

FM

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

Felipe Amadori Machado

**IMPLEMENTAÇÃO E INTEGRAÇÃO DE UMA ARQUITETURA DE
SOFTWARE PARA SIMULAÇÕES DE PROPAGAÇÃO DE DOENÇAS EM
ANIMAIS**

Santa Maria, RS
2024

Felipe Amadori Machado

**IMPLEMENTAÇÃO E INTEGRAÇÃO DE UMA ARQUITETURA DE SOFTWARE PARA
SIMULAÇÕES DE PROPAGAÇÃO DE DOENÇAS EM ANIMAIS**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação, Área de Concentração em Ciências Exatas e da Terra, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Sistemas de Informação**.

Orientador: Prof. Alencar Machado

Coorientador: Prof. Celio Trois

Santa Maria, RS
2024

Felipe Amadori Machado

**IMPLEMENTAÇÃO E INTEGRAÇÃO DE UMA ARQUITETURA DE SOFTWARE PARA
SIMULAÇÕES DE PROPAGAÇÃO DE DOENÇAS EM ANIMAIS**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação, Área de Concentração em Ciências Exatas e da Terra, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Sistemas de Informação**.

Aprovado em 22 de julho de 2024:

Alencar Machado, Dr. (UFSM)
(Presidente/Orientador)

Antonio Marcos de Oliveira Candia, Me. (UFSM)

Giliane Bernardi, Dra. (UFSM)

Santa Maria, RS
2024

RESUMO

IMPLEMENTAÇÃO E INTEGRAÇÃO DE UMA ARQUITETURA DE SOFTWARE PARA SIMULAÇÕES DE PROPAGAÇÃO DE DOENÇAS EM ANIMAIS

AUTOR: Felipe Amadori Machado

Orientador: Alencar Machado

Coorientador: Celio Trois

Os sistemas de previsão de propagação de doenças têm se destacado, especialmente após a pandemia de COVID-19. Eles são ferramentas valiosas para controlar a disseminação de doenças infecciosas e otimizar a alocação de recursos para pesquisa e desenvolvimento. O modelo MHASpread é um sistema inteligente que utiliza dados reais para simular a propagação de doenças em animais como gado, suínos e pequenos ruminantes. Esse sistema permite que pesquisadores testem diferentes estratégias de controle dessas doenças, fornecendo informações valiosas para intervenções de saúde pública. No entanto, a versão atual do sistema é acessível apenas a usuários com conhecimentos em programação e requer recursos computacionais substanciais, representando uma barreira significativa para a adoção por profissionais veterinários. Este trabalho visa melhorar a escalabilidade e acessibilidade do modelo, permitindo que epidemiologistas e profissionais de sanidade animal insiram dados de forma eficiente, executem simulações e visualizem resultados de maneira intuitiva. A nova arquitetura de software suporta o paralelismo de requisições de usuários e agregará informações em gráficos e mapas. Integrada à Plataforma de Defesa Sanitária Animal (PDSA-RS), a arquitetura proposta eliminará a necessidade de configuração de ambiente pelos usuários, oferecendo uma ferramenta eficaz para simular a disseminação de doenças e avaliar o impacto das medidas de controle implementadas. A arquitetura proposta foi avaliada e testada durante um treinamento de espalhamento de doenças, realizado na Secretaria da Agricultura, Pecuária, Produção Sustentável e Irrigação (SEAPI), com a participação de cerca de vinte médicos veterinários estaduais. Ao final do treinamento, concluiu-se que o trabalho conseguiu cumprir com seu objetivo de suportar múltiplos usuários executando em paralelo, além de contribuir como uma ferramenta para auxiliar na tomada de decisão dos usuários envolvidos.

Palavras-chave: Sistemas Inteligentes, Sistemas de Apoio à Decisão, Controle de Doenças

ABSTRACT

IMPLEMENTATION AND INTEGRATION OF A SOFTWARE ARCHITECTURE FOR DISEASE SPREAD SIMULATIONS IN ANIMALS

AUTHOR: Felipe Amadori Machado

ADVISOR: Alencar Machado

CO-ADVISOR: Celio Trois

Disease propagation prediction systems have gained prominence, especially following the COVID-19 pandemic. They are valuable tools for controlling the spread of infectious diseases and optimizing resource allocation for research and development. The MHASpread model is an intelligent system that uses real data to simulate the spread of diseases in animals such as cattle, swine, and small ruminants. This system allows researchers to test different control strategies for these diseases, providing valuable information for public health interventions. However, the current version of the system is accessible only to users with programming knowledge and requires substantial computational resources, representing a significant barrier to adoption by veterinary professionals. This work aims to improve the scalability and accessibility of the model, allowing epidemiologists and animal health professionals to efficiently input data, run simulations, and visualize results intuitively. The new software architecture will support parallel user requests and aggregate information into charts and maps. Integrated with the Animal Health Defense Platform (PDSA-RS), the proposed architecture will eliminate the need for users to configure their environment, offering an effective tool to simulate disease spread and assess the impact of implemented control measures. The proposed architecture was evaluated and tested during a disease spread training session held at the State Secretariat of Agriculture, Livestock, Sustainable Production and Irrigation (SEAPI), with the participation of about twenty state veterinarians. At the end of the training, it was concluded that the work achieved its goal of supporting multiple users executing in parallel, as well as contributing as a tool to assist in the decision-making of the involved users.

Keywords: Intelligent Systems, Decision-Support Systems, Diseases Control

LISTA DE FIGURAS

Figura 1 – Gráfico de movimentação de animais entre fazendas.	14
Figura 2 – Simulação de propagação de doenças.	15
Figura 3 – Arquitetura PDSA-RS	18
Figura 4 – Containers Docker vs. Containers Linux Tradicionais	21
Figura 5 – Diagrama de arquitetura do sistema	24
Figura 6 – Fluxo de comunicação entre o cliente e a API do modelo para criar uma análise.	25
Figura 7 – Tela para criação de novo cenário	28
Figura 8 – Aplicativo mostrando um gráfico de linha ilustrando o quantidade de fazen- das infectadas ao longo do tempo.	28
Figura 9 – Aplicativo mostrando um mapa de propagação de doenças.	29
Figura 10 – Recurso de ações de controle da aplicação.....	30

LISTA DE ABREVIATURAS

API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LAB	Laboratório
LGPD	Lei Geral de Proteção de Dados Pessoais
LSP	Language Server Protocol
MAPA	Ministério Da Agricultura Pecuária e Abastecimento
PDSA-RS	Plataforma de defesa sanitária animal do Rio Grande do Sul
RT	Responsável Técnico
SVE	Serviço veterinário Estadual
STDIN	Standard Input
STDOUT	Standard Output
TCP	Transmission Control Protocol

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVO GERAL	10
1.2	OBJETIVOS ESPECÍFICOS	11
1.2.1	Aplicar os Conceitos de REST e Microserviços	11
1.2.2	Encapsular o Modelo de Simulação de Doenças em um Contêiner Docker	11
1.2.3	Implementar e Integrar a Arquitetura de Software Proposta	11
1.3	ESTRUTURA DO TEXTO	12
2	REFERENCIAL TEÓRICO	13
2.1	SISTEMA DE PROPAGAÇÃO E CONTROLE DE DOENÇAS	13
2.1.1	Representação de Movimentação Animal	14
2.1.2	Dinâmica de Espalhamento de Doença	14
2.1.3	Ações de Controle	15
2.1.4	Vacinação	16
2.2	PLATAFORMA DE DEFESA SANITÁRIA ANIMAL (PDSA-RS)	16
2.2.1	Módulo de Gerenciamento para Transporte de Animal Morto (DTAM)	17
2.2.2	Módulo de Gerenciamento de Granjas de Suínos Certificados (GRSC)	17
2.2.3	Módulo de Gerenciamento de Registro de Granjas de Aves Comerciais	17
2.2.4	Módulo de Gerenciamento e Simulação de Foco de Doenças	18
2.2.5	Arquitetura da PDSA-RS	18
2.3	REST API E MICROSERVIÇOS	19
2.3.1	REST	19
2.3.2	Microserviços	20
2.4	DOCKER	20
2.4.1	Containers Docker vs. Containers Linux Tradicionais	20
2.4.2	Vantagens dos Containers Docker	21
3	TRABALHOS RELACIONADOS	22
4	IMPLEMENTAÇÃO DA ARQUITETURA	23
4.1	API CLIENTE	23
4.2	MODELO ESTOCÁSTICO	23
4.3	API DO MODELO	24
4.4	FLUXO DE COMUNICAÇÃO	25
5	ESTUDO DE CASO	27
5.1	CENÁRIO FICTÍCIO	27
5.2	USO DA APLICAÇÃO	27
5.3	RELATÓRIO E PREVISÕES	28
5.4	MAPAS E LINHA DO TEMPO	29

5.5	AÇÕES DE CONTROLE	29
5.6	PROCESSAMENTO INTERNO	30
6	RESULTADOS E DISCUSSÃO	31
6.1	TREINAMENTO SEAPI OUTUBRO/2023	31
7	CONCLUSÃO	33
7.1	TRABALHOS FUTUROS	33
	REFERÊNCIAS BIBLIOGRÁFICAS	35

1 INTRODUÇÃO

Os sistemas de previsão de propagação de doenças têm se destacado cada vez mais, especialmente devido à pandemia de COVID-19. Eles oferecem informações valiosas para controlar a disseminação de doenças infecciosas e alocar recursos para pesquisa e desenvolvimento. Métodos de aprendizado de máquina, como redes neurais artificiais e redes LSTM (redes de memória de longo e curto prazo), têm sido amplamente utilizados na previsão de séries temporais, incluindo o número diário de casos, óbitos e casos recuperados de doenças como a COVID-19. Um exemplo de aplicação de Deep Learning com LSTM foi demonstrado em um estudo recente (CHANDRA; JAIN; CHAUHAN, 2022). Esses modelos auxiliam profissionais de saúde pública e formuladores de políticas a tomar decisões informadas com base em análises e previsões precisas, contribuindo assim para um controle mais eficaz de surtos de doenças.

Paralelamente, os modelos matemáticos de propagação de doenças são ferramentas fundamentais na epidemiologia, sendo utilizados há bastante tempo para compreender a dinâmica dos surtos. Esses modelos, baseados em equações e algoritmos, capturam a complexidade dos surtos ao considerar fatores como a transmissibilidade da doença, a resposta imunológica da população e as intervenções de controle. Em conjunto com dados de prevalência, esses modelos são essenciais para estudar a dinâmica, analisar as causas e identificar fatores-chave dos surtos de diversas doenças.

Exemplos notáveis incluem o desenvolvimento de modelos para COVID-19 por Sha HE (HE et al., 2020), SARS por Zhou e Ma (ZHOU; MA; BRAUER, 2004), e febre aftosa por Ferguson, Keeling e Kao (FERGUSON; DONNELLY; ANDERSON, 2001; KEELING et al., 2001; KAO, 2003). Esses modelos matemáticos têm sido importantes na epidemiologia, mesmo antes da ascensão das técnicas de inteligência artificial, e continuam sendo uma ferramenta essencial na área.

A implementação de sistemas inteligentes desempenha um papel fundamental no controle e prevenção de doenças animais, impactando significativamente a saúde pública. Isso engloba uma série de intervenções estratégicas, incluindo vacinação animal, controle de vetores, protocolos rígidos de higiene e vigilância de doenças em animais. A globalização e o extenso comércio de animais e seus produtos têm aumentado a disseminação global de doenças zoonóticas, destacando a necessidade urgente de estratégias robustas de prevenção e controle para evitar epidemias potenciais. Além disso, a dinâmica operacional dentro das fazendas introduz riscos inerentes para a transmissão de doenças. Fatores como a segregação de grupos de animais, padrões de tráfego humano e veicular, bem como o layout espacial das fazendas, podem influenciar significativamente a transmissão potencial de doenças. Por exemplo, a proximidade de lotes de animais pode facilitar a propagação de doenças, enquanto medidas eficazes de isolamento podem ajudar a conter

surtos. O movimento humano e veicular nas fazendas pode introduzir patógenos, destacando a importância de medidas de biossegurança. Além disso, a organização espacial das fazendas pode impactar a disseminação de patógenos, destacando a necessidade de medidas de controle adaptadas para mitigar os riscos de transmissão de doenças (GALVIS et al., 2022).

No estudo realizado por (CARDENAS et al., 2021), foi proposto um sistema inteligente que utiliza dados reais para simular a propagação de doenças em animais, como gado, suínos e pequenos ruminantes. Esse sistema permite que os pesquisadores testem diferentes estratégias de controle para essas doenças. Após a análise, o sistema disponibiliza o modelo em uma interface de usuário para ação adicional. Essa abordagem enfrenta os desafios de controlar a propagação de doenças infecciosas e oferece dados valiosos para intervenções de saúde pública.

O sistema de controle de doenças serve como uma ferramenta valiosa, no entanto, seu design atual atende principalmente a usuários com experiência em programação, que podem navegar na instalação de dependências e interagir com ele por meio de uma interface de programação. Isso é evidente em sua distribuição como uma biblioteca para download, o que exige um certo nível de proficiência técnica. Além disso, a análise do sistema exige recursos computacionais substanciais, com testes indicando a necessidade de mais de 4GB de RAM para que uma única análise seja executada efetivamente. Essa alta barreira técnica representa um desafio significativo para a adoção generalizada, especialmente entre profissionais veterinários que podem não ter habilidades de programação. Acessar e utilizar o sistema não apenas exige conhecimento sobre doenças de animais de criação e epidemiologia, mas também proficiência na linguagem de programação do sistema.

1.1 OBJETIVO GERAL

Este projeto de pesquisa visa superar essas limitações desenvolvendo uma arquitetura de software capaz de escalar a execução do modelo para suportar o paralelismo de requisições de usuários e agregar as informações de cada execução em gráficos e mapas intuitivos. Isso permitirá que profissionais com conhecimento em sanidade animal e epidemiologistas tomem decisões informadas sobre esforços de prevenção, intervenção e resposta, ajustando parâmetros, executando simulações e visualizando resultados de forma mais acessível.

A proposta inclui a implementação do modelo em um contêiner Docker para facilitar sua execução e escalabilidade, permitindo uma resposta rápida e intervenção generalizada em situações que exigem implantação ágil, como surtos de doenças. O desenvolvimento de sistemas de previsão de propagação de doenças representa uma abordagem

promissora para controlar a disseminação de doenças infecciosas e alocar recursos para pesquisa e desenvolvimento. Compreender a arquitetura de implantação desses modelos nesse contexto é fundamental para melhorar o controle da propagação de doenças e mitigar seu impacto na sociedade.

Além disso, a proposta abrange ainda a criação de uma interface de usuário amigável e a integração eficiente entre a API, a UI e o modelo de simulação. Espera-se que essa abordagem melhore a capacidade de simulação e a precisão das informações, contribuindo para decisões mais eficazes no controle da febre aftosa e mitigação do impacto das doenças infecciosas. A arquitetura será integrada na Plataforma de Defesa Sanitária Animal (PDSA-RS) (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020; DESCOVI et al., 2021), ampliando seu papel na criação de ferramentas de gestão e apoio à tomada de decisão em sanidade animal.

1.2 OBJETIVOS ESPECÍFICOS

Esta seção contextualiza os objetivos específicos, são metas que devem ser alcançadas para atingir um objetivo. Esses objetivos detalham as ações específicas que devem ser realizadas para alcançar o objetivo geral fornecendo uma direção mais clara para atingir o objetivo.

1.2.1 Aplicar os Conceitos de REST e Microserviços

Compreender os conceitos de arquitetura REST e microserviços, explorando seus princípios e benefícios e compreendendo sua aplicabilidade no contexto do projeto.

1.2.2 Encapsular o Modelo de Simulação de Doenças em um Contêiner Docker

Compreender o uso do Docker para encapsular o modelo de simulação de doenças de forma otimizada, visando garantir uma escalabilidade eficiente, reprodução facilitada e integração apropriada do modelo.

1.2.3 Implementar e Integrar a Arquitetura de Software Proposta

Implementar e integrar a arquitetura de software proposta à PDSA-RS. Isso envolve analisar a arquitetura de software existente da PDSA-RS, identificar pontos de integração

com a nova arquitetura proposta, projetar e implementar interfaces de comunicação entre os sistemas, e testar e validar a integração para garantir sua eficiência e evitar impactos negativos na PDSA-RS.

1.3 ESTRUTURA DO TEXTO

O trabalho está organizado da seguinte forma. O Capítulo 2 apresenta o contexto, fornecendo uma breve introdução sobre o modelo estocástico desenvolvido por Cardenas (CARDENAS; LOPES; MACHADO, 2022), que aborda a propagação de doenças e o sistema de controle, além de apresentar a plataforma na qual será integrado, conhecida como PDSA-RS (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020; DESCOVI et al., 2021), e os conceitos necessários para a implementação e integração da arquitetura proposta. O Capítulo 3 descreve os trabalhos relacionados, indicando os pontos que foram concluídos com sucesso e os que não foram. O Capítulo 4 descreve o sistema proposto para integrar o modelo e fornecer uma solução para o problema descrito. O Capítulo 5 descreve a avaliação do sistema, apresentando um cenário de avaliação para validar a solução. O Capítulo 6 apresenta os resultados. O Capítulo 7 apresenta as conclusões deste trabalho.

2 REFERENCIAL TEÓRICO

Este capítulo oferece uma visão abrangente do Sistema de Propagação e Controle de Doenças utilizado neste estudo, destacando os desafios atuais que esta pesquisa pretende resolver. Além disso, aborda a Plataforma de Defesa Sanitária Animal (PDSA-RS), na qual o sistema será integrado, enfatizando como essa integração, baseada em Docker e RestAPIs/Microserviços, potencializa a eficácia das ferramentas de controle e gestão de doenças.

2.1 SISTEMA DE PROPAGAÇÃO E CONTROLE DE DOENÇAS

O sistema de propagação e controle de doenças, introduzido por (CARDENAS et al., 2021; CARDENAS; LOPES; MACHADO, 2022; CARDENAS et al., 2022), foi projetado para demonstrar o potencial de disseminação de doenças infecciosas entre populações de animais de criação, considerando a transmissão via movimento dos animais. Este modelo incorpora simulações de várias ações de controle, como a prevenção de entrada ou saída de animais infectados nas fazendas (por exemplo, abate, isolamento de animais), melhoria das práticas higiênicas ou implementação de programas de vacinação.

Esse sistema oferece aos usuários a oportunidade de prever o impacto de surtos infecciosos e desenvolver estratégias proativas para mitigar riscos. Ao alavancar a modelagem computacional, o sistema fornece informações sobre a dinâmica da propagação de doenças em diferentes cenários, permitindo que as partes interessadas tomem decisões informadas sobre prevenção, intervenção e esforços de resposta. A funcionalidade central do sistema reside na simulação da disseminação de agentes infecciosos entre populações de animais de criação, levando em consideração fatores como padrões de movimento dos animais, medidas de biossegurança e condições ambientais. Os usuários podem inserir dados do mundo real relacionados à demografia animal, localizações geográficas e ocorrências históricas de doenças, permitindo-lhes gerar modelos personalizados adaptados às necessidades regionais específicas.

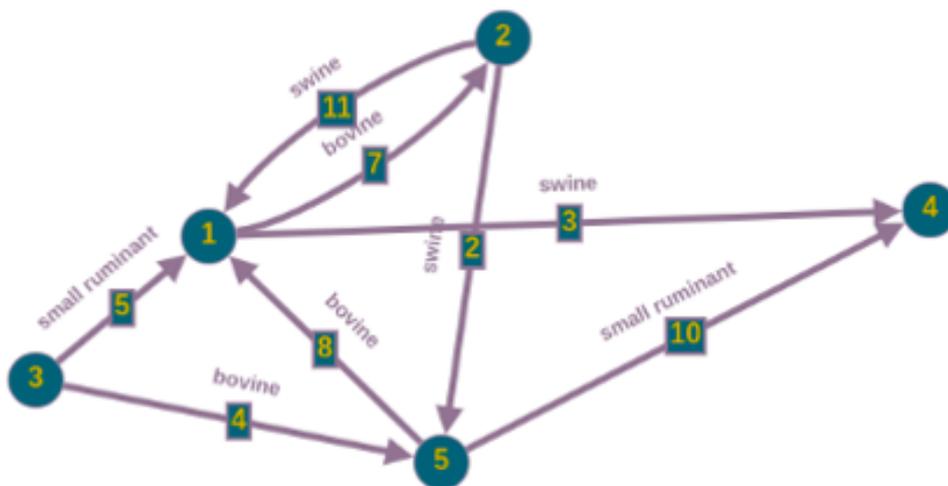
Uma vez criado um modelo, os usuários podem explorar vários cenários "e se", testando planos alternativos de ações de controle contra os resultados simulados. Por exemplo, eles podem examinar as consequências da imposição de zonas de quarentena ao redor de fazendas afetadas, avaliar a eficácia de campanhas de vacinação direcionadas ou analisar as compensações custo-benefício associadas a diferentes opções de controle. Ao fornecer uma estrutura abrangente de suporte à decisão, o sistema capacita formuladores de políticas, veterinários e produtores a antecipar e responder de forma eficaz às ameaças emergentes representadas por doenças infecciosas em populações de animais de criação.

Este trabalho foca em utilizar este modelo para fornecer uma ferramenta para veterinários tomarem ações e explorarem cenários, auxiliando-os em situações críticas de tomada de decisão.

2.1.1 Representação de Movimentação Animal

O sistema utiliza métodos de Análise de Redes Sociais (SNA) (CARDENAS et al., 2022) para caracterizar os padrões de comércio animal, e o total de animais movidos entre fazendas é representado no sistema como um grafo direcionado, onde cada fazenda é representada como um nó e os movimentos entre fazendas são representados como arestas. Cada aresta conecta uma origem específica a um destino específico, mantendo também o tipo e o número de animais sendo movidos. Um exemplo de representação gráfica de movimento é mostrado na Figura 2.

Figura 1 – Gráfico de movimentação de animais entre fazendas.



Fonte: Cardenas, Lopes e Machado (2022)

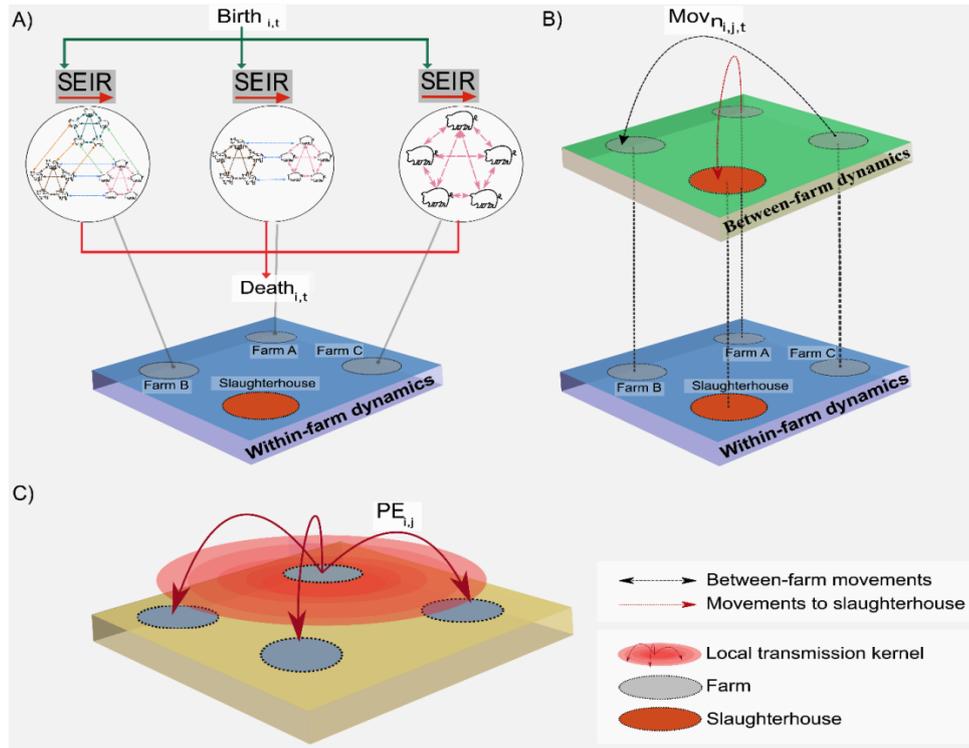
O sistema mantém um conjunto de dados de eventos com informações sobre a origem, destino, tipo e número de animais, que é usado para construir o grafo de movimento. Os movimentos entre fazendas de diferentes espécies consideram, portanto, uma rede de contato real multi-hospedeiro com dados de movimento coletados.

2.1.2 Dinâmica de Espalhamento de Doença

O sistema aplica um algoritmo de simulação estocástica (SSA) (CARDENAS et al., 2022) para simular a disseminação da doença, bem como a dinâmica vital (nascimento e morte) dos animais dentro de cada fazenda. O sistema incorpora dinâmicas dentro da

fazenda e entre fazendas através de um modelo suscetível-infectante usando dados de movimento animal temporais explicitamente com uma taxa de contato efetiva mais alta para garantir uma transmissão eficiente da doença durante as simulações. A Figura 3 ilustra a transição de estados durante as dinâmicas dentro e entre fazendas no modelo.

Figura 2 – Simulação de propagação de doenças.



Fonte: Cardenas, Lopes e Machado (2022)

2.1.3 Ações de Controle

Após uma proliferação inicial oculta onde animais em fazendas selecionadas contraem a doença, o controle e a contenção expedita do patógeno são fundamentais para eliminar a infecção e promover a recuperação. Medidas como quarentena e regulação dos movimentos dos animais servem para proteger a saúde animal, impedindo a transmissão da doença para populações não contaminadas (ROTH, 2007). Essas ações são executadas dentro de regiões de controle designadas, cujas fronteiras são determinadas por parâmetros do usuário. No entanto, apesar da eficácia dessas medidas, ainda há uma necessidade urgente de simplificar o processo de aplicação para os veterinários. Atualmente, a implementação desses controles requer conhecimento especializado e competência técnica, limitando o alcance e a utilidade dessas medidas entre a comunidade veterinária mais ampla.

2.1.4 Vacinação

Dentro do sistema, o processo de vacinação é simulado ao longo da disseminação da doença. Após a vacinação, os animais transitam dos compartimentos Suscetível-Exposto-Infectante-Recuperado (SEIR) (CARDENAS et al., 2022) para um compartimento distinto de Vacinação (V). Essa segregação nos permite rastrear precisamente o impacto da vacinação na disseminação da doença dentro da população.

A transição dos animais para o compartimento V é influenciada por dois fatores chave: a eficácia da vacina e uma taxa de conversão diária definida pelo usuário para ações de controle. A eficácia da vacina determina sua capacidade de conferir imunidade, afetando a proporção de animais vacinados protegidos da infecção. Enquanto isso, a taxa de conversão diária dita o ritmo no qual os indivíduos suscetíveis são vacinados, influenciando assim a taxa de acumulação dentro do compartimento V ao longo do tempo.

2.2 PLATAFORMA DE DEFESA SANITÁRIA ANIMAL (PDSA-RS)

Constituem objetivos da Plataforma de Defesa Sanitária Animal do Rio Grande do Sul PDSA-RS, proporcionar a organização sistêmica das atividades, garantir o desenvolvimento e proteção de informações sanitárias das atividades de produção, harmonizando e integrando todas as políticas governamentais, a cooperação técnico-científica, com as participações de instituições, organizações e setores públicos e privados, de forma a garantir o desenvolvimento sanitário no Estado do Rio Grande do Sul (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020).

A integração entre o setor produtivo, academia, diagnóstico animal e serviço veterinário oficial, para o desenvolvimento de software, caracteriza um modelo de gestão público-privada validado pela Organização Mundial de Saúde Animal e o Serviço Veterinário Oficial Brasileiro, uma vez que promove sinergia de recursos humanos e financeiros, alto nível de engajamento e metas compartilhadas, proporcionando agilidade na tomada de decisões por cada agente (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020).

Buscando consolidar esses princípios, a Plataforma de Defesa Sanitária Animal do Rio Grande do Sul PDSA-RS está disponível em sítio web e aplicativo aos técnicos do SVO, responsáveis técnicos de agroindústrias no RS, e laboratórios credenciados. Esta plataforma contempla inserção e gestão de dados com diferentes agentes participantes, representados por usuários com níveis distintos de acesso ao sistema por intermédio de Portal Responsável Técnico (RT), Portal Serviço Veterinário Estadual (SVE), Portal Serviço Veterinário Federal (SVO-MAPA), Portal Laboratórios Credenciados (Laboratório) e Portal do Cidadão, que oferece serviços a atores da cadeia produtiva que não necessitam de

cadastro prévio na PDSA-RS para acessar funcionalidades (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020).

2.2.1 Módulo de Gerenciamento para Transporte de Animal Morto (DTAM)

Este módulo tem como objetivo gerenciar todas as informações geradas a partir da emissão do Documento de Transporte de Animal Morto (DTAM), buscando assim o controle efetivo da origem e destino desses animais e seus resíduos, bem como o rastreamento das rotas percorridas no deslocamento desses animais, tornando-se de suma importância para o controle sanitário sob o aspecto de disseminação de doenças (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020).

2.2.2 Módulo de Gerenciamento de Granjas de Suídeos Certificados (GRSC)

Este módulo tem como objetivo gerenciar todas as informações que envolvem a emissão do certificado de granjas de GRSC e SUIFEM. A Secretaria da Agricultura monitora a prevalência de doenças na produção de suínos em diferentes níveis da cadeia, tanto nas granjas de reprodutores quanto nas granjas de produção de corte. Granjas de Reprodutores de Suídeos Certificados (GRSC) estão no topo da cadeia de produção e mantêm a linhagem genética e produção de animais para as granjas de corte. Nessas granjas, monitoramento contínuo de doenças, tais como Peste Suína Clássica, Brucelose, Tuberculose, Doença de Aujeszky e Sarna, são realizados pelas empresas e supervisionados pelo serviço veterinário oficial para controle que afetam a pirâmide de produção. Em relação às granjas de corte comercial, mensalmente os responsáveis técnicos desses estabelecimentos devem preencher e enviar ao serviço veterinário oficial a ficha epidemiológica mensal (SUIFEM) para controle de doenças, tais como Botulismo, Cirovirose, Coccidiose, Doença de Glasser, entre outras. O monitoramento e controle permanente dessas doenças controla a dispersão e prevalências de doenças conhecidas, desta forma sendo possível determinar o estado epidemiológico dos lotes em produção (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020).

2.2.3 Módulo de Gerenciamento de Registro de Granjas de Aves Comerciais

Este módulo tem como objetivo gerenciar todas as informações que envolvem o registro de estabelecimentos avícolas de corte do Rio Grande do Sul, os quais estão condicionados ao atendimento de requisitos estruturais, operacionais de biossegurança e de

avaliação de riscos. Dentre as diretrizes de atuação do Programa Nacional de Sanidade Avícola estão os procedimentos para registro, fiscalização e o controle sanitário dos estabelecimentos avícolas comerciais e de ensino ou pesquisa, os quais são definidos por meio da Instrução Normativa nº 56, de 4 de dezembro de 2007. Compete aos Órgãos Estaduais de Saúde Animal a adoção dos procedimentos para registro dos estabelecimentos avícolas comerciais e de ensino ou pesquisa (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020).

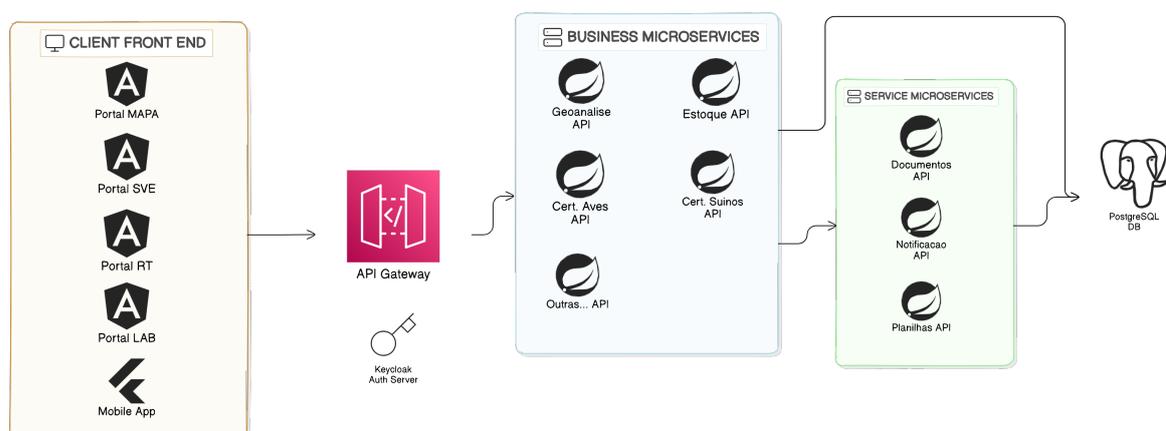
2.2.4 Módulo de Gerenciamento e Simulação de Foco de Doenças

Este módulo tem como objetivo gerar informações para tomada de decisão em resposta a focos de doenças. Esse módulo é composto por dois grupos de funcionalidades: (a) resposta ao foco, que busca gerar um ambiente para gerar informações para a tomada de decisão frente a um foco de doença existente e (b) abarcar a estrutura em sistema web para rodar modelos de simulação de espalhamento de doenças, com ambiente para suportar as rodadas da simulação e ferramenta de visualização (Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020).

2.2.5 Arquitetura da PDSA-RS

A Figura 4 ilustra a arquitetura da plataforma já existente:

Figura 3 – Arquitetura PDSA-RS



Fonte: Elaborado pelo autor

Observa-se que a arquitetura já existente aplica os conceitos de microserviços, contendo múltiplas APIs, cada uma desempenhando um papel específico. Nosso trabalho será

integrado à API responsável pelo controle do Módulo de Gerenciamento e Simulação de Foco de Doenças, denominada Geoanálise API. Considerando que a arquitetura atual da PDSA utiliza os conceitos de microserviços e REST, decidiu-se manter a mesma abordagem para o desenvolvimento da nossa arquitetura, facilitando assim a integração. Os conceitos serão explicados nas próximas sessões.

2.3 REST API E MICROSERVIÇOS

Uma API (Interface de Programação de Aplicações) é um conjunto de definições e protocolos para construir e integrar software. Funciona como um contrato entre um provedor de informações e um usuário, especificando o que o consumidor deve fornecer (a chamada) e o que o produtor deve retornar (a resposta) (Red Hat, Inc., 2024a). Em outras palavras, uma API ajuda a comunicar o que você deseja a um sistema para que ele entenda e atenda à sua solicitação. Ela atua como um mediador entre usuários ou clientes e os recursos ou serviços web que eles desejam acessar, mantendo segurança, controle e autenticação (Red Hat, Inc., 2024a).

2.3.1 REST

REST (Representational State Transfer) é um conjunto de restrições arquitetônicas, não um protocolo ou padrão. APIs RESTful transferem uma representação do estado do recurso para o solicitante ou endpoint via HTTP, em formatos como JSON, HTML, XML, Python, PHP ou texto simples, sendo o JSON o mais popular (Red Hat, Inc., 2024a).

Para ser considerada RESTful, uma API deve atender aos seguintes critérios:

- Arquitetura cliente-servidor, com solicitações gerenciadas através de HTTP.
- Comunicação sem estado, onde nenhuma informação do cliente é armazenada entre solicitações.
- Dados cacheáveis para otimizar interações cliente-servidor.
- Interface uniforme para transferência de informações em formato padrão.
- Sistema em camadas, organizando os servidores em hierarquias invisíveis ao cliente.
- Código sob demanda (opcional), permitindo que o servidor envie código executável ao cliente.

2.3.2 Microserviços

Microserviços são uma abordagem arquitetônica onde uma aplicação é estruturada como um conjunto de pequenos serviços independentes, cada um executando um processo específico e comunicando-se entre si através de APIs. Cada microserviço é desenvolvido, implementado e escalado de forma independente, o que proporciona maior flexibilidade e agilidade no desenvolvimento de software (LEWIS; FOWLER, 2014).

Vantagens dos microserviços incluem:

- **Escalabilidade:** Cada serviço pode ser escalado individualmente conforme a demanda.
- **Resiliência:** Falhas em um serviço não necessariamente afetam outros serviços.
- **Flexibilidade Tecnológica:** Diferentes serviços podem ser desenvolvidos usando diferentes tecnologias que sejam mais adequadas para cada tarefa específica.
- **Desenvolvimento Independente:** Equipes podem trabalhar simultaneamente em diferentes serviços sem causar dependências ou conflitos.

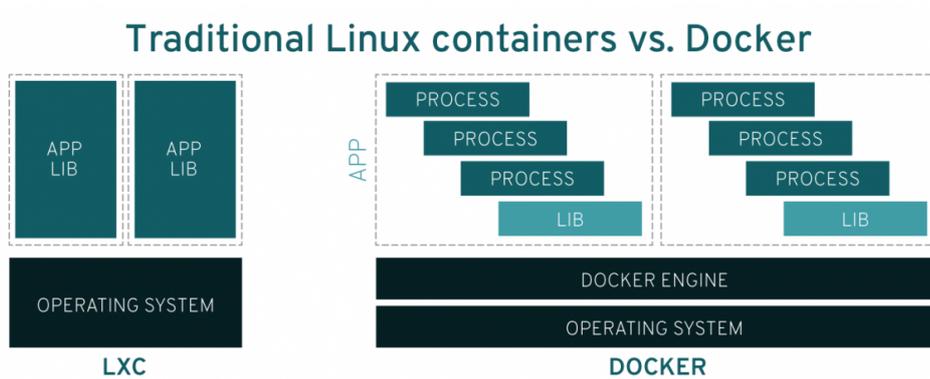
2.4 DOCKER

Docker é uma tecnologia de containerização que permite a criação e o uso de containers Linux de forma modular e leve. Docker utiliza o kernel do Linux e suas funcionalidades, como cGroups e namespaces, para segregar processos e executá-los de maneira independente. Essa segregação permite uma melhor utilização da infraestrutura e manutenção da segurança. Docker adota um modelo de implantação baseado em imagem, facilitando o compartilhamento de aplicativos e suas dependências em diferentes ambientes. Além disso, automatiza a implantação de aplicações dentro de containers (Red Hat, Inc., 2024b).

2.4.1 Containers Docker vs. Containers Linux Tradicionais

Embora o Docker tenha se originado da tecnologia LXC (Linux Containers), ele evoluiu para oferecer uma experiência mais amigável para usuários e desenvolvedores. Diferentemente dos containers Linux tradicionais, que gerenciam vários processos como uma unidade, Docker promove a segregação de aplicações em processos separados, oferecendo ferramentas para criação, construção, envio e controle de versão de containers (Red Hat, Inc., 2024b).

Figura 4 – Containers Docker vs. Containers Linux Tradicionais



Fonte: Red Hat, Inc. (2024b)

2.4.2 Vantagens dos Containers Docker

- **Modularidade:** Permite a atualização ou reparo de partes de uma aplicação sem interrompê-la totalmente, facilitando uma abordagem baseada em microsserviços.
- **Camadas e Controle de Versão:** Docker cria novas camadas sempre que há mudanças na imagem, facilitando o controle de versões e a reutilização de camadas para novos containers.
- **Reversão:** É possível reverter para versões anteriores de uma imagem, o que suporta práticas de desenvolvimento ágil e CI/CD.
- **Implantação Rápida:** A criação e a implantação de containers Docker ocorrem em segundos, sem a necessidade de inicializar o sistema operacional, o que reduz drasticamente o tempo de implantação.

3 TRABALHOS RELACIONADOS

O *Australian Animal Disease Spread Model* (BRADHURST et al., 2015), AADIS, é uma ferramenta de suporte à decisão que auxilia na formulação de políticas e estratégias de resposta para doenças animais emergenciais. O modelo integra abordagens matemáticas, baseadas em agentes, de redes e de autômatos celulares para simular a incursão, detecção, vigilância, controle e prova de liberdade de doenças animais emergenciais. AADIS é projetado para fornecer dados valiosos sobre a disseminação e o manejo de doenças animais, auxiliando no desenvolvimento de estratégias de resposta eficazes.

O *Animal Disease Spread Model* (SCHOENBAUM et al., 2024), (ADSM), é um modelo estocástico, espacialmente explícito e compartimental que simula a propagação de doenças animais altamente contagiosas entre rebanhos. Desenvolvido pelo USDA-APHIS-VS-CEAH, o ADSM é projetado para avaliar diferentes estratégias de controle, fornecer recomendações sobre alocação de recursos e estimar os impactos econômicos das opções de controle de doenças. O modelo é baseado no *North American Disease Spread Model* (NAADSM) e tem sido usado para simular surtos de doenças como a febre aftosa nos Estados Unidos, focando em estratégias de controle de abate, abate e vacinação. O ADSM é construído em Django e Python e apresentado como um aplicativo web, permitindo que os usuários interajam com o modelo por meio de uma interface amigável. O modelo usa multithreading para maximizar a utilização do sistema e fornece saídas detalhadas e saídas complementares para análises adicionais.

Os dois trabalhos apresentados nesta seção fornecem um modelo para a propagação de doenças e uma interface de usuário para os usuários. Os principais problemas com ambos os trabalhos são que eles fornecem um aplicativo a ser executado na máquina do usuário, sendo um aplicativo focado em desktop, onde um usuário final não apenas precisaria configurar seu ambiente para poder executar o aplicativo, ter um bom conhecimento da doença e do modelo para poder usar o aplicativo, e também ter uma máquina boa o suficiente para executar a simulação. Essas questões tornam o aplicativo não amigável ao usuário e inadequado para nosso caso de uso. Com esse problema também vem o fato de que os modelos são de código fechado, então não podemos usar o código para fazer nossas próprias alterações e fornecer um aplicativo para o usuário, tornando ambos os trabalhos inadequados para nosso uso.

4 IMPLEMENTAÇÃO DA ARQUITETURA

A solução proposta para analisar a propagação de doenças em fazendas é um sistema web que combina uma interface de usuário (UI), uma interface de programação de aplicativos (API) e o modelo proposto por (CARDENAS et al., 2021) implementado em Python, para fornecer uma plataforma abrangente e amigável para rastrear e prever a propagação de doenças. O sistema consiste em três componentes principais: uma API cliente que integrará ao sistema, a API do modelo que irá despachar, gerenciar e escalar uma instância do modelo, e o próprio modelo (EBLING et al., 2024).

4.1 API CLIENTE

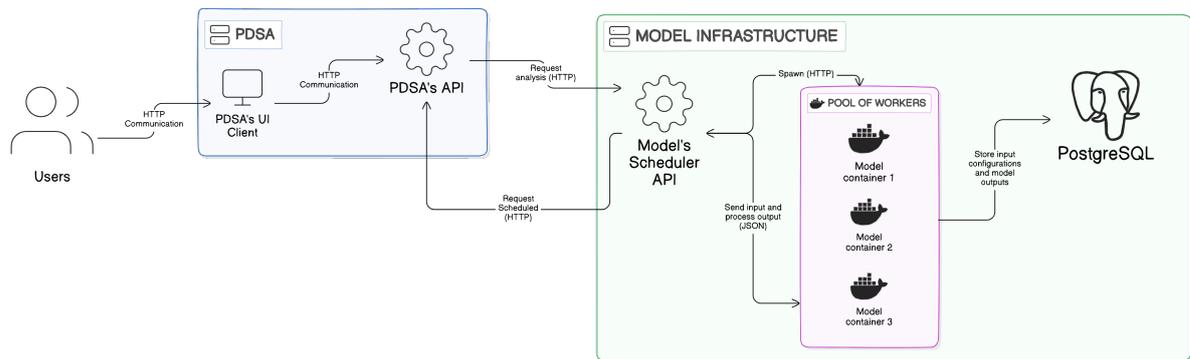
A API cliente é o sistema que possui os dados e deseja integrar o modelo. A API cliente terá acesso a informações sobre fazendas, movimentos e eventos, e poderá usá-las para consultar a API do modelo, a fim de criar e visualizar análises. Além de possuir os dados para análise, a única implementação necessária neste componente é a capacidade de se comunicar com uma API RESTful (EHSAN et al., 2022). No nosso caso, a API cliente é a denominada Geoanálise API, que faz parte da plataforma já existente, a PDSA-RS, apresentada no Capítulo 2.

4.2 MODELO ESTOCÁSTICO

O modelo estocástico foi inicialmente implementado como uma biblioteca em R (CARDENAS et al., 2021), o que dificulta sua acessibilidade para os usuários, pois eles precisam interagir diretamente com o código e configurar as dependências necessárias em suas máquinas pessoais. Para resolver isso, o próximo componente é um wrapper fino em torno do modelo que fornece duas funcionalidades importantes para permitir seu gerenciamento pela API: a capacidade de ser executado como um contêiner Docker (MERKEL et al., 2014) e um protocolo baseado em JSON para comunicação via STDIO. A execução como contêiner Docker é fundamental, pois possibilita a escalabilidade das instâncias do modelo e permite que a API controle seus recursos de forma eficiente. O protocolo baseado em JSON, por sua vez, permite que a API monitore e se comunique com o modelo, obtendo dados em tempo real sobre o progresso das análises. A escolha de usar o transporte via STDIO, em vez de TCP ou outro mecanismo de transporte comum, foi fortemente inspirada pelas implementações do Protocolo de Servidor de Linguagem da

Microsoft (LSP), onde a maioria das implementações utiliza o STUDIO para se comunicar com um editor por meio de um protocolo baseado em JSON para enviar informações sobre um projeto. Com isso, conseguimos ter APIs RESTful se comunicando via protocolo JSON e o modelo se comunicando também por meio de um protocolo JSON, mas utilizando o STUDIO para interação com a API do modelo. Isso é ilustrado na Figura 5, que mostra o tipo de comunicação entre os componentes. Além disso, a PDSA-RS é uma plataforma já existente, enquanto a nova arquitetura é descrita como "Model Infrastructure".

Figura 5 – Diagrama de arquitetura do sistema



Fonte: Elaborado pelo autor

4.3 API DO MODELO

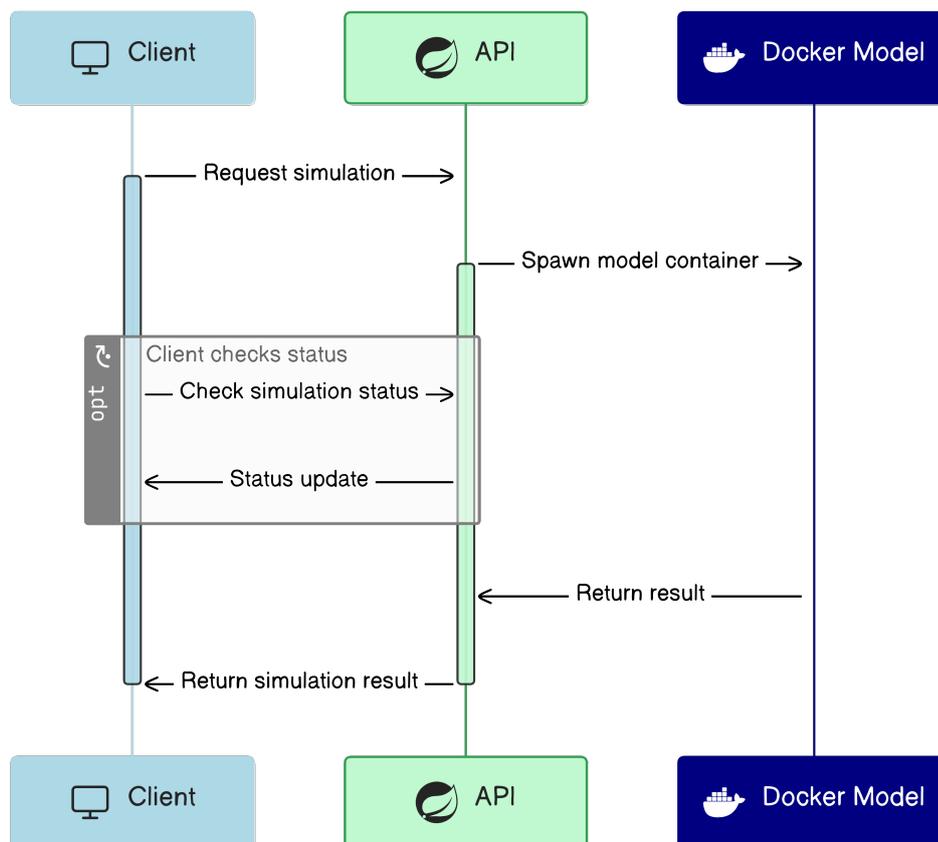
A API do modelo é o ponto de entrada para acessar as funcionalidades do modelo estocástico, sendo responsável por gerar, agendar e gerenciar as solicitações de análise utilizando o modelo. Implementada como uma API RESTful em Java Spring Boot, ela se comunica com a API cliente para o gerenciamento de dados, fornecendo atualizações em tempo real sobre as previsões de propagação de doenças e o impacto das estratégias de intervenção. Sua principal função é fazer a ponte entre as APIs clientes e o modelo de propagação de doenças. Para isso, a API gerencia um pool de workers do modelo, onde, ao receber uma solicitação de análise, o trabalho é despachado para esse pool. Cada worker é executado como um contêiner Docker de forma assíncrona. A API gera e monitora o stream STDOUT do contêiner, analisando suas mensagens de log, que contêm informações em JSON, obtendo assim dados em tempo real sobre o progresso do modelo. Outro papel importante dessa API é armazenar os dados das análises executadas. Cada solicitação de análise despacha uma instância do modelo, e o resultado do contêiner precisa ser armazenado em um banco de dados relacional. A API persiste os dados sobre os modelos em execução, garantindo que todas as informações necessárias estejam disponíveis para

futuras consultas e análises. Posteriormente, esses dados são consultados para fornecer os resultados às APIs clientes. Com isso, a API fornece um modelo escalável para executar os modelos estocásticos, onde o trabalho pode ser despachado para um cluster Docker para execução.

4.4 FLUXO DE COMUNICAÇÃO

Com todos os componentes juntos, uma análise pode ser iniciada a partir da API cliente, onde enviará dados a serem analisados para a API do modelo, juntamente com os parâmetros para executar o modelo. A API do modelo instanciará um contêiner Docker para executar essa análise, monitorando e agregando sua saída. Enquanto isso, o cliente pode consultar a API para obter o progresso em tempo real da análise e, quando concluída, terá acesso total à saída do modelo. Esse fluxo do processo é ilustrado na Figura 6.

Figura 6 – Fluxo de comunicação entre o cliente e a API do modelo para criar uma análise.



Fonte: Elaborado pelo autor

Em conclusão, a solução proposta combina várias tecnologias para tentar fornecer uma solução escalável para a execução de um modelo estocástico. Este sistema oferece

informações valiosas para controlar a propagação de doenças e alocar recursos para pesquisa e desenvolvimento.

5 ESTUDO DE CASO

Para avaliar a viabilidade da abordagem e a implementação desenvolvida, foi desenvolvido um estudo de caso. Este estudo de caso tem como objetivo demonstrar o uso da nossa aplicação em um cenário onde há suspeita de um surto de doença nas instalações. Esta aplicação tem como principal objetivo ajudar veterinários a combater e controlar surtos de forma proativa.

5.1 CENÁRIO FICTÍCIO

Este cenário fictício é baseado no trabalho cotidiano dos veterinários estaduais no Brasil e visa avaliar a eficácia da aplicação em ajudar o veterinário a controlar e identificar outras fazendas que poderiam ser afetadas. Para o estudo de caso, o seguinte cenário foi considerado para descrever a abordagem suportada pela aplicação desenvolvida. Imagine "João", um veterinário oficial do Estado, no Brasil. Como veterinário estadual, uma de suas muitas tarefas é auditar instalações e certificar-se de que estão livres de doenças comuns que podem ser prejudiciais ao ecossistema de produção pecuária. Um dia, enquanto realizava auditorias de rotina, João é notificado de que há sintomas de uma doença em uma fazenda. Sintomas suspeitos em alguns animais e taxas de mortalidade incomuns levantam preocupações. Reconhecendo a urgência da situação, João rapidamente coleta amostras e as envia para testes para confirmar a presença de uma doença contagiosa.

5.2 USO DA APLICAÇÃO

Ao receber os resultados dos testes, João faz login no sistema da PDSA-RS (DESCOVI et al., 2021; Plataforma de Defesa Sanitária Animal do Rio Grande do Sul, 2020) para usar o módulo de controle de doenças para ajudá-lo a planejar suas ações. O sistema fornece uma interface amigável onde ele insere o caso confirmado e os detalhes relevantes, ilustrado na Figura 7. Aproveitando os algoritmos da aplicação, ela rapidamente analisa os dados, considerando fatores como proximidade geográfica, registros de movimentação de animais e condições ambientais.

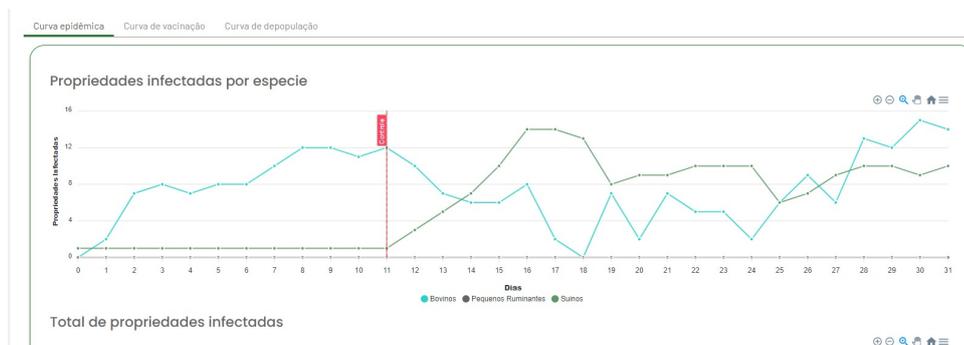
Figura 7 – Tela para criação de novo cenário

Fonte: Imagem do sistema

5.3 RELATÓRIO E PREVISÕES

Em questão de minutos, a aplicação gera um relatório abrangente que descreve o risco potencial e identifica outras instalações com alto risco de serem afetadas pelo surto. Os algoritmos de modelagem preditiva levam em conta vários fatores, incluindo redes de transporte de animais, padrões de vento e dados históricos de propagação de doenças. Isso fornece a João diversas informações, mostrando como essa doença se espalhará, infectando instalações ao redor, inclusive de outras espécies, como ilustrado na Figura 8.

Figura 8 – Aplicativo mostrando um gráfico de linha ilustrando o quantidade de fazendas infectadas ao longo do tempo.

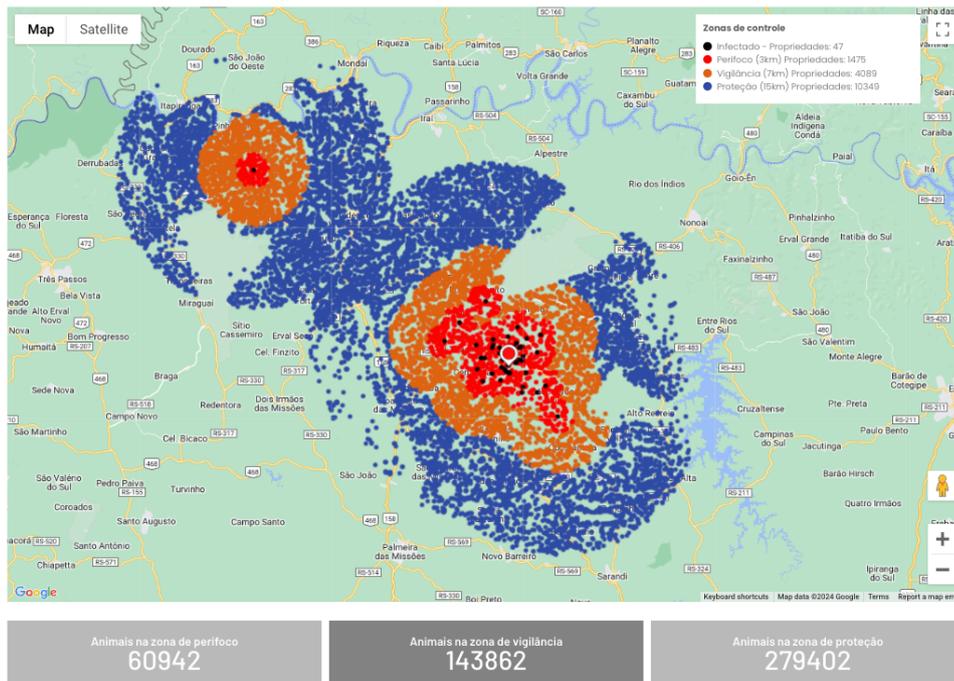


Fonte: Imagem do sistema

5.4 MAPAS E LINHA DO TEMPO

João recebe um mapa destacando as fazendas mais suscetíveis ao contágio, como mostrado na Figura 9. O sistema também fornece uma linha do tempo estimada para possíveis surtos nessas instalações identificadas, permitindo que João priorize e planeje medidas de resposta emergencial de forma eficaz.

Figura 9 – Aplicativo mostrando um mapa de propagação de doenças.



Fonte: Imagem do sistema

5.5 AÇÕES DE CONTROLE

Para planejar suas ações, João usa o recurso de ações de controle da aplicação, onde ele pode aplicar ações a um conjunto de instalações e ver o impacto das ações para controlar o surto da doença. Isso se torna um processo iterativo, onde o veterinário atualizará o sistema com o estado atual da propagação da doença, usará o sistema para ajudar a planejar as próximas ações, executará as ações e reiniciará o processo até que o surto seja contido. A figura 10 ilustra, as medidas de controle que João pode aplicar.

Munido dessas informações vitais, João inicia a comunicação imediata com as fazendas identificadas, informando-as sobre a ameaça potencial e aconselhando medidas de precaução. As capacidades em tempo real do sistema permitem que João monitore a situação conforme ela se desenrola, recebendo atualizações sobre o progresso do surto e tomando decisões informadas para conter sua propagação.

Figura 10 – Recurso de ações de controle da aplicação.

The image shows a web application interface with a green sidebar on the left and a main content area on the right. The sidebar contains the following menu items: Página inicial, Ajuda, GEO ANÁLISE (with sub-items: Análise geral, Análise por foco, Resposta ao foco, Ações no Foco), ESPALHAMENTO DE DOENÇAS (with sub-item: Simulação), and ADMINISTRAÇÃO (with sub-item: Upload de movimentações). The main content area is divided into three sections:

- Restrição de movimentação animal:**
 - Dias de bloqueio: 5
 - Contatos diretos: SIM
 - Zona de perifoco?: NÃO
 - Zona de vigilância?: NÃO
 - Zona de proteção?: SIM
- Depopulação:**
 - Zona de perifo...: SIM
 - Apenas propri...: SIM
 - Límite de proprie...: 10
 - Profissionais disp...: 150
- Vacinação:**
 - Dias para o inici...: 5
 - Dias para consid...: 5
 - Eficiência da vac...: 8000000000
 - Propriedades i...: NÃO
 - Bovinos: SIM
 - Suínos: NÃO
 - Peq. Ruminantes: NÃO
 - Zona d...: NÃO
 - Número...: 0
 - Zona d...: SIM
 - Número...: 15
 - Profissionais disponíveis: 50

Fonte: Imagem do sistema

5.6 PROCESSAMENTO INTERNO

Por trás das cenas, sempre que o usuário cria uma nova análise, o sistema passa pelo seguinte processo para produzir um relatório:

- A PDSA-RS solicita uma nova análise, enviando os dados de instalações, e o período de movimentações e eventos para a API do modelo.
- A API do Scheduler do modelo criará um contêiner Docker, informando os parâmetros e executando a instância do modelo.
- A API agregará a saída do modelo e armazenará os dados finais.
- Enquanto o Scheduler realiza seu trabalho, a PDSA-RS consultará seu status e, quando estiver pronto, solicitará o resultado processado da análise.

6 RESULTADOS E DISCUSSÃO

Este estudo de caso foi bem-sucedido, demonstrando a eficácia da ferramenta desenvolvida para auxiliar veterinários na tomada de decisões críticas durante surtos de doenças. O trabalho consistiu na criação de uma plataforma integrada, utilizando contêineres Docker para facilitar a execução e a escalabilidade do modelo. A plataforma foi projetada para simular a propagação de doenças e desenvolver medidas de controle, com uma interface de usuário intuitiva e uma integração eficiente entre a API, a UI e o modelo de simulação.

Dessa forma, a aplicação se mostra essencial para capacitar veterinários, como João, a responder rapidamente e estrategicamente a surtos de doenças, minimizando o impacto no gado e garantindo a saúde geral do ecossistema agropecuário. Este cenário evidencia a praticidade e a eficácia da aplicação desenvolvida em um ambiente do mundo real, demonstrando seu potencial para revolucionar os esforços de controle de doenças na indústria pecuária.

A obtenção dos dados utilizados no desenvolvimento e validação da aplicação foi possível graças à colaboração dos oficiais veterinários do Rio Grande do Sul. Atualmente, a principal doença que o modelo simula é a febre aftosa, uma preocupação constante para a indústria pecuária. O Rio Grande do Sul é um estado que recebeu o status de zona livre de aftosa sem vacinação, o que ressalta a importância de ferramentas como esta para manter o atual status sanitário. A aplicação não só auxilia na rápida identificação e resposta a potenciais surtos, mas também na manutenção de um rigoroso controle sanitário, essencial para preservar a saúde do rebanho e a segurança alimentar.

Dos dados fornecidos pelo Estado, temos cerca de 353.746 propriedades com seus respectivos saldos de animais, totalizando para esse estudo de caso:

- 12.001.796 Bovinos e Bubalinos
- 6.618.751 Suínos
- 3.146.164 Pequenos ruminantes

Além disso, foi fornecida uma vasta quantidade de dados de movimentação animal, totalizando mais de 1 milhão de movimentações.

6.1 TREINAMENTO SEAPI OUTUBRO/2023

Em outubro de 2023, foi realizado um treinamento sobre febre aftosa na Secretaria da Agricultura, Pecuária, Produção Sustentável e Irrigação (SEAPI), com a participação de

cerca de vinte médicos veterinários estaduais. A maioria dos participantes já tinha acesso à plataforma PDSA-RS, enquanto os demais foram registrados durante o evento.

O treinamento teve duração de três dias. No primeiro dia, um veterinário apresentou uma palestra teórica sobre a febre aftosa, abordando aspectos epidemiológicos da doença. No segundo dia, os participantes foram divididos em grupos, onde cada grupo utilizou a plataforma para simular o espalhamento da doença e desenvolver medidas de controle para o seu cenário específico. No terceiro dia, cada grupo apresentou seu caso, explicando as decisões tomadas e avaliando a eficácia das estratégias implementadas.

Com essa abordagem, a arquitetura proposta demonstrou sua capacidade de facilitar e escalar o modelo, permitindo que os médicos veterinários se concentrassem na análise dos cenários em vez de se preocupar com a configuração do ambiente de execução. Durante a dinâmica, foram realizadas cerca de cem simulações, abrangendo tanto a propagação de doenças quanto as ações de controle. Além disso, o treinamento, que anteriormente durava cinco dias utilizando a arquitetura antiga, onde o modelo era disponibilizado como uma biblioteca em R, foi reduzido para três dias. Isso ocorreu porque eliminamos a necessidade dos usuários configurarem seus computadores pessoais e a interação direta com o código.

Para testar a escalabilidade inicial do modelo, organizamos uma ação em que todos os médicos veterinários presentes executaram as simulações simultaneamente. No primeiro teste de carga, ultrapassamos o limite de armazenamento do servidor, pois os dados de eventos e da população de cada simulação eram salvos como arquivos JSON em um diretório para fins de depuração. Após identificar esse problema, limpamos a pasta de armazenamento e ajustamos o código para não salvar mais esses dados, já que é possível acessá-los sem essa etapa. Com essa correção, realizamos uma segunda rodada do teste de carga, na qual conseguimos criar e executar aproximadamente vinte contêineres simultaneamente com sucesso. Enquanto atendíamos às necessidades dos usuários, notamos que há espaço para aprimorar a interface, tornando-a mais intuitiva e fácil de entender.

7 CONCLUSÃO

Durante o treinamento realizado em outubro de 2023, na Secretaria da Agricultura, Pecuária, Produção Sustentável e Irrigação (SEAPI), com a participação de cerca de vinte médicos veterinários estaduais, a ferramenta foi testada. O treinamento teve duração de três dias, com atividades práticas que incluíram palestras teóricas, simulações em grupo e apresentações de casos. Este ambiente colaborativo permitiu que os veterinários utilizassem a plataforma para simular o espalhamento da febre aftosa e desenvolver estratégias de controle específicas para seus cenários.

Os resultados do treinamento mostraram que a ferramenta atendeu às necessidades dos usuários. Cerca de cem simulações foram realizadas durante o evento, incluindo simulações de espalhamento e ações de controle. Para testar a escalabilidade inicial do modelo, uma ação coordenada foi realizada, onde todos os veterinários presentes executaram simultaneamente cerca de vinte contêineres, demonstrando que a plataforma pode lidar com múltiplas instâncias sem comprometer o desempenho.

A análise das atividades do treinamento revelou que os veterinários foram capazes de se concentrar na análise dos cenários e na tomada de decisões, em vez de se preocupar com a configuração do ambiente para a execução do modelo, essencial para a resposta rápida e eficaz em situações de surtos de doenças.

A arquitetura desenvolvida neste estudo de caso mostrou-se eficaz e escalável, atendendo às necessidades dos veterinários e proporcionando uma ferramenta valiosa para o controle e contenção de surtos de doenças na indústria de produção pecuária. Os resultados positivos do treinamento validam a abordagem adotada e sugerem que a ferramenta pode ser amplamente utilizada em outros contextos e regiões para melhorar a gestão de crises sanitárias na pecuária.

7.1 TRABALHOS FUTUROS

Para trabalhos futuros, será importante focar ainda mais na redução dos custos, tornando a aplicação mais escalável para usuários em geral.

Além disso, outra necessidade identificada é a capacidade de executar múltiplas simulações para um mesmo cenário de forma simultânea. No momento, é possível executar mais de uma simulação ao mesmo tempo, mas apenas para cenários diferentes. Devido à natureza estocástica do modelo, para obter maior precisão nos resultados, é comum executar dezenas ou centenas de simulações para um mesmo cenário e agregar os resultados para obter uma média. Implementar essa funcionalidade permitirá uma análise mais robusta e confiável, proporcionando dados ainda mais precisos para o controle de

doenças.

No modelo antigo, era necessário disponibilizar toda a base de dados de população e eventos para cada usuário durante o treinamento. Com a nova arquitetura, os usuários não têm acesso direto a esses dados. Futuras análises devem avaliar como a nova arquitetura se alinha com os requisitos da LGPD e assegurar a conformidade com as normas de proteção de dados.

REFERÊNCIAS

BRADHURST, R. A. et al. A hybrid modeling approach to simulating foot-and-mouth disease outbreaks in australian livestock. **Frontiers in Environmental Science**, Frontiers Media SA, v. 3, p. 17, 2015.

CARDENAS, N. C.; LOPES, F. P.; MACHADO, G. Modeling foot-and-mouth disease dissemination in brazil and evaluating the effectiveness of control measures. **bioRxiv**, Cold Spring Harbor Laboratory, p. 2022–06, 2022.

CARDENAS, N. C. et al. Use of network analysis and spread models to target control actions for bovine tuberculosis in a state from brazil. **Microorganisms**, MDPI, v. 9, n. 2, p. 227, 2021.

_____. Multiple species animal movements: network properties, disease dynamics and the impact of targeted control actions. **Veterinary Research**, Springer, v. 53, n. 1, p. 14, 2022.

CHANDRA, R.; JAIN, A.; CHAUHAN, D. S. Deep learning via lstm models for covid-19 infection forecasting in india. **PloS one**, Public Library of Science San Francisco, CA USA, v. 17, n. 1, p. e0262708, 2022.

DESCOVI, G. et al. Towards a blockchain architecture for animal sanitary control. In: **ICEIS (1)**. [S.l.: s.n.], 2021. p. 305–312.

EBLING, D. S. et al. A distributed processing architecture for disease spread analysis in the pdsa-rs platform. 2024.

EHSAN, A. et al. Restful api testing methodologies: Rationale, challenges, and solution directions. **Applied Sciences**, MDPI, v. 12, n. 9, p. 4369, 2022.

FERGUSON, N. M.; DONNELLY, C. A.; ANDERSON, R. M. The foot-and-mouth epidemic in great britain: pattern of spread and impact of interventions. **Science**, American Association for the Advancement of Science, v. 292, n. 5519, p. 1155–1160, 2001.

GALVIS, J. A. et al. Modeling between-farm transmission dynamics of porcine epidemic diarrhea virus: Characterizing the dominant transmission routes. **Preventive Veterinary Medicine**, Elsevier, v. 208, p. 105759, 2022.

HE, S. et al. A discrete stochastic model of the covid-19 outbreak: Forecast and control. **Math. Biosci. Eng**, v. 17, n. 4, p. 2792–2804, 2020.

KAO, R. R. The impact of local heterogeneity on alternative control strategies for foot-and-mouth disease. **Proceedings of the Royal Society of London. Series B: Biological Sciences**, The Royal Society, v. 270, n. 1533, p. 2557–2564, 2003.

KEELING, M. J. et al. Dynamics of the 2001 uk foot and mouth epidemic: stochastic dispersal in a heterogeneous landscape. **Science**, American Association for the Advancement of Science, v. 294, n. 5543, p. 813–817, 2001.

LEWIS, J.; FOWLER, M. **Microservices: a definition of this new architectural term**. 2014. Acessado em: 2024-06-16. Disponível em: <<https://martinfowler.com/articles/microservices.html>>.

MERKEL, D. et al. Docker: lightweight linux containers for consistent development and deployment. **Linux j**, v. 239, n. 2, p. 2, 2014.

Plataforma de Defesa Sanitária Animal do Rio Grande do Sul. **PDSA-RS**. 2020. Acessado em: 2024-06-01. Disponível em: <<https://www.pdsa.rs.br>>.

Red Hat, Inc. **What is a REST API?** 2024. Acessado em: 2024-06-16. Disponível em: <<https://www.redhat.com/en/topics/api/what-is-a-rest-api>>.

_____. **What is Docker?** 2024. Acessado em: 2024-06-16. Disponível em: <<https://www.redhat.com/pt-br/topics/containers/what-is-docker>>.

ROTH, J. A. Animal disease information and prevention materials developed by the center for food security and public health. **Iowa State University Animal Industry Report**, Iowa State University Digital Press, v. 4, n. 1, 2007.

SCHOENBAUM, M. et al. Simulating foot-and-mouth disease in the united states using the animal disease spread model. 2024.

ZHOU, Y.; MA, Z.; BRAUER, F. A discrete epidemic model for sars transmission and control in china. **Mathematical and Computer Modelling**, Elsevier, v. 40, n. 13, p. 1491–1506, 2004.