo da Mão 06 - Tronco 07 - Caxa 08 - Perna 09 - Pé 10 - Dedo do Pé 99 - Ignorado		
	44	Manifestações Locais 1 - Sim 2 - Não 9 - Ignorado		
	45	Se Manifestações Locais Sim, especificar: 1 - Sim 2 - Não 9 - Ignorado Dor Edema Equimose Necrose Outras (Espec.)		
	46	Manifestações Sistêmicas 1 - Sim 2 - Não 9 - Ignorado neuroparalíticas (ptose palpebral, turvação visual) hemorrágicas (gengivorragia, outros sangramentos) vagais (vômitos, diarreias) miolíticas/hemolíticas (mialgia, anemia, urina escura) renais (oligúria/anúria) Outras (Espec.)		
Dados do Acidente	47	Tempo de Coagulação 1 - Normal 2 - Alterado 9 - Não realizado		
	48	Tipo de Acidente 1 - Serpente 2 - Aranha 3 - Escorpião 4 - Lagarta 5 - Abelha 6 - Outros 9 - Ignorado		
	49	Serpente - Tipo de Acidente 1 - Botrópico 2 - Crotálico 3 - Elapídico 4 - Laquético 5 - Serpente Não Peçonhenta 9 - Ignorado		
	50	Aranha - Tipo de Acidente 1 - Foneutrismo 2 - Loxoscelismo 3 - Latrodectismo 4 - Outra Aranha 9 - Ignorado		
	51	Lagarta - Tipo de Acidente 1 - Lonómia 2 - Outra lagarta 9 - Ignorado		

Figura C.2 – Ficha de notificações de AAP - página 2

Tratamento	49) Classificação do Caso 1 - Leve 2 - Moderado 3 - Grave 9 - Ignorado <input type="checkbox"/>		50) Soroterapia 1 - Sim 2 - Não 9 - Ignorado <input type="checkbox"/>	
	51) Se Soroterapia Sim, especificar número de ampolas de soro:			
	Antibotrópico (SAB) <input type="checkbox"/>	Anticrotálico (SAC) <input type="checkbox"/>	Antiaracnídico (SAAr) <input type="checkbox"/>	
	Antibotrópico-laquético (SABL) <input type="checkbox"/>	Antielapídico (SAE) <input type="checkbox"/>	Antiloxoscélico (SALox) <input type="checkbox"/>	
Antibotrópico-crotálico (SABC) <input type="checkbox"/>	Antiescorpiónico (SAEs) <input type="checkbox"/>	Antilonômico (SALon) <input type="checkbox"/>		
Condição	52) Complicações Locais <input type="checkbox"/> 1 - Sim 2 - Não 9 - Ignorado		53) Se Complicações Locais Sim, especificar: 1 - Sim 2 - Não 9 - Ignorado	
	<input type="checkbox"/>		<input type="checkbox"/> Infecção Secundária <input type="checkbox"/> Necrose Extensa <input type="checkbox"/> Síndrome Compartimental <input type="checkbox"/> Déficit Funcional <input type="checkbox"/> Amputação	
	54) Complicações Sistêmicas <input type="checkbox"/> 1 - Sim 2 - Não 9 - Ignorado		55) Se Complicações Sistêmicas Sim, especificar: 1 - Sim 2 - Não 9 - Ignorado	
	<input type="checkbox"/>		<input type="checkbox"/> Insuficiência Renal <input type="checkbox"/> Insuficiência Respiratória / Edema Pulmonar Agudo <input type="checkbox"/> Septicemia <input type="checkbox"/> Choque	
56) Acidente Relacionado ao Trabalho 1 - Sim <input type="checkbox"/> 2 - Não <input type="checkbox"/> 9 - Ignorado <input type="checkbox"/>		57) Evolução do Caso <input type="checkbox"/> 1-Cura <input type="checkbox"/> 2-Óbito por acidentes por animais peçonhentos <input type="checkbox"/> 3-Óbito por outras causas <input type="checkbox"/> 9-Ignorado <input type="checkbox"/>		58) Data do Óbito <input type="text"/>
				59) Data do Encerramento <input type="text"/>
<b>Acidentes com animais peçonhentos: manifestações clínicas, classificação e soroterapia</b>				
	<b>Tipo</b>	<b>Manifestações Clínicas</b>	<b>Tipo Soro</b>	<b>Nº ampolas</b>
ORFISMO	<b>Botrópico</b> jararaca jararacuçu urutu caicaca	Leve: dor, edema local e equimose discreto	SAB	2 - 4
		Moderado: dor, edema e equimose evidentes, manifestações hemorrágicas discretas		4 - 8
		Grave: dor e edema intenso e extenso, bolhas, hemorragia intensa, oligoanúria, hipotensão		12
	<b>Crotálico</b> cascavel boicivanga	Leve: ptose palpebral, turvação visual discretos de aparecimento tardio, sem alteração da cor da urina, mialgia discreta ou ausente	SAC	5
		Moderado: ptose palpebral, turvação visual discretos de início precoce, mialgia discreta, urina escura		10
		Grave: ptose palpebral, turvação visual evidentes e intensos, mialgia intensa e generalizada, urina escura, oligúria ou anúria		20
<b>Laquético</b> surucuru pico-de-jaca	Moderado: dor, edema, bolhas e hemorragia discreta	SABL	10	
	Grave: dor, edema, bolhas, hemorragia, cólicas abdominais, diarreia, bradicardia, hipotensão arterial		20	
<b>Elapídico</b> coral verdadeira	Grave: dor ou parestesia discreta, ptose palpebral, turvação visual	SAEL	10	
ESCORPIONISMO	<b>Escorpiónico</b> escorpião	Leve: dor, eritema e parestesia local	SAEsc ou SAA	---
		Moderado: sudorese, náuseas, vômitos ocasionais, taquicardia, agitação e hipertensão arterial leve		2 - 3
		Grave: vômitos profusos e incoercíveis, sudorese profusa, prostração, bradicardia, edema pulmonar agudo e choque		4 - 6
ARANHEISMO	<b>Loxoscélico</b> aranha-marrom	Leve: lesão incaracterística sem aranha identificada	SAA ou SALox	---
		Moderado: lesão sugestiva com equimose, palidez, eritema e edema endurecido local, cefaléia, febre, exantema		5
		Grave: lesão característica, hemólise intravascular		10
	<b>Foneutrismo</b> aranha-armadeira aranha-da-banana	Leve: dor local	SAA	---
Moderado: sudorese ocasional, vômitos ocasionais, agitação, hipertensão arterial	2 - 4			
Grave: sudorese profusa, vômitos frequentes, priapismo, edema pulmonar agudo, hipotensão arterial	5 - 10			
LONCOMIA	Lafurana oruga	Leve: dor, eritema, adenomegalia regional, coagulação normal, sem hemorragia	SALon	---
		Moderado: alteração na coagulação, hemorragia em pele e/ou mucosas		5
		Grave: alteração na coagulação, hemorragia em vísceras, insuficiência renal		10
<b>Informações complementares e observações</b>				
Anotar todas as informações consideradas importantes e que não estão na ficha (ex: outros dados clínicos, dados laboratoriais, laudos de outros exames e necropsia, etc.)				
Investigador	Município/Unidade de Saúde		Cód. da Unid. de Saúde	
	Nome	Função	Assinatura	
Animais Peçonhentos		Sinan Net	SVS 19/01/2006	

Fonte: Próprio autor

## Anexo D - Tabelas de Hospitais por Soro

Figura D.1 – Tabela de hospitais com soro antibotrópico

MUNICIPIO	UNIDADE_DE_SAUDE	ENDERECO	LATITUDE	LONGITUDE	TELEFONES	CNES	SORO	ANIMAL	
0	Arcoverde	Hospital e Policlínica Rui de Barros Correia	Avenida Agamenon Magalhães s/n - São Miguel	-8.422456	-37.054930	(87) 3821-8296	2551764	Soro Antibotrópico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
1	Caruaru	Hospital Mestre Vitalino	Avenida Amazonas 175 - Bairro Universitário	-8.245172	-35.973795	(81) 3725-7750	7498810	Soro Antibotrópico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
2	Ouricuri	Hospital Regional Fernando Bezerra	Rua Teobaldo Gomes Torres 8 - Centro	-7.882962	-40.084261	(87) 3874-4844	2712040	Soro Antibotrópico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
3	Petrolina	Hospital Universitário de Petrolina (HU-UNIVASF)	Avenida José de Sá Maniçoba s/n - Centro	-9.392355	-40.495607	(87) 2101-6501/6526	6042414	Soro Antibotrópico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
4	Recife	Hospital da Restauração	Avenida Gov. Agamenon Magalhães s/n - Derby	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	655	Soro Antibotrópico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
5	Salgueiro	Hospital Regional Inácio de Sá	Rua Antonio de Alencar Sampaio 346 - Planalto	-8.058329	-39.113021	(87) 3276-1190	2356287	Soro Antibotrópico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)
6	Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior 955 - Nossa Senhora da...	-7.988751	-38.298864	(87) 3831-9600	2348489	Soro Antibotrópico (SAB) - Jararaca, Jararacuçu	Serpente (cobra)

Fonte: Próprio autor

Figura D.2 – Tabela de hospitais com soro anticrotático

MUNICIPIO	UNIDADE_DE_SAUDE	ENDERECO	LATITUDE	LONGITUDE	TELEFONES	CNES	SORO	ANIMAL	
0	Arcoverde	Hospital e Policlínica Rui de Barros Correia	Avenida Agamenon Magalhães s/n - São Miguel	-8.422456	-37.054930	(87) 3821-8296	2551764	Soro Anticrotático (SAC) - cascavel	Serpente (cobra)
1	Caruaru	Hospital Mestre Vitalino	Avenida Amazonas 175 - Bairro Universitário	-8.245172	-35.973795	(81) 3725-7750	7498810	Soro Anticrotático (SAC) - cascavel	Serpente (cobra)
2	Ouricuri	Hospital Regional Fernando Bezerra	Rua Teobaldo Gomes Torres 8 - Centro	-7.882962	-40.084261	(87) 3874-4844	2712040	Soro Anticrotático (SAC) - cascavel	Serpente (cobra)
3	Petrolina	Hospital Universitário de Petrolina (HU-UNIVASF)	Avenida José de Sá Maniçoba s/n - Centro	-9.392355	-40.495607	(87) 2101-6501/6526	6042414	Soro Anticrotático (SAC) - cascavel	Serpente (cobra)
4	Recife	Hospital da Restauração	Avenida Gov. Agamenon Magalhães s/n - Derby	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	655	Soro Anticrotático (SAC) - cascavel	Serpente (cobra)
5	Salgueiro	Hospital Regional Inácio de Sá	Rua Antonio de Alencar Sampaio 346 - Planalto	-8.058329	-39.113021	(87) 3276-1190	2356287	Soro Anticrotático (SAC) - cascavel	Serpente (cobra)
6	Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior 955 - Nossa Senhora da...	-7.988751	-38.298864	(87) 3831-9600	2348489	Soro Anticrotático (SAC) - cascavel	Serpente (cobra)

Fonte: Próprio autor

Figura D.3 – Tabela de hospitais com soro antielapídico

MUNICIPIO	UNIDADE_DE_SAUDE	ENDEREÇO	LATITUDE	LONGITUDE	TELEFONES	CNES	SORO	ANIMAL	
0	Arcoverde	Hospital e Policlínica Rui de Barros Correia	Avenida Agamenon Magalhães s/n - São Miguel	-8.422456	-37.054930	(87) 3821-8296	2551764	Soro Antielapídico (SAE) - coral verdadeira	Serpente (cobra)
1	Caruaru	Hospital Mestre Vitalino	Avenida Amazonas 175 - Bairro Universitário	-8.245172	-35.973795	(81) 3725-7750	7498810	Soro Antielapídico (SAE) - coral verdadeira	Serpente (cobra)
2	Ouricuri	Hospital Regional Fernando Bezerra	Rua Teobaldo Gomes Torres 8 - Centro	-7.882962	-40.084261	(87) 3874-4844	2712040	Soro Antielapídico (SAE) - coral verdadeira	Serpente (cobra)
3	Petrolina	Hospital Universitário de Petrolina (HU-UNIVASF)	Avenida José de Sá Maniçoba s/n - Centro	-9.392355	-40.495607	(87) 2101-6501/6526	6042414	Soro Antielapídico (SAE) - coral verdadeira	Serpente (cobra)
4	Recife	Hospital da Restauração	Avenida Gov. Agamenon Magalhães s/n - Derby	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	655	Soro Antielapídico (SAE) - coral verdadeira	Serpente (cobra)
5	Salgueiro	Hospital Regional Inácio de Sá	Rua Antonio de Alencar Sampaio 346 - Planalto	-8.058329	-39.113021	(87) 3276-1190	2356287	Soro Antielapídico (SAE) - coral verdadeira	Serpente (cobra)
6	Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior 955 - Nossa Senhora da...	-7.988751	-38.298864	(87) 3831-9600	2348489	Soro Antielapídico (SAE) - coral verdadeira	Serpente (cobra)

Fonte: Próprio autor

Figura D.4 – Tabela de hospitais com soro antiescorpiônico

MUNICIPIO	UNIDADE_DE_SAUDE	ENDEREÇO	LATITUDE	LONGITUDE	TELEFONES	CNES	SORO	ANIMAL	
0	Afogados da Ingazeira	Hospital Regional Emília Câmara	Rua São Paulo Km 1 PE 320 s/n - Alto Bela Vista	-7.737366	-37.650591	(87) 3838-8845	2428385	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
1	Arcoverde	Hospital e Policlínica Rui de Barros Correia	Avenida Agamenon Magalhães s/n - São Miguel	-8.422456	-37.054930	(87) 3821-8296	2551764	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
2	Caruaru	Hospital Mestre Vitalino	Avenida Amazonas 175 - Bairro Universitário	-8.245172	-35.973795	(81) 3725-7750	7498810	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
3	Garanhuns	Hospital Dom Moura	Avenida Simoa Gomes s/n - Heliópolis	-8.882305	-36.482957	(87) 3761-6100	2702983	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
4	Goiana	Hospital Belarmino Correia	Praça Correia Picanço s/n - Jardim Novo Mundo	-7.557763	-35.002270	(81) 3626-8639	2711885	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
5	Jaboatão dos Guararapes	Hospital Jabotão Prazeres	Rua Recife s/n - Cajueiro Seco	-8.165936	-34.923169	(81) 3184-4201	2711990	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
6	Limoeiro	Hospital Regional de Limoeiro	Rua Padre Nicolau Pimentel 224 - Centro	-7.881954	-35.448565	(81) 3628-8800	2712032	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
7	Ouricuri	Hospital Regional Fernando Bezerra	Rua Teobaldo Gomes Torres 8 - Centro	-7.882962	-40.084261	(87) 3874-4844	2712040	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
8	Palmares	Hospital Regional Silvío Magalhães	BR 101 Km 185 s/n - Engenho Quilombo dos Palmares	-8.686613	-35.574855	(81) 3661-8400	2428393	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
9	Petrolina	Hospital Universitário de Petrolina (HU-UNIVASF)	Avenida José de Sá Maniçoba s/n - Centro	-9.392355	-40.495607	(87) 2101-6501/6526	6042414	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
10	Petrolina	Hospital Dom Malan/IMIP	Avenida Joaquim Nabuco s/n - Centro	-9.394637	-40.499618	(87) 3202-7000	2430711	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
11	Recife	Hospital da Restauração	Avenida Gov. Agamenon Magalhães s/n - Derby	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	655	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
12	Salgueiro	Hospital Regional Inácio de Sá	Rua Antonio de Alencar Sampaio 346 - Planalto	-8.058329	-39.113021	(87) 3276-1190	2356287	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
13	Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior 955 - Nossa Senhora da...	-7.988751	-38.298864	(87) 3831-9600	2348489	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião
14	Vitória de Santo Antão	Hospital João Murilo	Avenida Henrique de Holanda 87 - Matriz	-8.112985	-35.288086	(81) 3526-8833	2712008	Soro Antiescorpiônico (SAEsc) - escorpião amarelo	Escorpião

Fonte: Próprio autor

Figura D.5 – Tabela de hospitais com soro antiaracnídico

MUNICIPIO	UNIDADE_DE_SAUDE	ENDERECO	LATITUDE	LONGITUDE	TELEFONES	CNES	SORO	ANIMAL	
0	Arcoverde	Hospital e Policlínica Rui de Barros Correia	Avenida Agamenon Magalhães s/n - São Miguel	-8.422456	-37.054930	(87) 3821-8296	2551764	Soro Antiaracnídico (SAA) - aranha armadeira	Aranha
1	Caruaru	Hospital Mestre Vitalino	Avenida Amazonas 175 - Bairro Universitário	-8.245172	-35.973795	(81) 3725-7750	7498810	Soro Antiaracnídico (SAA) - aranha armadeira	Aranha
2	Ouricuri	Hospital Regional Fernando Bezerra	Rua Teobaldo Gomes Torres 8 - Centro	-7.882962	-40.084261	(87) 3874-4844	2712040	Soro Antiaracnídico (SAA) - aranha armadeira	Aranha
3	Petrolina	Hospital Universitário de Petrolina (HU-UNIVASF)	Avenida José de Sá Maniçoba s/n - Centro	-9.392355	-40.495607	(87) 2101-6501/6526	6042414	Soro Antiaracnídico (SAA) - aranha armadeira	Aranha
4	Recife	Hospital da Restauração	Avenida Gov. Agamenon Magalhães s/n - Derby	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	655	Soro Antiaracnídico (SAA) - aranha armadeira	Aranha
5	Salgueiro	Hospital Regional Inácio de Sá	Rua Antonio de Alencar Sampaio 346 - Planalto	-8.058329	-39.113021	(87) 3276-1190	2356287	Soro Antiaracnídico (SAA) - aranha armadeira	Aranha
6	Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior 955 - Nossa Senhora da...	-7.988751	-38.298864	(87) 3831-9600	2348489	Soro Antiaracnídico (SAA) - aranha armadeira	Aranha

Fonte: Próprio autor

Figura D.6 – Tabela de hospitais com soro antilaquéutico

MUNICIPIO	UNIDADE_DE_SAUDE	ENDERECO	LATITUDE	LONGITUDE	TELEFONES	CNES	SORO	ANIMAL	
0	Recife	Hospital da Restauração	Avenida Gov. Agamenon Magalhães s/n - Derby	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	655	Soro Antilaquéutico (SALaq) - surucucu	Serpente (cobra)
1	Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior 955 - Nossa Senhora da...	-7.988751	-38.298864	(87) 3831-9600	2348489	Soro Antilaquéutico (SALaq) - surucucu	Serpente (cobra)

Fonte: Próprio autor

Figura D.7 – Tabela de hospitais com soro antiloxoscélico

MUNICIPIO	UNIDADE_DE_SAUDE	ENDERECO	LATITUDE	LONGITUDE	TELEFONES	CNES	SORO	ANIMAL	
0	Arcoverde	Hospital e Policlínica Rui de Barros Correia	Avenida Agamenon Magalhães s/n - São Miguel	-8.422456	-37.054930	(87) 3821-8296	2551764	Soro Antiloxoscélico (SALox) - aranha marrom	Aranha
1	Caruaru	Hospital Mestre Vitalino	Avenida Amazonas 175 - Bairro Universitário	-8.245172	-35.973795	(81) 3725-7750	7498810	Soro Antiloxoscélico (SALox) - aranha marrom	Aranha
2	Ouricuri	Hospital Regional Fernando Bezerra	Rua Teobaldo Gomes Torres 8 - Centro	-7.882962	-40.084261	(87) 3874-4844	2712040	Soro Antiloxoscélico (SALox) - aranha marrom	Aranha
3	Petrolina	Hospital Universitário de Petrolina (HU-UNIVASF)	Avenida José de Sá Maniçoba s/n - Centro	-9.392355	-40.495607	(87) 2101-6501/6526	6042414	Soro Antiloxoscélico (SALox) - aranha marrom	Aranha
4	Recife	Hospital da Restauração	Avenida Gov. Agamenon Magalhães s/n - Derby	-8.053359	-34.898141	0800-7226001/(81) 3181-6453	655	Soro Antiloxoscélico (SALox) - aranha marrom	Aranha
5	Salgueiro	Hospital Regional Inácio de Sá	Rua Antonio de Alencar Sampaio 346 - Planalto	-8.058329	-39.113021	(87) 3276-1190	2356287	Soro Antiloxoscélico (SALox) - aranha marrom	Aranha
6	Serra Talhada	Hospital Regional Professor Agamenon Magalhães	Rua Comandante Superior 955 - Nossa Senhora da...	-7.988751	-38.298864	(87) 3831-9600	2348489	Soro Antiloxoscélico (SALox) - aranha marrom	Aranha

Fonte: Próprio autor

## Anexo E - Código-Fonte do Sistema de Mapeamento

```

import pandas as pd
import streamlit as st
import streamlit_js_eval
import folium
from streamlit_folium import st_folium, folium_static

# Configurações da página
st.set_page_config(
    page_title="Mapeamento de Soros Antiveneno em Pernambuco",
    page_icon='doc/cobra.png',
    layout="wide",
    initial_sidebar_state='collapsed'
)

col1, col2, col3 = st.columns([1,4,1])

col1.image('doc/cobra.png', width=200)
col2.title('Mapeamento de Soros Antiveneno em Pernambuco')
#col3.image('doc/deptEngBio.png', width=300)

st.markdown(
    """
    <style>
    p {
        font-weight: bold; /* Negrito */
    }
    .st-emotion-cache-10trblm.e1nzilvr1 {
        color: #FF5733;
    }
    </style>
    """
    ,
    unsafe_allow_html=True,
)

with col2:

    tab1, tab2, tab3, tab4 = st.tabs([ "Buscador de Soros", "CEATOX-PE", "Sobre", "Contato"])

    with tab1:
        st.markdown(
            """
            <h2 style="text-align: center;">Buscador de Soros</h2><br><br>
            """
            ,
            unsafe_allow_html=True,
        )

    dados_geral = pd.read_csv('doc/hospital_mais_proximo.csv')
    cidades = pd.read_csv('doc/latitude-longitude-cidades.csv')

```

```

cidades = cidades[cidades['uf'] == 'PE']

lista_mun_distinct = sorted(cidades['municipio'].unique())

#municipios
col5,espaco2,col4 = st.columns([5,1,7])
with col5:
    st.markdown(
        """
        <br><br><br>
        """
        ,
        unsafe_allow_html=True,
    )
    animal = st.selectbox("Qual tipo de animal causou o acidente?",
dados_geral['animal'].unique(), index=None, placeholder="Selecione o animal")
    soro = st.selectbox('Soro Antiveneno',
dados_geral[dados_geral['animal']==animal]['soro'].unique(),
index=None,
placeholder="Selecione o Soro Antiveneno")

    try:
        container = st.container(border=True)
        with container:
            st.write(soro, dicionario_explicacao[soro])
    except:
        st.write("")

    mun_origem = st.selectbox('Município onde está o paciente', lista_mun_distinct,
index=None, placeholder="Selecione o município onde está o paciente")

    try:
        #Filtro destino
        filtro = (dados_geral['soro'] == soro)&(dados_geral['cidade_origem'] ==
mun_origem)
        municipio_origem = dados_geral[filtro]
        municipio_origem['Legenda'] = 'Origem'

        #municipio_origem = municipio_origem.reset_index(drop=True)
        mun_destino = municipio_origem.dropna()['cidade_hospital'].values[0]

        filtro_destino = (dados_geral['soro'] == soro)&(dados_geral['cidade_origem'] ==
mun_destino)
        municipio_destino = dados_geral[filtro_destino].dropna()
        municipio_destino['Legenda'] = 'Destino'

        latitude_media = (municipio_origem['latitude_origem'].values[0] +
municipio_destino['latitude_hospital'].values[0])/2
        longitude_media = (municipio_origem['longitude_origem'].values[0] +
municipio_destino['longitude_hospital'].values[0])/2

        mapa = folium.Map([latitude_media, longitude_media], zoom_start=9.6)

        folium.Marker(
            location= [municipio_origem['latitude_origem'].values,
municipio_origem['longitude_origem'].values],

```

```

        tooltip="Origem",
        popup="Você está aqui",
        icon=folium.Icon(color="green",icon="house",prefix="fa"),
    ).add_to(mapa)

    folium.Marker(
        location= [municipio_destino['latitude_hospital'].values,
municipio_destino['longitude_hospital'].values],
        tooltip="Destino",
        popup=f"O soro está aqui, na cidade de
{municipio_destino['cidade_hospital'].values[0]}, {municipio_destino['hospital'].values[0]}",
        icon=folium.Icon(color="red", icon="hospital",prefix="fa"),
    ).add_to(mapa)

#folium.TileLayer('MapQuest Open Aerial').add_to(mapa)

with col4:
    st.subheader('Hospital mais próximo')
    st_data = folium_static(mapa, width=710, height=500)
with col5:
    mun_destino = municipio_origem.dropna()['cidade_hospital'].values[0]
    distancia = municipio_origem.dropna()['distancia_km'].values[0]
    local = municipio_origem.dropna()['hospital'].values[0]
    endereco = municipio_origem.dropna()['endereco'].values[0]
    telefone = municipio_origem.dropna()['telefone'].values[0]
    container_respostas = st.container(border=True)
    with container_respostas:
        st.write(f'Município onde está o soro mais próximo: **{mun_destino}**')
        st.write(f'Local: **{local}**')
        st.write(f'Endereço: **{endereco}**')
        st.write(f'Telefone: **{telefone}**')
        st.write(f'Distância: **{distancia} km**')
        st.write('**ATENÇÃO**': ligue para o local para fazer a confirmação da
disponibilidade do soro.')
except:
    with col4:
        if soro:
            filtro = (dados_geral['soro'] == soro)&(dados_geral['animal'] == animal)
            dados_mapa_vazio = dados_geral[filtro]

        elif animal:
            filtro = (dados_geral['animal'] == animal)
            dados_mapa_vazio = dados_geral[filtro]

        else:
            dados_mapa_vazio = dados_geral.copy()

    pontos = dados_mapa_vazio.drop_duplicates(['hospital'])

    mapa_vazio = folium.Map([-8.3831638, -37.7753284], zoom_start=7)

        for latitude, longitude, hospital, endereco in zip(pontos['latitude_hospital'],
pontos['longitude_hospital'], pontos['hospital'], pontos['endereco']):
            folium.Marker(

```

```

location= [latitude, longitude],
tooltip=hospital,
popup=endereco,
icon=folium.Icon(color="red", icon="hospital",prefix="fa"),
).add_to(mapa_vazio)

```

```

ceatox_html = """
    <h3>CEATOX - Centro de Assistência Toxicológica</h3>
    
    <p>O Ceatox faz o acompanhamento dos pacientes por meio de
evolução clínica diária,
    até a alta médica. Além da assistência aos casos de intoxicações e
acidentes por animais
    peçonhentos, o Centro de Informação e Assistência Toxicológica de
Pernambuco (CIAtox)
    desenvolve atividades de ensino e pesquisa através de parcerias com
instituições de
    ensino e pesquisa.
    <b>Telefone: 0800.722.6001 (tele-atendimento 24h por dia)</b></p>
    """

```

```

folium.Marker(
    location= [-8.053771365450142, -34.89005840177061],
    tooltip="CEATOX - PE",
    popup=ceatox_html,
    icon=folium.Icon(color="lightblue",icon="hospital",prefix="fa"),
).add_to(mapa_vazio)

```

```

st.subheader('Todos os hospitais')
st_data = folium_static(mapa_vazio, width=710, height=400)

```

```

creditos = st.container(border=True)
with creditos:
    col6,espaco3,col7,col8,espaco4 = st.columns([7,2,1,1,1])
    with col6:
        st.write('Aplicação desenvolvida pelo Departamento de Engenharia Biomédica da
Universidade Federal de Pernambuco.')
        st.write('Integrantes: José Filipe....')
    with col7:

st.image('https://upload.wikimedia.org/wikipedia/commons/a/ae/Logo-ufpe-2-2.jpg',width=100
)
    with col8:
        st.image('doc/deptEngBio.png', width=150)

```

## Anexo F - Modelo de Teste de GUI

Tabela F.1 – Tabela de teste de GUI

Teste	Objetivo	Procedimento	Resultado Esperado	Resultado Obtido
<b>Funcionalidade de Filtro: Animal</b>	Verificar se o filtro de tipo de animal funciona corretamente.	Selecionar diferentes tipos de animais.	O mapa deverá ser atualizado.	Em teste ▾
<b>Funcionalidade de Filtro: Soro</b>	Testar a funcionalidade de filtragem de soro após selecionar o animal.	Selecionar um animal e testar todas as combinações de soro disponíveis.	O sistema deve permitir a escolha de soro apenas após seleção do animal; O mapa deverá ser atualizado.	Em teste ▾
<b>Funcionalidade de Filtro: Cidade</b>	Verificar se o filtro de cidade funciona corretamente.	Verificar a lista de cidades e validar se estão em ordem alfabética.	A lista de cidades deve estar ordenada. O mapa deverá ser atualizado.	Em teste ▾
<b>Renderização de Mapa (Desktop)</b>	Verificar se o mapa é renderizado corretamente no Desktop.	Acessar a aplicação em um navegador desktop e verificar a visualização correta do mapa.	O mapa deve ser exibido sem falhas no notebook.	Em teste ▾
<b>Renderização de Mapa (Mobile)</b>	Verificar se o mapa é renderizado corretamente no celular.	Acessar a aplicação em um navegador mobile e verificar a visualização correta do mapa.	O mapa deve ser exibido sem falhas em dispositivos móveis.	Em teste ▾
<b>Marcações no Mapa</b>	Verificar se as marcações de hospitais e cidades estão corretas no mapa.	Selecionar qualquer cidade e verificar se os hospitais próximos aparecem no mapa com as marcações adequadas.	As marcações devem ser precisas e aparecer corretamente no mapa.	Em teste ▾
<b>Distância Cidade-Hospital (Filtro 1)</b>	Verificar se a distância entre uma cidade e o hospital mais próximo está correta (Filtro 1).	Selecionar os filtros e comparar a distância retornada com a do Google Maps.	A distância deve ser compatível com o valor obtido no Google Maps.	Em teste ▾
<b>Distância Cidade-Hospital (Filtro 2)</b>	Verificar se a distância entre uma cidade e o hospital mais próximo está correta (Filtro 2).	Selecionar outra combinação de cidade e hospital e comparar a distância com o	A distância deve ser compatível com o valor obtido no Google Maps.	Em teste ▾

		Google Maps.		
<b>Distância Cidade-Hospital (Filtro 3)</b>	Verificar se a distância entre uma cidade e o hospital mais próximo está correta (Filtro 3).	Selecionar uma terceira combinação de cidade e hospital e comparar a distância com o Google Maps.	A distância deve ser compatível com o valor obtido no Google Maps.	Em teste ▾
<b>Carregamento de Dados</b>	Testar a exibição dos dados do hospital ao lado do mapa.	Selecionar qualquer cidade e verificar se as informações do hospital mais próximo são exibidas corretamente.	Informações do hospital devem aparecer de forma completa e correta.	Em teste ▾
<b>Testes de Performance (Desktop)</b>	Avaliar o tempo de resposta ao carregar o mapa em dispositivos desktop.	Medir o tempo que a aplicação leva para processar e carregar o mapa no Desktop.	O mapa deve ser carregado em menos de 3 segundos.	Em teste ▾
<b>Testes de Performance (Mobile)</b>	Avaliar o tempo de resposta ao carregar o mapa em dispositivos móveis.	Medir o tempo que a aplicação leva para processar e carregar o mapa no celular.	O mapa deve ser carregado em menos de 3 segundos.	Em teste ▾

Fonte: Próprio autor



UNIVERSIDADE  
FEDERAL  
DE PERNAMBUCO

**DEBM**  
Dep'to de Engenharia Biomédica

# **DOCUMENTAÇÃO DO CÓDIGO-FONTE DO SISTEMA DE MAPEAMENTO DE HOSPITAIS POR SOROTERAPIA EM PERNAMBUCO**

**VERSÃO 1.0**

AUTOR: JOSÉ FILIPE SILVA DE ANDRADE  
REVISORA: PROF.<sup>a</sup> DR.<sup>a</sup> CRISTINE MARTINS GOMES DE GUSMÃO

**RECIFE**

**2024**



<b>Histórico de Revisões</b>			
<b>Versão</b>	<b>Data</b>	<b>Descrição</b>	<b>Autor</b>
1.0	30/10/2024	A versão inicial da documentação apresenta a estrutura do sistema de mapeamento de hospitais. Inclui uma visão geral do software, realização de caso de uso, explicação das funcionalidades principais, e a interação entre frontend e backend no streamlit.	José Filipe S. de Andrade

## Sumário

<b>1. Introdução.....</b>	<b>4</b>
<b>2. Configuração do Ambiente de Desenvolvimento.....</b>	<b>5</b>
2.1. Clonando o Repositório.....	5
2.2. Requisitos de Sistema.....	6
2.3. Instalação de Dependências.....	6
2.4. Configuração da IDE.....	6
2.5. Execução Local da Aplicação.....	7
2.6. Considerações Adicionais.....	8
<b>3. Visão Geral do Código-Fonte.....</b>	<b>8</b>
3.1. Carregamento de Bibliotecas e Arquivos Externos.....	8
3.2. Streamlit.....	9
3.3. Configurações Iniciais.....	10
3.4. Estrutura Layout.....	10
3.5. Formatação de Estilos.....	10
3.6. Estrutura de Abas.....	11
3.7. Leitura e Manipulação de Dados.....	11
3.8. Inputs do Usuário.....	12
3.9. Exibição de Mapas Interativos.....	14
3.10. Renderização do Mapa Interativo.....	15
3.11. Adicionando Marcadores ao Mapa.....	15
3.12. Exibição e Detalhes do Hospital.....	16
3.13. Default do Mapa.....	17
3.14. Créditos.....	19
<b>4. Caso de Uso: Função Principal do Sistema.....</b>	<b>19</b>
4.1. Descrição Técnica do Caso de Uso.....	20
<b>5. Testes.....</b>	<b>21</b>
5.1. Quadro de Testes.....	21
<b>6. Avaliação do Sistema.....</b>	<b>23</b>
<b>7. Referências.....</b>	<b>23</b>
<b>8. Aprovação da Revisão.....</b>	<b>23</b>

# Documento de Arquitetura de Software

## 1. Introdução

Este documento tem como objetivo descrever o projeto do Sistema de Mapeamento de Centros de Referência em Soroterapia no Estado de Pernambuco. O sistema foi desenvolvido como uma aplicação *web*, utilizando a linguagem *Python* e hospedado na plataforma *Streamlit Cloud*, para auxiliar na identificação do hospital mais próximo que oferece o tipo de soro antiveneno necessário, de acordo com o município de origem do paciente.

A aplicação é alimentada por um *dataframe* contendo dados sobre os hospitais de referência e permite ao usuário selecionar o tipo de animal causador do acidente, o soro antiveneno apropriado e o município onde o paciente se encontra. Com base nessas informações, o sistema utiliza as informações disponíveis no *dataframe* para apresentar a distância entre a cidade de origem e o hospital mais próximo que disponibiliza o soro requerido.

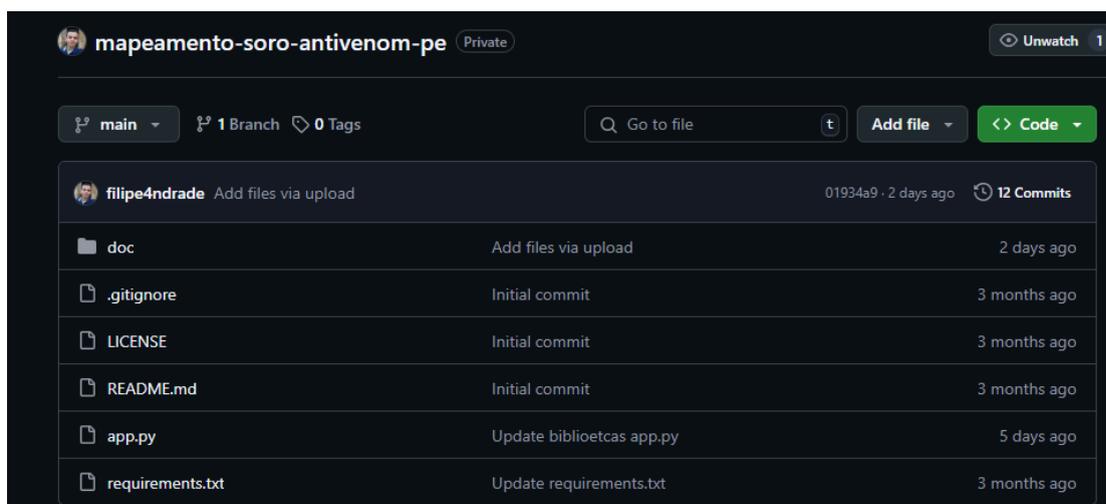
O sistema foi desenvolvido utilizando as seguintes bibliotecas: *pandas* para manipulação de dados, *folium* para geração dos mapas, e *streamlit* para a interface *web*.

Além disso, a aplicação também apresenta informações adicionais sobre cada hospital, incluindo o nome da instituição, endereço, telefone e a distância entre o hospital e o local onde o paciente se encontra. Isso oferece uma ferramenta útil para profissionais de saúde, gestores e a população em geral, visando otimizar o atendimento emergencial nesses casos.

Este documento visa detalhar as funcionalidades implementadas e as tecnologias que compõem o projeto, fornecendo uma visão clara e concisa do sistema a fim de que outros desenvolvedores possam contribuir para o sistema.

## 2. Configuração do Ambiente de Desenvolvimento

Este capítulo descreve o processo necessário para configurar o ambiente de desenvolvimento local para rodar e desenvolver o projeto. O código-fonte do projeto está disponível em um repositório do *GitHub*, permitindo que seja clonado e utilizado por outros desenvolvedores.



Abaixo estão detalhadas as etapas e requisitos para preparar o ambiente de desenvolvimento.

### 2.1. Clonando o Repositório

O primeiro passo para configurar o ambiente de desenvolvimento é clonar o repositório do *GitHub* onde o sistema está hospedado (<https://github.com/filipe4ndrade/mapeamento-soro-antivenom-pe>). Para isso, é necessário ter o *Git* instalado na sua máquina. Caso não tenha, você pode baixá-lo e instalá-lo a partir do site oficial (<https://git-scm.com/downloads>).

Uma vez com o *Git* instalado, abra o terminal ou o *prompt* de comando do *Git Bash* em alguma pasta local, e execute o comando **git clone repositório** (repositório é a url do projeto no *github* )para clonar o repositório, assim:



```
Filipe@DESKTOP-LSPD6G1 MINGW64 ~/Documents  
$ git clone https://github.com/filipe4ndrade/mapeamento-soro-antivenom-pe
```

Esse comando criará uma cópia do projeto na sua máquina.

## 2.2. Requisitos de Sistema

Para executar o sistema localmente, é necessário ter o *Python* instalado, a partir da versão 3.9. Você pode baixar a versão mais recente do *Python* no site oficial (<https://www.python.org/downloads/>). Durante a instalação, certifique-se de marcar a opção **Add Python to PATH** para facilitar o uso no terminal.

Verifique se o *Python* foi instalado corretamente executando o comando **python --version**. A saída deve mostrar a versão instalada, que deve ser 3.9 ou superior.

## 2.3. Instalação de Dependências

O projeto utiliza várias bibliotecas e pacotes *Python*, que estão listados no arquivo **requirements.txt** presente no repositório clonado. No diretório do projeto, execute o seguinte comando para instalar as dependências:

```
pip install -r requirements.txt
```

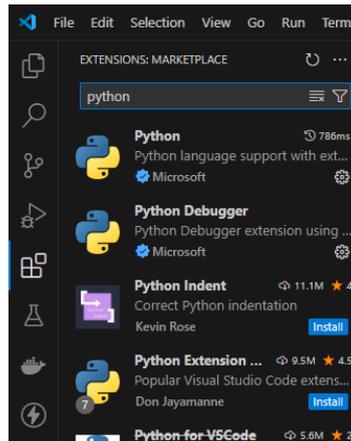
Esse comando instalará todas as bibliotecas necessárias para o funcionamento do sistema, como streamlit, folium, pandas, entre outras. O arquivo requirements.txt já contém todas essas dependências, garantindo que o ambiente será configurado de acordo com o necessário para rodar o sistema.

## 2.4. Configuração da IDE

Para desenvolver e editar o código, recomenda-se o uso de uma IDE (Ambiente de Desenvolvimento Integrado). A IDE mais recomendada para este projeto é o *Visual Studio Code* (VSCode), devido à sua compatibilidade com *Python* e suas ferramentas úteis para

desenvolvimento. Você pode baixar o VSCode em: <https://code.visualstudio.com/>. Após a instalação, abra o projeto no VSCode da seguinte maneira:

No VSCode, clique em **File > Open Folder** e selecione a pasta onde o repositório foi clonado. Certifique-se de que a extensão *Python* está instalada no VSCode. Caso contrário, você pode instalar a extensão buscando por "*Python*" na aba de extensões:



## 2.5. Execução Local da Aplicação

Depois de clonar o repositório e instalar as dependências, você já está pronto para rodar a aplicação localmente. No terminal, dentro do diretório do projeto, execute o seguinte comando:

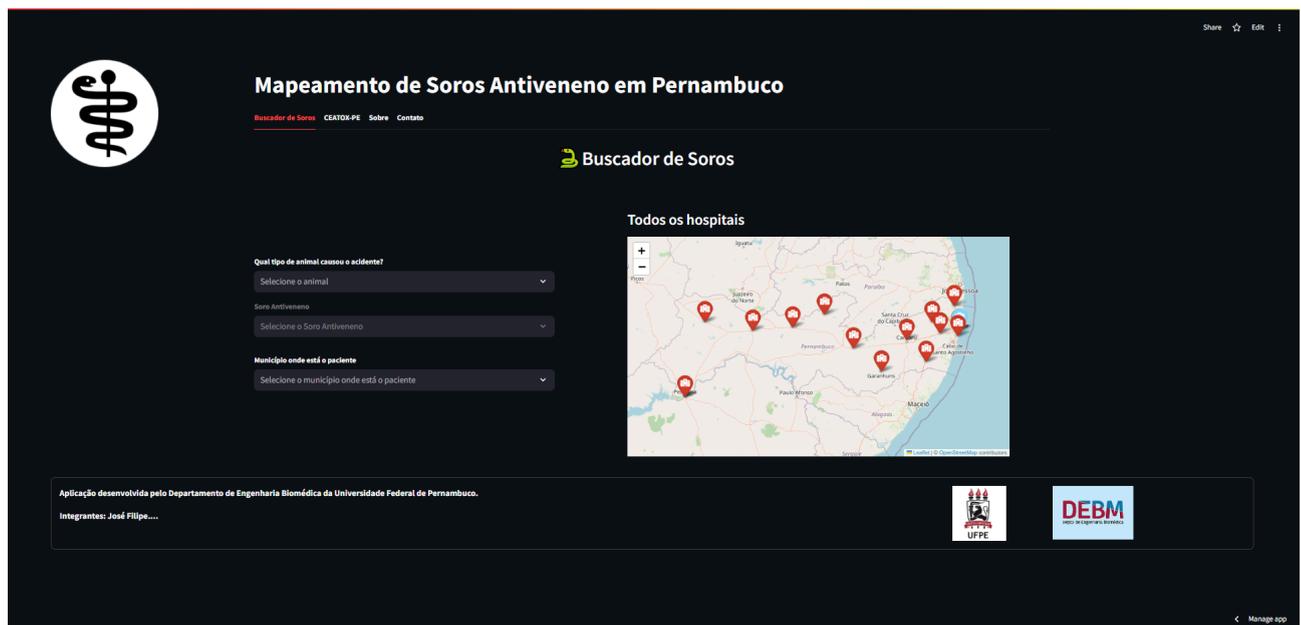
```
PS C:\Users\Filipe\Documents\Projetos\TCC_Projetos\Antivenom> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.16:8501
```

Este comando irá iniciar o servidor local da aplicação, e você poderá acessá-la através do navegador no endereço <http://localhost:8501>.

Página principal do sistema que deverá ser exibida:



## 2.6. Considerações Adicionais

Gerenciamento de Bibliotecas: Recomenda-se o uso de um ambiente virtual (*virtual environment*) para isolar as dependências da aplicação e evitar conflitos com outras bibliotecas instaladas globalmente no seu sistema.

## 3. Visão Geral do Código-Fonte

Este capítulo detalha o funcionamento do código-fonte da aplicação. O código consiste em uma arquitetura monolítica e foi implementado no ambiente de desenvolvimento VSCode, e o controle de versão foi feito utilizando GitHub. Abaixo, são descritas as etapas de funcionamento do código.

### 3.1. Carregamento de Bibliotecas e Arquivos Externos

A primeira etapa do código consiste no carregamento das bibliotecas e na leitura dos arquivos externos, os *dataframes*. O arquivo *requirements.txt* foi utilizado para especificar as dependências do projeto, como *pandas*, *streamlit*, *folium* e outras. Este arquivo foi gerado para facilitar a instalação dos pacotes necessários. No código, é necessário fazer as importações:

```
import pandas as pd
import streamlit as st
import streamlit_js_eval
import folium
from streamlit_folium import st_folium, folium_static
```

**pandas:** Para manipulação de dados em formato tabular (CSV).

**streamlit:** Para criação da interface do usuário.

**streamlit\_js\_eval:** Utilizado para executar scripts em JavaScript (embora não tenha uso visível no código atual).

**folium:** Para a geração de mapas interativos.

**streamlit\_folium:** Integra os mapas Folium à interface do Streamlit.

### 3.2. Streamlit

Alguns métodos do streamlit utilizados no sistema:

Comando	Descrição
<code>st.title()</code>	Exibe o título principal da aplicação.
<code>st.image()</code>	Carrega e exibe imagens na aplicação, como logos ou ícones.
<code>st.markdown()</code>	Renderiza <i>Hypertext Markup Language</i> (HTML) e <i>Markdown</i> para formatação personalizada de texto e interfaces.
<code>st.selectbox()</code>	Cria um menu suspenso, permitindo ao usuário selecionar uma opção de uma lista (ex: animal e soro).
<code>st.container()</code>	Cria um <i>container</i> para agrupar elementos de interface e controles.
<code>st.columns()</code>	Organiza o layout da interface em colunas, facilitando a disposição de diferentes elementos na tela.
<code>st.subheader()</code>	Adiciona subtítulos na aplicação, organizando sessões de maneira hierárquica.
<code>st.write()</code>	Exibe texto e variáveis formatadas na aplicação.
<code>st.tabs()</code>	Cria abas interativas para organizar e exibir diferentes conteúdos em uma mesma página.
<code>st.set_page_config()</code>	Define configurações da página como título, ícone e layout.
<code>folium_static()</code> (via <code>st_folium</code> )	Renderiza mapas interativos Folium diretamente no aplicativo Streamlit.

### 3.3. Configurações Iniciais

O bloco a seguir configura a página inicial da aplicação:

```
# Configurações da página
st.set_page_config(
    page_title="Mapeamento de Soros Antiveneno em Pernambuco",
    page_icon='doc/cobra.png',
    layout="wide",
    initial_sidebar_state='collapsed'
)
```

**st.set\_page\_config:** Define o título da página, o ícone, o layout em tela cheia e o estado inicial da barra lateral como recolhida.

### 3.4. Estrutura *Layout*

O *layout* é dividido em colunas utilizando o recurso **st.columns()** do Streamlit, onde três colunas são criadas para estruturar os elementos da página, proporcionais ao tamanho definido:

```
col1, col2, col3 = st.columns([1,4,1])

col1.image('doc/cobra.png', width=200)
col2.title('Mapeamento de Soros Antiveneno em Pernambuco')
```

**col1:** Exibe o ícone do projeto, uma imagem de uma cobra.

**col2:** Exibe o título "Mapeamento de Soros Antiveneno em Pernambuco" centralizado.

**col3:** Exibe um espaço.

### 3.5. Formatação de Estilos

A personalização de estilos é feita por meio de CSS, utilizando a função **st.markdown()** com o parâmetro **unsafe\_allow\_html=True**, permitindo a inclusão de código HTML na página:



```
st.markdown(  
    """  
    <style>  
    p {  
        font-weight: bold; /* Negrito */  
    }  
    .st-emotion-cache-10trblm.e1nzilvr1 {  
        color: #FF5733;  
    }  
    </style>  
    """,  
    unsafe_allow_html=True,  
)
```

Essa configuração adiciona negrito ao texto e altera a cor de alguns elementos do layout.

### 3.6. Estrutura de Abas

Utilizando **with col2:**, conseguimos fazer modificações apenas na coluna col2. São criadas quatro abas, utilizando o comando **st.tabs: Buscador de Soros, CEATOX-PE, Sobre, e Contato**. A aba principal é o **Buscador de Soros**, e foi a única implementada:

```
with col2:  
    tab1, tab2, tab3, tab4 = st.tabs(["Buscador de Soros", "CEATOX-PE", "Sobre", "Contato"])
```

No GUI:



A seguir, detalharemos a lógica implementada no buscador de soros, que está na aba 1. Foi utilizado o **with tab1:**, para selecionarmos a aba1.

### 3.7. Leitura e Manipulação de Dados

Os dados são lidos de dois arquivos CSV:

```
dados_geral = pd.read_csv('doc/hospital_mais_proximo.csv')  
cidades = pd.read_csv('doc/latitude-longitude-cidades.csv')  
cidades = cidades[cidades['uf'] == 'PE']  
  
lista_mun_distinct = sorted(cidades['municipio'].unique())
```

**hospital\_mais\_proximo.csv**: Contém informações sobre hospitais, soros e suas respectivas distâncias em relação aos municípios, além das informações dos municípios. Este é o arquivo *core* do sistema e está na variável **dados\_geral**. Todas as informações de **hospital\_mais\_proximo.csv**:

```
[ ] resultados_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1295 entries, 0 to 1294
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---                -
0   cidade_origem         1295 non-null   object
1   latitude_origem       1295 non-null   float64
2   longitude_origem      1295 non-null   float64
3   hospital              1295 non-null   object
4   cidade_hospital       1295 non-null   object
5   latitude_hospital     1295 non-null   float64
6   longitude_hospital    1295 non-null   float64
7   telefone              1295 non-null   object
8   endereco              1295 non-null   object
9   distancia_km          1295 non-null   float64
10  soro                  1295 non-null   object
11  animal                1295 non-null   object
dtypes: float64(5), object(7)
memory usage: 121.5+ KB
```

**latitude-longitude-cidades.csv**: Tabela que mapeia latitude e longitude dos municípios do Brasil.

Os dados de cidades são filtrados para retornarem apenas as cidades de Pernambuco. A variável **lista\_mun\_distinc** organiza em ordem alfabética todos os nomes de cidades diferentes presentes no arquivo.

### 3.8. Inputs do Usuário

Ainda dentro da **col2**, é criado 3 novas colunas. Onde na primeira coluna, **col5**, o código cria uma interface para o usuário selecionar o animal responsável pelo acidente, o tipo de soro necessário, e o município onde o paciente se encontra. A interação ocorre através de caixas de seleção (**st.selectbox**)

```
#municipios
col5,espaco2,col14 = st.columns([5,1,7])
with col5:
    st.markdown(
        """
        <br><br><br>
        """
    ,
    unsafe_allow_html=True,
    )
    animal = st.selectbox("Qual tipo de animal causou o acidente?", dados_geral['animal'].unique(), index=None, placeholder="Selecione o animal")
    soro = st.selectbox("Soro Antiveneno", dados_geral[dados_geral['animal']==animal]['soro'].unique(), index=None, placeholder="Selecione o Soro Antiveneno")

    try:
        container = st.container(border=True)
        with container:
            st.write(soro, dicionario_explicacao[soro])
    except:
        st.write("")

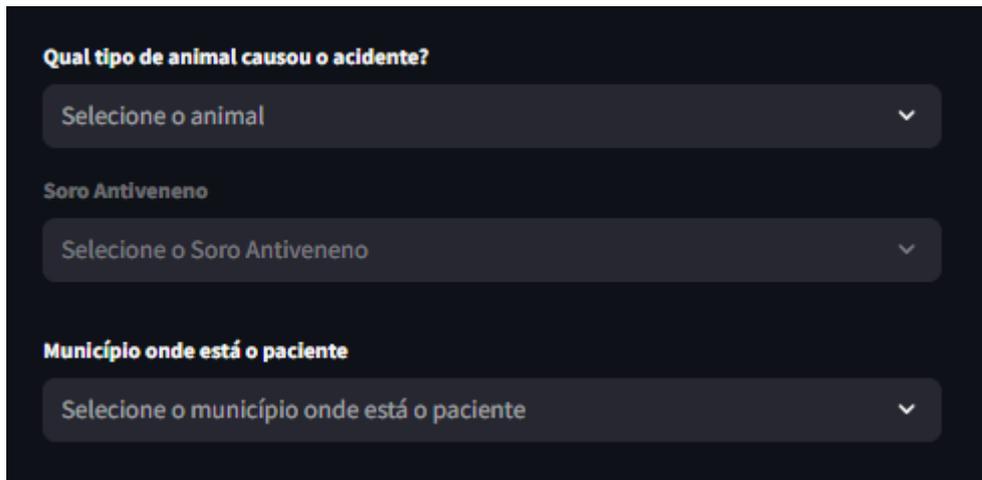
    mun_origem = st.selectbox("Município onde está o paciente", lista_mun_distinct, index=None, placeholder="Selecione o município onde está o paciente")
```

**animal:** Exibe uma lista de animais: Serpente(cobra), Escorpião e Aranha.

**soro:** Mostra os soros disponíveis com base no animal selecionado.

**mun\_origem:** Lista os municípios de Pernambuco para o usuário selecionar a origem.

No GUI:



Usando a caixa de seleção:



Qual tipo de animal causou o acidente?

Selecione o animal

Serpente (cobra)

Escorpião

Aranha

Município onde está o paciente

Selecione o município onde está o paciente

### 3.9. Exibição de Mapas Interativos

Após a seleção dos inputs, o código faz uma filtragem dos dados para determinar o hospital mais próximo:

```
#Filtro destino
filtro = (dados_geral['soro'] == soro)&(dados_geral['cidade_origem'] == mun_origem)
municipio_origem = dados_geral[filtro]
municipio_origem['Legenda'] = 'Origem'

#municipio_origem = municipio_origem.reset_index(drop=True)
mun_destino = municipio_origem.dropna()['cidade_hospital'].values[0]

filtro_destino = (dados_geral['soro'] == soro)&(dados_geral['cidade_origem'] == mun_destino)
municipio_destino = dados_geral[filtro_destino].dropna()
municipio_destino['Legenda'] = 'Destino'
```

O código começa com a filtragem dos dados para identificar a cidade de origem do paciente e a cidade onde está o hospital mais próximo com o soro requerido. Em **filtro**, filtra o *DataFrame* **dados\_geral** para encontrar as informações do município de origem com base no soro selecionado e na cidade de origem escolhida.

Em **municipio\_origem**, os **dados\_geral** recebem o filtro para retornar a linha exata com o município de origem, além de ser adicionado uma nova coluna chamada **Legenda** e adicionando o dado **Origem**.

Em seguida, uma segunda filtragem é realizada para capturar os detalhes da cidade de destino (onde está o hospital), e adiciona uma coluna **Legenda** com o dado **Destino**.

### 3.10. Renderização do Mapa Interativo

Depois de obter os dados de origem e destino, o mapa é gerado utilizando a biblioteca **Folium**. O mapa centraliza nas coordenadas médias entre as cidades de origem e destino:

```
latitude_media = (municipio_origem['latitude_origem'].values[0] + municipio_destino['latitude_hospital'].values[0])/2
longitude_media = (municipio_origem['longitude_origem'].values[0] + municipio_destino['longitude_hospital'].values[0])/2

mapa = folium.Map([latitude_media, longitude_media], zoom_start=9.6)
```

Onde **latitude\_media** e **longitude\_media** calculam a média das latitudes para centralizar o mapa entre os dois pontos (origem e destino). Então, **mapa** inicia o mapa com as coordenadas médias calculadas e um nível de zoom apropriado para mostrar ambos os pontos.

### 3.11. Adicionando Marcadores ao Mapa

Dois marcadores são adicionados ao mapa para representar a cidade de origem e o hospital de destino:

```
folium.Marker(
    location= [municipio_origem['latitude_origem'].values, municipio_origem['longitude_origem'].values],
    tooltip="Origem",
    popup="Você está aqui",
    icon=folium.Icon(color="green", icon="house", prefix="fa"),
).add_to(mapa)

folium.Marker(
    location= [municipio_destino['latitude_hospital'].values, municipio_destino['longitude_hospital'].values],
    tooltip="Destino",
    popup=f"O soro está aqui, na cidade de {municipio_destino['cidade_hospital'].values[0]}, {municipio_destino['hospital'].values[0]}",
    icon=folium.Icon(color="red", icon="hospital", prefix="fa"),
).add_to(mapa)
```

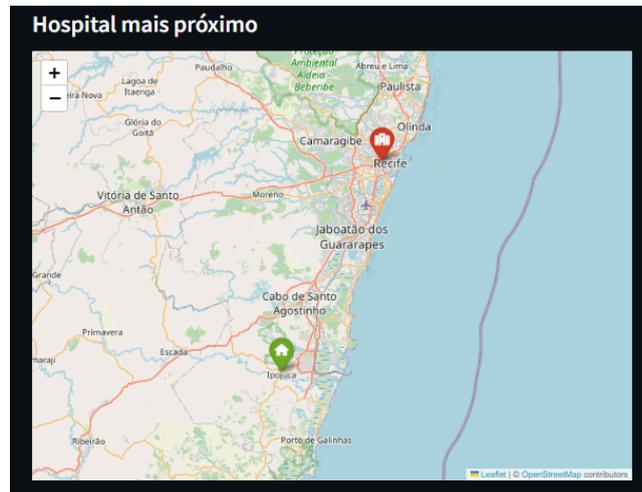
O primeiro adiciona um marcador na localização da cidade de origem com um ícone de "casa" (verde), indicando "Você está aqui". O segundo adiciona um marcador na localização do hospital de destino, com um ícone de "hospital" (vermelho) e uma mensagem **popup** detalhando o local e o hospital onde o soro está disponível.

O mapa então, é exibido com tamanho definido (710x500 pixels), na **col4** dentro da **col2** e com título Hospital mais próximo:

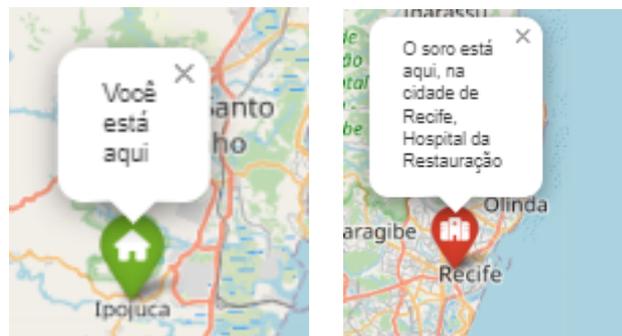


```
with col4:  
    st.subheader('Hospital mais próximo')  
    st_data = folium_static(mapa, width=710, height=500)
```

No GUI:



Popups:



### 3.12. Exibição e Detalhes do Hospital

Além do mapa, detalhes sobre o hospital (endereço, telefone e distância) são coletados do dataframe e exibidos em uma área abaixo dos filtros selectbox na **col5**:



```
with col5:  
    mun_destino = municipio_origem.dropna()['cidade_hospital'].values[0]  
    distancia = municipio_origem.dropna()['distancia_km'].values[0]  
    local = municipio_origem.dropna()['hospital'].values[0]  
    endereco = municipio_origem.dropna()['endereco'].values[0]  
    telefone = municipio_origem.dropna()['telefone'].values[0]  
    container_respostas = st.container(border=True)  
    with container_respostas:  
        st.write(f'Município onde está o soro mais próximo: **{mun_destino}**')  
        st.write(f'Local: **{local}**')  
        st.write(f'Endereço: **{endereco}**')  
        st.write(f'Telefone: **{telefone}**')  
        st.write(f'Distância: **{distancia} km**')  
        st.write(f'**ATENÇÃO**: ligue para o local para fazer a confirmação da disponibilidade do soro.')
```

No sistema:

```
Município onde está o soro mais próximo: Recife  
Local: Hospital da Restauração  
Endereço: Avenida Gov. Agamenon Magalhães s/n - Derby  
Telefone: 0800-7226001/(81) 3181-6453  
Distância: 41.626 km  
ATENÇÃO: ligue para o local para fazer a confirmação da disponibilidade do soro.
```

### 3.13. Default do Mapa

Quando os selectbox não são utilizados, ou seja, quando o usuário não faz nenhuma seleção, o sistema exibe todos os hospitais disponíveis no mapa:

```
except:
with col4:
if soro:
filtro = (dados_geral['soro'] == soro)&(dados_geral['animal'] == animal)
dados_mapa_vazio = dados_geral[filtro]

elif animal:
filtro = (dados_geral['animal'] == animal)
dados_mapa_vazio = dados_geral[filtro]

else:
dados_mapa_vazio = dados_geral.copy()

pontos = dados_mapa_vazio.drop_duplicates(['hospital'])

mapa_vazio = folium.Map([-8.3831638, -37.7753284], zoom_start=7)

for latitude, longitude, hospital, endereco in zip(pontos['latitude_hospital'], pontos['longitude_hospital'], pontos['hospital'], pontos['endereco']):
folium.Marker(
location= [latitude, longitude],
tooltip=hospital,
popup=endereco,
icon=folium.Icon(color="red", icon="hospital",prefix="fa"),
).add_to(mapa_vazio)
```

No sistema:



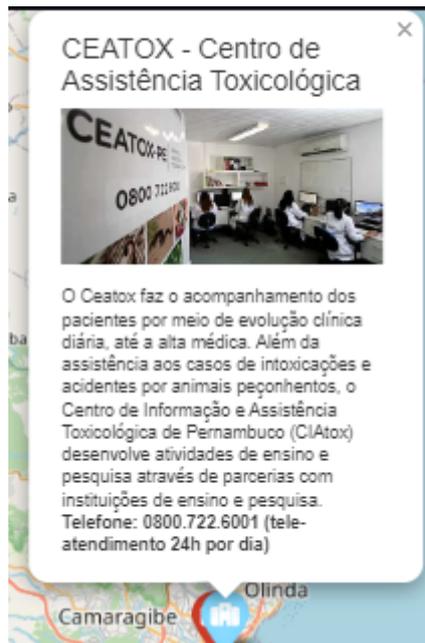
Ainda foi adicionado com marcador azul, a identificação do CIATox-PE, e uma breve descrição sobre ele:

```
ceatox_html = """
<h3>CEATOX - Centro de Assistência Toxicológica</h3>

<p> Ceatox faz o acompanhamento dos pacientes por meio de evolução clínica diária, até a alta médica. Além da assistência aos casos de intoxicações e acidentes por animais peçonhentos, o Centro de Informação e Assistência Toxicológica de Pernambuco (CIATox) desenvolve atividades de ensino e pesquisa através de parcerias com instituições de ensino e pesquisa.
<b>Telefone: 0800.722.6001 (tele-atendimento 24h por dia)</b></p>
"""

folium.Marker(
location= [-8.053771365450142, -34.89005840177061],
tooltip="CEATOX - PE",
popup=ceatox_html,
icon=folium.Icon(color="lightblue",icon="hospital",prefix="fa"),
).add_to(mapa_vazio)
```

No sistema:



### 3.14. Créditos

Ao final da página, uma área de créditos é exibida, informando sobre os desenvolvedores do projeto e exibindo logos institucionais, através do **st.container**:

```
creditos = st.container(border=True)
with creditos:
    col6,espaco3,col7,col8,espaco4 = st.columns([7,2,1,1,1])
    with col6:
        st.write('Aplicação desenvolvida pelo Departamento de Engenharia Biomédica da Universidade Federal de Pernambuco.')
        st.write('Integrantes: José Filipe...')
    with col7:
        st.image('https://upload.wikimedia.org/wikipedia/commons/a/ae/Logo-ufpe-2-2.jpg',width=100)
    with col8:
        st.image('doc/deptEngBio.png', width=150)
```

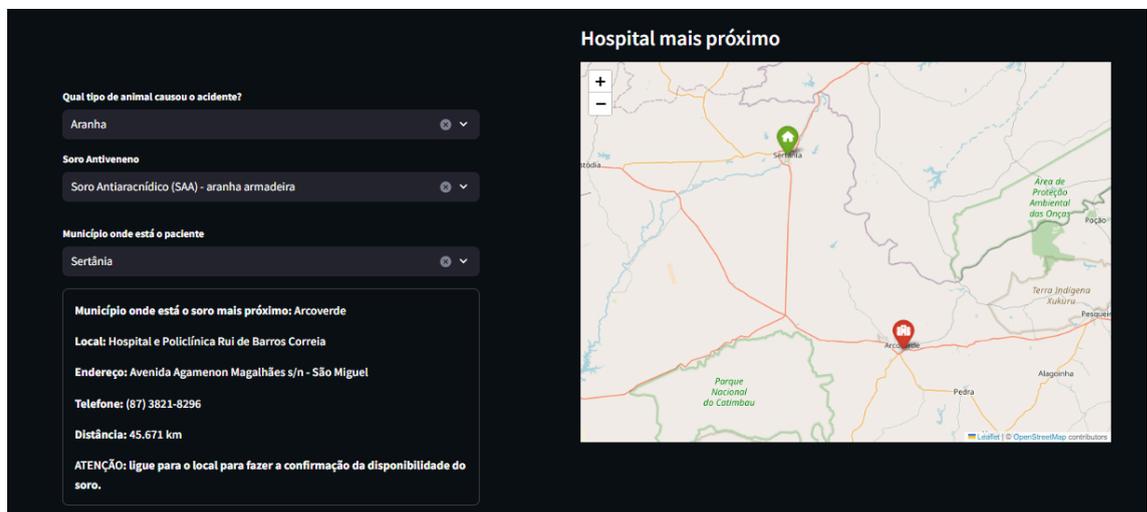
No sistema:



## 4. Caso de Uso: Função Principal do Sistema

João, residente da zona rural de Sertânia - PE, foi picado por uma aranha enquanto trabalhava no campo. Sabendo da gravidade da picada, e identificando que se trata de uma aranha armadeira, João lembrou que o hospital mais próximo que oferecia

soroterapia ficava em Afogados da Ingazeira - PE, uma cidade próxima. Assim, ao utilizar o sistema de mapeamento, ele inseriu as informações necessárias: selecionou a aranha armadeira no **selectbox** de animal, escolheu o tipo de soro adequado e informou sua cidade de origem, Sertânia. No entanto, o sistema rapidamente processou os dados e indicou que o hospital de Afogados da Ingazeira não disponibilizava o soro para aranha armadeira. Ao invés disso, a cidade mais próxima com o tratamento disponível era Arcoverde - PE. O sistema forneceu o endereço e telefone do hospital em Arcoverde, o que permitiu a João organizar seu transporte e receber o tratamento necessário a tempo, evitando complicações graves.



#### 4.1. Descrição Técnica do Caso de Uso

Neste cenário, o usuário interage com o sistema para buscar rapidamente o hospital mais próximo que oferece o tratamento adequado para acidentes com animais peçonhentos. A interface é simples e eficiente, permitindo ao usuário selecionar o tipo de animal causador do acidente, o soro necessário e sua cidade de origem.

- **Frontend (Streamlit):** A interface do usuário exibe componentes como selectboxes para escolher o animal e o tipo de soro. Após a seleção e confirmação da cidade de origem, o sistema realiza uma busca automática e exibe o hospital mais próximo que dispõe do soro, através de um mapa interativo, criado com a biblioteca Folium.
- **Backend (Python, Pandas):** No backend, os dados são processados utilizando um dataframe com informações geográficas dos hospitais de Pernambuco. Com base

nas seleções do usuário, o sistema filtra os hospitais que possuem o soro necessário e calcula a distância entre a cidade de origem do usuário e os hospitais. O hospital mais próximo é então selecionado e as informações relevantes são extraídas para exibição.

- **Visualização (Folium):** Um marcador verde no mapa indica a localização atual do usuário, enquanto um marcador vermelho destaca o hospital mais próximo com o soro disponível. O sistema também exibe detalhes como o nome do hospital, endereço e telefone ao lado do mapa, garantindo que o usuário tenha fácil acesso às informações necessárias para planejar seu deslocamento.

## 5. Testes

Este capítulo tem como objetivo identificar defeitos no sistema, validar suas funções, verificar se os requisitos foram implementados corretamente e avaliar a qualidade geral do software. Para garantir que o sistema de mapeamento de soros antiveneno atenda às expectativas, foram realizados diferentes tipos de testes em um quadro de testes.

### 5.1. Quadro de Testes

O quadro apresenta os casos de testes com o nome do teste, objetivo, o procedimento e o resultado esperado. O sistema foi testado e o quadro preenchido por Fábio, membro do grupo de extensão do projeto. Para o resultado obtido, foi adicionado “passou” ou “falhou”. A análise dos resultados foi discutida posteriormente.

Teste	Objetivo	Procedimento	Resultado Esperado	Resultado Obtido
<b>Funcionalidade de Filtro: Animal</b>	Verificar se o filtro de tipo de animal funciona corretamente.	Selecionar diferentes tipos de animais.	O mapa deverá ser atualizado.	Passou ▾
<b>Funcionalidade de Filtro: Soro</b>	Testar a funcionalidade de filtragem de soro após selecionar o animal.	Selecionar um animal e testar todas as combinações de soro disponíveis.	O sistema deve permitir a escolha de soro apenas após seleção do animal; O mapa deverá ser atualizado.	Passou ▾

<b>Funcionalidade de Filtro: Cidade</b>	Verificar se o filtro de cidade funciona corretamente.	Verificar a lista de cidades e validar se estão em ordem alfabética.	A lista de cidades deve estar ordenada. O mapa deverá ser atualizado.	Passou ▾
<b>Renderização de Mapa (Desktop)</b>	Verificar se o mapa é renderizado corretamente no Desktop.	Acessar a aplicação em um navegador desktop e verificar a visualização correta do mapa.	O mapa deve ser exibido sem falhas no computador.	Passou ▾
<b>Renderização de Mapa (Mobile)</b>	Verificar se o mapa é renderizado corretamente no celular.	Acessar a aplicação em um navegador mobile e verificar a visualização correta do mapa.	O mapa deve ser exibido sem falhas em dispositivos móveis.	Passou ▾
<b>Marcações no Mapa</b>	Verificar se as marcações de hospitais e cidades estão corretas no mapa.	Selecionar qualquer cidade e verificar se os hospitais próximos aparecem no mapa com as marcações adequadas.	As marcações devem ser precisas e aparecer corretamente no mapa.	Passou ▾
<b>Distância Cidade-Hospital (Filtro 1)</b>	Verificar se a distância entre uma cidade e o hospital mais próximo está correta (Filtro 1).	Selecionar os filtros e comparar a distância retornada com a do Google Maps.	A distância deve ser compatível com o valor obtido no Google Maps.	Passou ▾
<b>Distância Cidade-Hospital (Filtro 2)</b>	Verificar se a distância entre uma cidade e o hospital mais próximo está correta (Filtro 2).	Selecionar outra combinação de cidade e hospital e comparar a distância com o Google Maps.	A distância deve ser compatível com o valor obtido no Google Maps.	Passou ▾
<b>Distância Cidade-Hospital (Filtro 3)</b>	Verificar se a distância entre uma cidade e o hospital mais próximo está correta (Filtro 3).	Selecionar uma terceira combinação de cidade e hospital e comparar a distância com o Google Maps.	A distância deve ser compatível com o valor obtido no Google Maps.	Passou ▾
<b>Carregamento de Dados</b>	Testar a exibição dos dados do hospital ao lado do mapa.	Selecionar qualquer cidade e verificar se as informações do hospital mais próximo são exibidas corretamente.	Informações do hospital devem aparecer de forma completa e correta.	Passou ▾

<b>Testes de Performance (Desktop)</b>	Avaliar o tempo de resposta ao carregar o mapa em dispositivos desktop.	Medir o tempo que a aplicação leva para processar e carregar o mapa no notebook.	O mapa deve ser carregado em menos de 3 segundos.	Passou ▾
<b>Testes de Performance (Mobile)</b>	Avaliar o tempo de resposta ao carregar o mapa em dispositivos móveis.	Medir o tempo que a aplicação leva para processar e carregar o mapa no celular.	O mapa deve ser carregado em menos de 3 segundos.	Passou ▾

## 6. Avaliação do Sistema

Com base no resultado dos testes realizados foi sugerida a alteração do formato da distância exibida (de 3 para 2 casas decimais), para uma melhor entendimento do valor da distância. O tempo de resposta ao carregar o mapa em dispositivos desktops foi 1 segundo. Já em dispositivos móveis, durou 2 segundos, o que é aceitável. Os filtros e a exibição dos dados funcionaram perfeitamente.

## 7. Referências

ANDRADE, J. F. S. **Proposta de sistema de suporte para aperfeiçoamento do atendimento a casos de envenenamento por animais peçonhentos em Pernambuco**. TCC – Departamento de Engenharia Biomédica, Universidade Federal de Pernambuco, Recife. 2024

## 8. Aprovação da Revisão

Aprovações		
Participante	Assinatura	Data
Prof. <sup>a</sup> Dr. <sup>a</sup> Cristine Martins Gomes de Gusmão		