

**INSTITUTO FEDERAL GOIANO - CAMPUS MORRINHOS  
CURSO SUPERIOR DE BACHARELADO EM CIÊNCIA DA  
COMPUTAÇÃO**

**JOÃO VÍTOR MENDES BORGES**

**ARQUITETURA DE MICROSERVIÇOS NO SETOR VETERINÁRIO:  
SOLUÇÕES ESCALÁVEIS PARA COMUNICAÇÃO E BEM-ESTAR  
ANIMAL**

**MORRINHOS - GO  
2025**

**JOÃO VÍTOR MENDES BORGES**

**ARQUITETURA DE MICROSERVIÇOS NO SETOR VETERINÁRIO:  
SOLUÇÕES ESCALÁVEIS PARA COMUNICAÇÃO E BEM-ESTAR  
ANIMAL**

Monografia apresentada ao Curso Superior de Bacharelado em Ciência da Computação do Instituto Federal Goiano – Campus Morrinhos, como requisito parcial para obtenção de título de Bacharel em Ciência da Computação.

**Área de concentração:** Ciência da Computação

**Orientador:** Prof. Dr. Alexandre Carvalho Silva

**MORRINHOS - GO  
2025**

**Dados Internacionais de Catalogação na Publicação (CIP)**  
**Sistema Integrado de Bibliotecas – SIBI/IF Goiano Campus Morrinhos**

B732a Borges, João Vítor Mendes.

Arquitetura de Microserviços no setor veterinário: soluções escaláveis para comunicação e bem-estar animal. / João Vítor Mendes Borges. – Morrinhos, GO: IF Goiano, 2025.  
70 f. : il. color.

Orientador: Dr. Alexandre Carvalho Silva.

Trabalho de conclusão de curso (graduação) – Instituto Federal Goiano Campus Morrinhos, Bacharelado em Ciências da Computação, 2025.

1. Microserviços. 2. Zoonoses. 3. Animais - Proteção I. Silva, Alexandre Carvalho. II. Instituto Federal Goiano. III. Título.

CDU 004.42:621.395

Fonte: Elaborado pela Bibliotecária-documentalista Morgana Guimarães, CRB1/2837

## TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

### IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

- |  |   |
|--|---|
| <input type="checkbox"/> Tese (doutorado)            | <input type="checkbox"/> Artigo científico              |
| <input type="checkbox"/> Dissertação (mestrado)      | <input type="checkbox"/> Capítulo de livro              |
| <input type="checkbox"/> Monografia (especialização) | <input type="checkbox"/> Livro                          |
| <input checked="" type="checkbox"/> TCC (graduação)  | <input type="checkbox"/> Trabalho apresentado em evento |

Produto técnico e educacional - Tipo:

Nome completo do autor:

João Vítor Mendes Borges

Matrícula:

2020104201940080

Título do trabalho:

ARQUITETURA DE MICROSERVIÇOS NO SETOR VETERINÁRIO: SOLUÇÕES ESCALÁVEIS PARA  
COMUNICAÇÃO E BEM-ESTAR ANIMAL

### RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial:  Não  Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIIF Goiano:  /  /

O documento está sujeito a registro de patente?  Sim  Não

O documento pode vir a ser publicado como livro?  Sim  Não

### DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

- Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Documento assinado digitalmente  
 JOAO VITOR MENDES BORGES  
Data: 04/04/2025 15:03:22-0300  
Verifique em <https://validar.iti.gov.br>

Morrinhos - GO  
Local

31 /03 /2025  
Data

Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:

Assinatura do(a) orientador(a)

Documento assinado digitalmente  
 ALEXANDRE CARVALHO SILVA  
Data: 04/04/2025 15:38:01-0300  
Verifique em <https://validar.iti.gov.br>



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 43/2025 - CCEG-MO/CEG-MO/DE-MO/CMPMHOS/IFGOIANO

### ATA DE DEFESA DE TRABALHO DE CURSO

### ATA DE DEFESA DA BANCA DE EXAME DE TRABALHO DE CURSO POR VIDEOCONFERÊNCIA

Aos 31 dias do mês de março de 2025, às 14 horas foi realizada a Banca de Exame, em formato remoto para a apresentação pública e defesa do trabalho de curso do discente **João Vitor Mendes Borges** intitulado: **Arquitetura de Microserviços no Setor Veterinário: Soluções Escaláveis para Comunicação e Bem-Estar Animal**, como requisito necessário para a conclusão do curso Bacharelado em Ciência da Computação - IFGoiano campus Morrinhos.

A Banca de Exame foi constituída pelos membros: **Prof. Dr. Alexandre Carvalho Silva (orientador)**, **Prof. Dr. Reidner Santos Cavalcante** e **Prof. Me. Diogo Aparecido Cavalcante de Lima (Uniessa)**. Após a análise, emitiram o seguinte resultado:

1 - (  ) **Aprovado**

2 - ( ) **Aprovado com ressalva**

(A Banca Examinadora deve definir as exigências a serem cumpridas pelo aluno na revisão, ficando o orientador responsável pela verificação do cumprimento das mesmas.)

Observações: \_\_\_\_\_

3 - ( ) **Reprovado com o seguinte parecer:** \_\_\_\_\_  
\_\_\_\_\_

Morrinhos -GO, 31 de março de 2025.

Por ser verdade firmamos a presente:

*(Assinado Eletronicamente)*

**Prof. Dr. Alexandre Carvalho Silva**

Orientador(a)

*(Assinado Eletronicamente)*

**Prof. Dr. Reidner Santos Cavalcante**

Membro

*(Assinado Eletronicamente)*

**Prof. Me. Diogo Aparecido Cavalcante de Lima**

Membro Externo



Documento assinado digitalmente

**DIOGO APARECIDO CAVALCANTE DE LIMA**

Data: 02/04/2025 14:54:22-0300

Verifique em <https://validar.iti.gov.br>

Documento assinado eletronicamente por:

- Alexandre Carvalho Silva, CHEFE - FG0002 - UPIC-MO, em 31/03/2025 15:26:40.
- Reidner Santos Cavalcante, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 31/03/2025 15:54:13.

Este documento foi emitido pelo SUAP em 31/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 693122

Código de Autenticação: 96e84333f2



INSTITUTO FEDERAL GOIANO

Campus Morrinhos

Rodovia BR-153, Km 633, Zona Rural, SN, Zona Rural, MORRINHOS / GO, CEP 75650-000

(64) 3413-7900

## RESUMO

Este trabalho tem como objetivo mostrar uma forma alternativa a fim de ajudar a informar os tutores dos animais sobre prevenção de zoonoses, colaborando também em função do bem estar animal, onde os tutores receberão lembretes a respeito da vacinação e utilização de remédios de forma regular, a fim de minimizar esquecimentos por parte dos usuários do serviço. Projetada para atuar em locais propícios para divulgação aos tutores como clínicas veterinárias e petshops, podendo atuar nos dois setores. Tendo a proposta de colaborar para a diminuição de zoonoses. Portanto o estudo busca ajudar a combater zoonoses utilizando recursos atuais da programação, sendo utilizado a arquitetura de microsserviços para realizar o repasse de imagens e mensageria aos tutores dos animais, seja de data de vacinação, medicação vencida, ou até mesmo de imagens que o estabelecimento queria divulgar sobre campanhas e serviços realizados dentro do estabelecimento, assim como promoções.

**Palavras-chave:** Zoonoses, Animais, Microsserviços, Mensageria, Imagens.

## **ABSTRACT**

This work aims to show an alternative way to help inform pet owners about the prevention of zoonoses, also contributing to animal welfare, where owners will receive reminders about vaccination and regular use of medicines, in order to minimize forgetfulness on the part of service users. Designed to operate in places suitable for dissemination to owners, such as veterinary clinics and pet shops, and can operate in both sectors. With the proposal to collaborate in the reduction of zoonoses. Therefore, the study seeks to help combat zoonoses using current programming resources, using the microservices architecture to forward images and messages to pet owners, whether about vaccination dates, expired medication, or even images that the establishment wanted to publicize about campaigns and services carried out within the establishment, as well as promotions.

**Keywords:** Zoonoses, Animal, Microservices, Messages, Images.

## FIGURAS

Figura 1 - Distribuição em percentagem das respostas dos tutores com relação a frequência que costumam levar seu animal ao Médico Veterinário. ....	11
Figura 2 - Distribuição em percentagem das respostas dos tutores sobre saberem ou não a importância da vacinação em cães e gatos.....	12
Figura 3 - Caracterização sócio demográfica e em relação ao conhecimento dos tutores dos cães sobre vacinação .....	13
Figura 4 - Frequência de encaminhar o animal para atendimento clínico veterinário e estado de vermifugação de cães e gatos atendidos em campanha de controle de endo e ectoparasitas no Centro de Saúde Animal do UNICERP. ....	14
Figura 5 - Evolução Temporal Da Mortalidade Absoluta Por Febre Maculosa No Brasil de 2001 a 2023 Em Relação ao Ano.....	15
Figura 6 - MailJet, Plataforma de entrega e rastreamento de e-mails baseado em nuvem, utilizada na aplicação para envio de notificações para usuários. ....	18
Figura 7 - Estruturação MongoDB, exemplo de documento JSON .....	18
Figura 8 - Estrutura SMTP.....	19
Figura 9 - Interface e-mail Sender.....	19
Figura 10 - Fluxograma da solicitação de adição de evento no sistema .....	20
Figura 11 - Tela para solicitar participação em evento cadastrado - Expositor .....	21
Figura 12 - Arquitetura de Serviços.....	26
Figura 13 - Caso de uso Clinica Pet.....	28
Figura 14 - Tela inicial Visual Studio Code.....	34
Figura 15 - Tela inicial MongoDB Compass .....	36
Figura 16 - Instalação Docker passo a passo .....	37
Figura 17 - Instalação Docker finalizada .....	38
Figura 18 - Página inicial do Docker.....	38
Figura 19 - Docker pós instalação das imagens.....	39
Figura 20 - Docker pós definição de portas.....	41
Figura 21 - CMD comando ipconfig.....	41
Figura 22 - Acesso local 192.168.100.30:5000 .....	42
Figura 23 - Formulário de cadastro Tutores e Animais .....	43
Figura 24 - Tela para editar tutores e animais.....	43

Figura 25 - Imagens Salvas.....	44
Figura 26 - Micro Vacinas 192.168.100.30:5001 .....	45
Figura 27 - Micro Remedios 192.168.100.30:5002 .....	45
Figura 28 - Micro Envia Imagens 192.168.100.30:5003 .....	45
Figura 29 - Disparo Vacina.....	46
Figura 30 - Disparo Remédio .....	46
Figura 31 - Disparo Imagens.....	47

## TABELAS

Tabela 1 - Comparativos dos trabalhos relacionados. ....	22
Tabela 2 - Descrição de caso de uso "Login".....	29
Tabela 3 - Descrição de caso de uso "Cadastrar Funcionario". ....	29
Tabela 4 - Descrição de caso de uso " Cadastrar Tutor".....	30
Tabela 5 - Descrição de caso de uso "Editar Tutor". ....	30
Tabela 6 - Descrição de caso de uso "Adicionar Observação". ....	30
Tabela 7 - Descrição de caso de uso "Visualizar Animais". ....	31
Tabela 8 - Descrição de caso de uso "Imagens Salvas". ....	31
Tabela 9 - Descrição de caso de uso "Atribuir Servicos".....	31
Tabela 10 - Comandos para instalar as imagens salvas no Docker Hub. ....	39
Tabela 11 - Iniciando Containers.....	40
Tabela 12 - UC03: Cadastrar Tutor. ....	48
Tabela 13 - Resultados UC03: Cadastrar Tutor. ....	48
Tabela 14 - UC08: Atribuir Serviços. ....	48
Tabela 15 - Resultados UC08: Atribuir Serviços. ....	48
Tabela 16 - Porta 5001: Micro Vacinas. ....	49
Tabela 17 - Resultados Porta 5001: Micro Vacinas. ....	49
Tabela 18 - Porta 5002: Micro Remédios. ....	49
Tabela 19 - Resultados Porta 5002: Micro Remédios. ....	49
Tabela 20 - Porta 5003: Micro Imagens. ....	50
Tabela 21 - Resultados porta 5003: Micro Imagens.....	50

## SIGLAS

Caso de Uso (UC)

Máquina virtual (VM)

Amazon Web Service (AWS)

Interface de Programação de aplicativos (API)

Corporação Internacional de Máquinas de Negócios (IBM)

Database Administrator (DBA's)

Sistema Operacional (SO)

Porta serial universal (USB)

Hypertext transfer protocol (HTTP)

Classificação Internacional de doenças (CID)

JavaScript Object Notation (JSON)

Formato de dados binários (BSON)

Open Charge Point Protocol (OCPP)

Simple Mail Transfer Protocol (SMTP)

Aplicativos (APP)

Portable Network Graphic (PNG)

Joint Photographic Experts Group (JPG)

Joint Photographic Experts Group (JPEG)

Requisitos funcionais (RF)

Requisitos não funcionais (RNF)

Visual Studio Code (VS)

Cascading Style Sheets (CSS)

Hypertext Markup Language (HTML)

Windows Subsystem for Linux (WSL)

Prompt de comando (CMD)

Internet Protocol (IP)

# SUMÁRIO

<b>FIGURAS</b> .....	<b>9</b>
<b>TABELAS</b> .....	<b>11</b>
<b>SIGLAS</b> .....	<b>12</b>
<b>INTRODUÇÃO</b> .....	<b>1</b>
ORGANIZAÇÃO DO TRABALHO .....	4
<b>1 REFERENCIAL TEÓRICO</b> .....	<b>5</b>
1.1 INTRODUÇÃO .....	5
1.2 DOCKER .....	5
1.2.1 Funcionamento.....	6
1.2.2 Funcionalidades .....	6
1.2.3 DOCKER OU (VM) .....	7
1.2.4 DOCKER VS MÁQUINA VIRTUAL .....	8
1.3 MICROSERVIÇOS E MONÓLITOS .....	9
1.4 DOCKER E MICROSERVIÇOS .....	10
1.5 ZOONOSES .....	10
1.5.1 Pesquisa sobre visitas ao veterinário .....	11
1.5.2 Tutores não sabem a importância da vacinação .....	11
1.5.3 Vermifugação .....	13
1.5.4 Febre Maculosa.....	14
1.6 CONSIDERAÇÕES FINAIS .....	15
<b>2 TRABALHOS CORRELATOS</b> .....	<b>16</b>
2.1 CONSIDERAÇÕES INICIAIS .....	16
2.2 ARQUITETURA BASEADA EM MICROSERVIÇO PARA GESTÃO ESCOLAR .....	16

2.3 IMPLEMENTAÇÃO DE UMA PLATAFORMA BACKEND PARA INTEGRAÇÃO COM PROTOCOLO OCPP .....	18
2.4 EXPOMAP: UMA APLICAÇÃO DE DIVULGAÇÃO DE EVENTOS AGROPECUÁRIOS NA REGIÃO NORDESTE .....	20
2.5 DESENVOLVIMENTO DE SISTEMA WEB PARA A ASSOCIAÇÃO DE TRANSPORTE URBANO DE VITÓRIA DA CONQUISTA (ATUV).....	21
2.6 COMPARAÇÃO DE TECNOLOGIAS UTILIZADAS.....	22
2.7 CONSIDERAÇÕES FINAIS .....	23
<b>3 ARQUITETURA .....</b>	<b>24</b>
3.1 CONSIDERAÇÕES INICIAIS .....	24
3.2 ARQUITETURA DO SISTEMA .....	25
3.3 REQUISITOS FUNCIONAIS.....	26
3.4 REQUISITOS NÃO FUNCIONAIS .....	27
3.5 CASO DE USO .....	27
3.5.1 Descrição dos Atores .....	28
3.5.2 Descrição dos casos de uso.....	29
3.6 CONSIDERAÇÕES FINAIS .....	32
<b>4 IMPLEMENTAÇÃO .....</b>	<b>33</b>
4.1 PYTHON.....	33
4.2 VISUAL STUDIO CODE .....	33
4.3 HTML E CSS .....	34
4.4 MONGODB.....	35
4.5 CONSIDERAÇÕES FINAIS .....	36
<b>5 FUNCIONAMENTO DO SISTEMA .....</b>	<b>37</b>
5.1 PREPARAÇÃO DO AMBIENTE .....	37
5.2 COMANDOS PARA INSTALAÇÃO DA WEB APLICAÇÃO .....	38
5.2.1 Funcionamento em rede local .....	41
5.2.2 Formulários de criação.....	42

5.2.3 Gerenciar imagens .....	44
5.3 MICROSERVIÇOS .....	44
5.4 ENVIO DA MENSAGEM .....	46
5.5 CONSIDERAÇÕES FINAIS .....	47
<b>6 TESTES DO SISTEMA .....</b>	<b>47</b>
6.1 CONSIDERAÇÕES INICIAIS .....	47
6.2 TESTE DE CAIXA PRETA.....	48
<b>7. CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>51</b>
7.1 CONCLUSÕES.....	51
7.2 TRABALHOS FUTUROS.....	52
<b>REFERÊNCIAS.....</b>	<b>53</b>

## INTRODUÇÃO

Com a integração da domesticação a relação humana e animal se tornou mutualística, onde ambas espécies são beneficiadas, facilitando a sobrevivência, não só servindo para alimentação, mas sendo importante para o desenvolvimento socioeconômico, sendo necessário para a construção da sociedade como conhecemos hoje e permanece em constante evolução (Freire, 2024).

O mutualismo decorrente da dependência dos animais para obter facilidades também resultou na transmissão de doenças conhecidas como zoonoses. Doenças zoonóticas podem ser transmitidas naturalmente de animais vertebrados para humanos ou de humanos para animais vertebrados (Rahman, 2020).

A raiva é uma zoonose viral com letalidade de aproximadamente 100%, raramente tendo casos de cura. O vírus rábico contido na saliva do animal penetra no organismo através de mordedura, raramente pela arranhadura e lambedura sendo que nas cidades, as principais fontes de infecção são os cães e os gatos. O morcego é o principal responsável pela manutenção da cadeia silvestre, podendo ter outros reservatórios silvestres como a raposa, canídeos silvestres, gato-do-mato, jaritaca, guaxinim e macacos (MINISTÉRIO DA SAÚDE, 2016).

A parvovirose canina, é uma das doenças que mais atingem os cães tendo uma taxa de mortalidade em filhotes que contraem o vírus chegando a 35%, afetando principalmente o trato gastrointestinal dos animais. Embora não seja transmitida para seres humanos, a doença tem um alto risco de contágio entre os cães (UFLA, 2019).

Usar medicamentos em tempo incorreto ou dosagem errada também causa riscos à saúde do animal. As utilizações inadequadas de antimicrobianos em animais contribuíram para a resistência ao medicamento, por parte do animal e também da própria bactéria, que pode disseminar bactérias resistentes (Sartori, 2024).

Tratando-se de parasitas, carrapatos e pulgas existem remédios com duração entre 30 a 90 dias, podendo variar dependendo do medicamento (Pimenta, 2022). O uso de medicamentos é muito importante para o combate a zoonoses em relação a doença do carrapato, tanto o comum quanto o carrapato estrela responsável

pela febre maculosa onde a Secretaria de Estado e saúde (SES-MG) registrou que entre 2018 e 2022, ocorreram 190 casos somente em Minas Gerais, onde 62 pacientes vieram a óbito, tendo uma taxa de letalidade de 33% (Castro Alves, 2023).

A evolução tecnológica tem transformado profundamente a maneira como as soluções de software são projetadas, desenvolvidas e mantidas. Em um cenário onde as demandas por escalabilidade, flexibilidade, eficiência e manutenção são crescentes, é essencial identificar soluções que minimizem problemas relacionados à comunicação com clientes e aos custos elevados de aplicações tradicionais oferecidas por serviços mais utilizados (Marques, 2024).

Este trabalho é especialmente relevante para estabelecimentos como: clínicas veterinárias e pet shops, que visa demanda de comunicação com seus clientes a fim de divulgar produtos ou serviços oferecidos pelo estabelecimento. Especialmente por cuidar de animais que podem conter esses parasitas ou necessitar de serviços como vacinação, a aplicação busca minimizar os casos alertando os clientes sobre a duração do remédio ou realização de vacinas.

Uma pesquisa recente feita no Nordeste revelou que 87% dos tutores não vacinam os animais domésticos representando a pior situação em comparação à média do país, indicando também que os tutores estão desatentos à imunização de seus pets, chegando a 92% (Lucena, 2024).

As vacinas anuais para cães e gatos são a polivalente e antirrábica, que previnem diversos tipos de contaminação. Não vacinar um animal anualmente pode desencadear problemas inclusive aos tutores, como por exemplo no caso da raiva, onde um animal contaminado coloca em risco seus tutores entre outras doenças como a febre maculosa causada pelo carrapato estrela.

Mesmo com a importância de manter os animais vacinados, o número de tutores que não realizam a vacinação ainda é alto, negligenciando as vacinas de seus animais e elevando o risco de doenças infecciosas como cinomose, parvovirose e leptospirose. Isso coloca em risco a saúde dos próprios animais e de seus tutores (Pires; Corrêa, 2020).

Para se prevenir dessas enfermidades infecciosas e parasitas, é necessário que os animais sejam vacinados e tratados, diminuindo riscos à saúde pública, sendo que no Brasil o incentivo da vacinação de cães e gatos feita pelos órgãos públicos. O médico veterinário é fundamental para disseminar informações e orientar os tutores sobre os protocolos de vacinas (Anjos, 2025).

A fim de prevenir e medicar os animais temos como forma de vacinas a polivalente e a antirrábica. Vacina polivalente protege contra doenças como cinomose, parvovirose, hepatite infecciosa, parainfluenza, adenovirose e leptospirose. Enquanto a antirrábica previne contra a raiva. Ambas vacinas necessitam de reforço anual.

De acordo com o portal Butantan, os casos recentes de raiva humana no Brasil apresentam uma letalidade de quase 100% tanto em animais quanto em pessoas infectadas. Ao longo dos anos, a vacina antirrábica, disponibilizada no país, tem sido fundamental no controle do vírus. Distribuída pelo Instituto Butantan para o Sistema Único de Saúde (SUS), essa vacina desempenha um papel essencial na prevenção da doença (Butantan, 2023).

Para ajudar no combate de zoonoses pode ser utilizado a arquitetura de microsserviços que oferece uma solução escalável, flexível e economicamente viável para enviar lembretes a respeito de zoonoses, ajudando a evitar esquecimento por parte dos tutores.

Portanto, este trabalho tem como objetivo demonstrar como a arquitetura de microsserviços pode atender clínicas veterinárias e pet shops, promovendo soluções que visam minimizar problemas relacionados à vacinação e parasitismo que podem fazer mal ao animal e conseqüentemente ao tutor. Espalhando assim informações sobre vencimento de medicações para os donos de animais, que possuem cadastros em clínicas veterinárias e petshops para ajudar no combate a zoonoses.

## ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em sete capítulos:

O primeiro capítulo apresenta definições e conceitos sobre as tecnologias utilizadas no trabalho e sua relevância.

O Segundo capítulo mostra trabalhos relacionados, onde temos a utilização de tecnologias semelhantes para propósitos diferentes e semelhantes.

O terceiro capítulo relata a arquitetura utilizada para implementação dos objetivos propostos.

O quarto capítulo apresenta as partes principais da implementação das aplicações propostas, bem como as tecnologias utilizadas.

O quinto capítulo apresenta a preparação do ambiente e modo de instalação e utilização da web aplicação proposta.

O sexto capítulo demonstra o teste da caixa preta, utilizado no trabalho.

No sétimo capítulo, é apresentada a conclusão e os trabalhos futuros que podem ser realizados a fim de melhorar o projeto.

# 1 REFERENCIAL TEÓRICO

## 1.1 INTRODUÇÃO

Este capítulo apresenta os fundamentos teóricos relacionados aos temas abordados neste trabalho, tendo objetivo de delimitar e aprofundar o conhecimento sobre os assuntos abordados.

## 1.2 DOCKER

Criado em 2013 pela empresa dotCloud, Inc que após lançar a ferramenta mudou o nome para Docker, INC. tendo em vista a necessidade de melhorar a demanda de máquinas virtuais. A criação de máquinas depende de grandes espaços de disco e outros recursos, que poderiam ser corrompidos em algumas situações pela própria máquina virtual. Tendo isso em vista surgiu o LXC.

Criado em 2008 LXC (Linux Containers) visa a criação de instâncias isoladas de um SO, dentro de uma máquina hospedeira sendo assim a base do Docker. Com a facilidade de uso teve um grande número de usuários aderindo à ferramenta, devido aos desenvolvedores conseguirem colocar suas ideias em produção utilizando apenas um notebook.

Em 2014 Docker se tornou a terceira plataforma mais utilizada pelos desenvolvedores e empresas famosas como a Microsoft, Google, Red HAT (IBM) começaram a utilizar sua capacidade. Com o investimento dessas grandes empresas, a comunidade de desenvolvimento ajudou a ferramenta a se aprimorar e os desenvolvedores e DBAs utilizarem Docker em seus projetos (4LINUX, 2025).

No Stack Overflow, Docker e Kubernetes sempre estiveram entre os tópicos mais buscados. Inicialmente uma tendência no mercado estrangeiro, essa popularidade fez com que essas tecnologias se tornassem uma realidade também no Brasil.

Docker é uma plataforma de contêiner de software projetado para desenvolver, enviar e executar aplicativos, aproveitando a tecnologia de contêiner (Oracle, 2025). Funcionando como empacotador de software, unidades que contêm o necessário para os programas funcionarem e serem executados em qualquer máquina que tenha Docker instalado.

Docker é uma ferramenta de código aberto, muito utilizada para criar, enviar e executar aplicativos distribuídos em diferentes formatos, utilizando containers, ambientes isolados dentro de um SO, podendo ser Windows, Linux ou MAC, obrigatoriamente tendo 64 bits (Garcia; Pereira, 2019).

### 1.2.1 Funcionamento

O Docker faz uso do kernel do Linux e seus recursos como Cgroups, Namespaces e UnionFS, utilizado para criação de imagens, essas imagens podem ser imutáveis e sua apresentação é feita em camadas, podendo ser compartilhada entre si.

Com as imagens é possível a criação de contêineres, e dentro existe um SO que compartilha com o kernel a máquina hospedeira. A partir do momento de sua criação, os processos ficam isolados através dos Namespaces, ficando totalmente separado de outras aplicações, conseguindo visualizar os processos dentro da máquina hospedeira para o contêiner, porém o inverso não é possível.

### 1.2.2 Funcionalidades

Versionamento: Docker permite ao usuário que versione as alterações dos contêineres. Podendo fazer commit, rollback, e observar a diferença entre as versões.

Compartilhamento: Utilizando o Docker hub, (mais conhecido como github de contêineres), é possível encontrá-los com aplicações já configuradas onde o

usuário pode fazer o download, para outras pessoas conseguirem ter acesso a mesma aplicação.

CLI (Command Line Interface): permite criar scripts e programas que interagem com os serviços Docker.

Open Source: Código livre licenciado pelo Apache, onde os usuários têm acesso à códigos-fonte.

Alta Performance: criar imagens e baixar imagens prontas de outros colaboradores é possível, e utilizar essa facilidade para criar um ambiente mais rápido utilizando pouco recurso de sua máquina.

### 1.2.3 DOCKER OU (VM)

Docker e máquinas virtuais são tecnologias utilizadas na implantação de softwares, com o passar do desenvolvimento da tecnologia a implantação prepara o código da aplicação para ser executado para seus usuários finais. Docker é uma plataforma capaz de empacotar softwares, em unidades padronizadas (contêineres).

O contêiner inclui o código da aplicação e seu ambiente, abrangendo todas as bibliotecas, ferramentas do sistema e dependências utilizadas na implementação, além do tempo de execução de cada ação (4LINUX, 2024). O uso do Docker ajuda a garantir que a aplicação será executada corretamente em qualquer sistema operacional, desde que a ferramenta esteja instalada.

As Máquinas virtuais (VMs) tem o objetivo de fornecer virtualização em uma máquina (servidor). Essa máquina tem a função de emular os componentes de um hardware físico como CPU, memória, placa de rede, placas de som e controladores USB. Podendo ser executado o SO convidado e várias aplicações no ambiente virtual gerado pela VM (AWS, 2025).

Máquinas virtuais possibilitam as tecnologias de nuvem da AWS, tendo seu nome chamado de instância. Essas instâncias de nuvem são de propriedades AWS e podem ser utilizadas via APIs.

## 1.2.4 DOCKER VS MÁQUINA VIRTUAL

Apesar de ambas abordarem o desafio de executar aplicações em diferentes ambientes, contam com motivos e abordagens diferentes.

Máquinas virtuais são projetadas para permitir vários SO executados em uma única máquina física. Seu objetivo é permitir que usuários criem um ambiente virtual isolado do hardware. VMs abstraem os detalhamentos do hardware para facilitar a execução das aplicações em diferentes arquiteturas de hardware a fim de usar os recursos com mais eficiência (AWS, 2025).

Docker possui a mesma proposta, porém fornece de uma maneira leve e portátil de empacotar e executar os softwares em ambiente isolado e reproduzível, abstraindo os detalhes do SO para enfrentar o desafio de implantar aplicações em diferentes ambientes como desenvolvimento, testes e produção.

Gerenciar atualizações no ambiente de software e garantir a consistência desse ambiente em diferentes locais é um desafio significativo para organizações que operam com centenas de aplicações ou dividem sistemas em múltiplos microsserviços. O Docker oferece uma solução eficaz para esse problema por meio de sua estrutura baseada em contêineres, que proporciona maior controle, uniformidade e simplicidade na implantação e manutenção de aplicações.

A utilização do Docker depende de fatores sendo recursos leves ou arquitetura de microsserviços, quando o ambiente de infraestrutura física é distribuído, incluindo serviços baseados em nuvem, ciclos de implantação rápidos e escalabilidade rápida.

Para a escolha de uma máquina virtual, é necessário ter requisitos mais específicos como dependência de um SO específico, requisitos de recursos de hardware substanciais, necessidade de definir vários controles em SO, aplicações legadas que não são mais executadas em um SO moderno ou diferentes requisitos do sistema operacional com uma única infraestrutura física (AWS, 2025).

### 1.3 MICROSERVIÇOS E MONÓLITOS

Microserviços é uma abordagem para desenvolver um único aplicativo com um conjunto de pequenos serviços, cada um rodando em seu próprio processo e se comunicando com mecanismos leves, via API ou HTTP (Fowler, 2014).

Cada microserviço é independente e implementa uma funcionalidade específica, sua capacidade de escalar serviços de forma individual permite eficiência no uso de recursos, a descentralização distribui a lógica do sistema em serviços separados. A comunicação via API facilita a comunicação entre os serviços, caso um dos serviços falhe, outros serviços podem continuar operando minimizando o impacto do sistema.

Os monólitos são sistemas cuja arquitetura concentra todas as regras de negócio em um único sistema, resultando em uma única unidade de implantação (deploy). Isso significa que, ao realizar uma alteração, é necessário implantar toda a aplicação novamente, uma vez que todas as regras e funcionalidades estão integradas em um único bloco dentro da aplicação (Newman, 2022).

Quando uma aplicação monolítica cresce e se torna maior do que o esperado, haverá problemas que dificultaram o entendimento e manutenção do código, onde dificulta ou impede a implementação de novas funcionalidades por ser preciso atualizar o sistema toda vez que for necessário implementar uma nova funcionalidade ou fazer alguma mudança (Richardson, 2014)

Muitas empresas ainda possuem sistemas monolíticos antigos que continuam funcionando por serem estáveis e atenderem, e-commerce de pequeno porte que possui poucas funcionalidades também podem fazer uso do benefício de monólitos. Ferramentas internas que não requerem grande escalabilidade para gerenciar tarefas monólitos ainda acabam sendo uma boa opção.

## 1.4 DOCKER E MICROSERVIÇOS

A utilização dessas tecnologias juntas é capaz de criar softwares flexíveis escaláveis e eficientes. Docker desempenha um papel crucial no desenvolvimento de aplicações baseadas em microsserviços devido à capacidade de isolamento, portabilidade, rapidez e reprodutibilidade.

Os benefícios de utilizar Docker em microsserviços se dá por conta do desenvolvimento desacoplado, onde cada equipe pode trabalhar em um microsserviço por serem independentes, utilizando linguagens, frameworks e bibliotecas diferentes, sem interferir em outros serviços.

Por ter containers leves, portáteis e isolados do ambiente de hospedagem, garante que serviços sejam executados em diferentes ambientes, independente da configuração de hardware e software (Costa, 2023). Facilitando assim tanto testes, quanto colocar a aplicação para “rodar” em outros computadores, que tenham o Docker instalado.

## 1.5 ZOONOSES

A relação de dependência entre seres humanos e animais, seja por apego, trabalho ou alimentação, levou ao surgimento de doenças entre as espécies. O termo zoonose como doença pode ser definido em doenças que são transmitidas dos animais para os seres humanos, ou do ser humano para os animais (Hubálek, 2003).

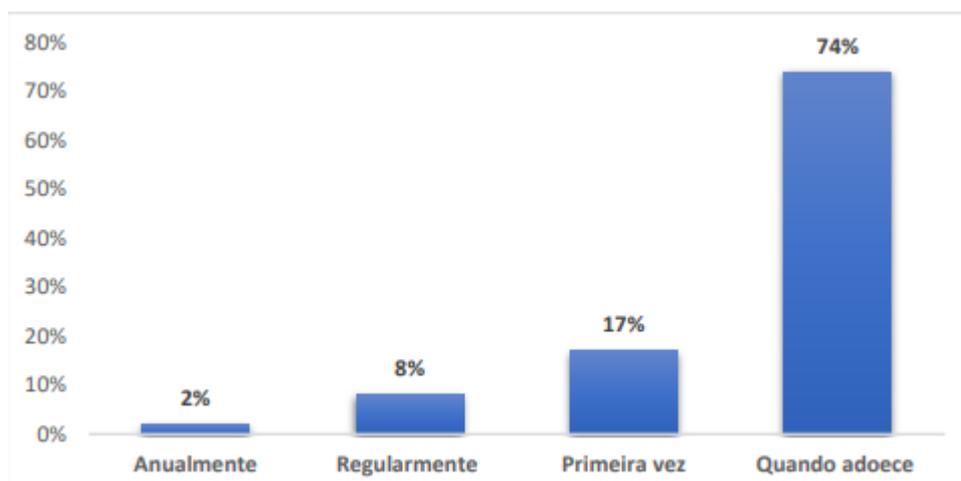
Utilizando a mesma abordagem, porém a mais atual sobre zoonoses, as zoonoses são doenças e infecções transmitidas de animais para seres humanos, sejam pelo contato com sangue ou saliva. Com a evolução da humanidade, começou-se a domesticação de diversos animais e, com isso, o surgimento de várias doenças (Oliveira Gomes, 2022). Tendo ciência de que dependemos deles tanto por vínculo afetivo ou para trabalho, é importante a vacinação anual e utilização de medicamentos para prevenção de zoonoses.

1. Vacina Polivalente: Protege contra cinomose, parvovirose, hepatite infecciosa, parainfluenza, adenovirose e leptospirose (dose anual).
2. Vacina Antirrábica: Protege contra raiva (dose anual).
3. Remédios contra parasitas: A proteção pode variar entre 30 e 90 dias, dependendo da idade e peso do animal.

### 1.5.1 Pesquisa sobre visitas ao veterinário

Conforme apresentado na Figura 1, em uma entrevista 106 tutores, com animais em atendimento no Hospital Veterinário da UFPB – Campus II, Areia (Paraíba), a pesquisa mostrou dados importantes quanto à cautela dos tutores em relação aos animais, o que demonstra quando os donos levam os animais ao médico veterinário (Morais, 2022).

Figura 1 - Distribuição em percentagem das respostas dos tutores com relação a frequência que costumam levar seu animal ao Médico Veterinário.



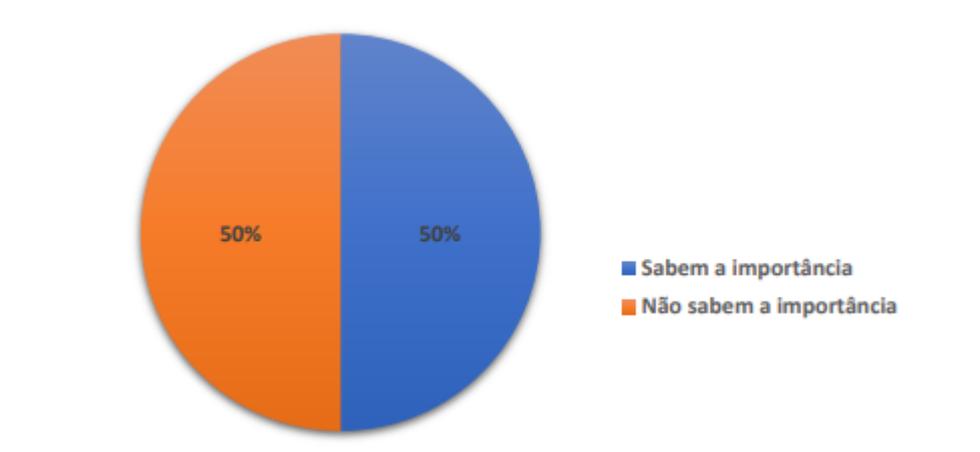
Fonte: Morais (2022).

### 1.5.2 Tutores não sabem a importância da vacinação

Mesmo com as campanhas de vacinação, ainda existem muitos tutores que não sabem da importância da vacinação em cães e gatos. Em uma pesquisa, os

números se empataram entre as pessoas que sabiam da importância da vacinação e aquelas que não tinham conhecimento sobre o porquê vacinar.

Figura 2 - Distribuição em percentagem das respostas dos tutores sobre saberem ou não a importância da vacinação em cães e gatos.



Fonte: Moraes (2022).

Em outra pesquisa realizada em Patos de Minas, é possível observar que as orientações são passadas por veterinários, amigos ou pela internet, com uma boa porcentagem de informações transmitidas por veterinários. Apesar de muitos terem ensino médio e superior completos, não conhecem ou não têm informações a respeito das doenças nem das vacinas.

Figura 3 - Caracterização sócio demográfica e em relação ao conhecimento dos tutores dos cães sobre vacinação.

Variável	Descrição da variável	Frequência absoluta(n)	Frequência relativa (%)
<b>Escolaridade</b>	Ensino fundamental completo	2	6,9
	Ensino médio completo	8	27,6
	Ensino superior completo	16	55,2
	Ensino superior incompleto	3	10,3
	Total	29	100
<b>Quantidade de cães que possui</b>	Um	9	31
	Dois	13	44,8
	Três	3	10,3
	Quatro	1	3,4
	Cinco	2	6,9
	Sete	1	3,4
	Total	29	100
<b>Fontes de informações a respeito das vacinas animais</b>	Faculdade	1	3,4
	Médico veterinário	10	34,5
	Médico veterinário e amigos	1	3,4
	Médico veterinário e internet	7	24,1
	Médico veterinário, internet e amigos	10	35,5
Total	29	100	
<b>Vacinas de cães que conhece</b>	Antirrábica	2	6,9
	Antirrábica e polivalente	27	93,1
	Total	29	100
<b>Sabe o motivo de vacinar</b>	Não	1	6,9
	Sim	28	96,6
	Total	29	100
<b>Frequência das orientações recebidas pelo Médico Veterinário</b>	Algumas vezes	8	27,6
	Várias vezes	15	51,7
	Nenhuma	6	20,7
	Total	29	100
<b>Quantidade de doenças caninas que conhece</b>	Apenas a raiva	5	17,2
	Raiva e cinomose	1	3,4
	Raiva e parvovirose	4	13,8
	Raiva, parvovirose e cinomose	16	55,6
	Mais que 4 doenças	3	10,3
Total	29	100	
<b>Conhecimento por folders informativos</b>	Não	9	31
	Sim	20	69
	Total	29	100

Fonte: Ferreira; Pereira; Vieira (2023).

### 1.5.3 Vermifugação

Mesmo sendo indicado a vermifugação do animal logo na infância, e continuar vermifugando ao longo de toda sua vida, muitos donos optam por realizar o ato somente quando notar algum problema em relação ao animal ou suas fezes. Grande porcentagem dos donos esquece de vermifugar seu animal ou nunca realizou.

Figura 4 - Frequência de encaminhar o animal para atendimento clínico veterinário e estado de vermifugação de cães e gatos atendidos em campanha de controle de endo e ectoparasitas no Centro de Saúde Animal do UNICERP.

Questões	Alternativas	Frequência (%)	p-valor
Idas ao veterinário	Sim (periodicamente)	32,91	0,984
	Sim (eventualmente)	32,91	
	Não	34,18	
Realização de vermifugação	Não lembra	22,78	p<0,001
	Menos de 6 meses	36,71	
	Entre 6 meses e um ano	24,05	
	Mais de um ano	11,39	
	Nunca foi realizado	5,06	

Fonte: Donis; Silva (2023).

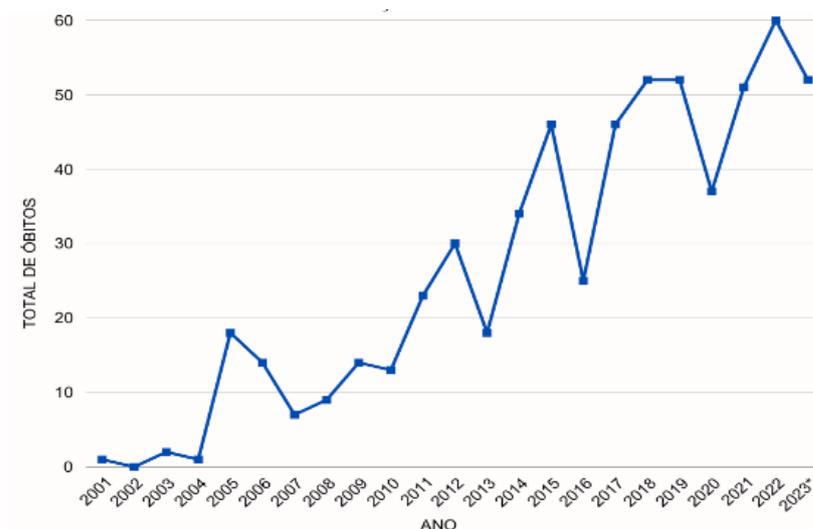
#### 1.5.4 Febre Maculosa

Apesar de não ser uma doença específica do carrapato comum, não é difícil encontrar o carrapato estrela entre os animais de estimação, essa espécie específica de carrapato pode causar a transmissão de doenças para os seres humanos.

Transmitida pelo carrapato estrela, para que ocorra a transmissão é necessário que o carrapato infectado fique ao menos quatro horas fixado na pele. Os carrapatos mais jovens e de menor tamanho são os que oferecem mais riscos por serem mais difíceis de serem vistos, (Varella, 2018).

A pesquisa realizada por Silva como mostra na figura 5 o Brasil entre os anos 2001 e 2023 teve uma evolução temporal da mortalidade por febre maculosa no país (Silva, 2024). Onde é mostrado que com o passar dos anos o número de mortes pela doença Febre Maculosa vem aumentando, tanto em números quanto em taxa de mortalidade.

Figura 5 - Evolução Temporal Da Mortalidade Absoluta Por Febre Maculosa No Brasil de 2001 a 2023 Em Relação ao Ano.



Fonte: Silva (2024).

## 1.6 CONSIDERAÇÕES FINAIS

Observando as informações e gráficos a respeito de zoonoses, é possível observar que mesmo com campanhas tanto de vacinação quanto de prevenção de zoonoses, os índices continuam muito altos mesmo nos dias atuais, seja por esquecimento ou negligência por parte dos tutores.

Espera-se que o modelo apresentado possa ajudar na divulgação e alertas sobre a prevenção e data correta da vacinação e uso dos medicamentos, como também colaborar com o bem-estar do animal em relação a prevenção contra essas enfermidades que colocam em risco a vida dos animais e tutores.

## 2 TRABALHOS CORRELATOS

### 2.1 CONSIDERAÇÕES INICIAIS

É possível identificar diversos trabalhos e aplicações relacionados à utilização de microsserviços, dockerização, envio automatizado de e-mails e notificações, tanto em áreas como saúde, comércio eletrônico, agropecuário, quanto em soluções voltadas à comunicação eficiente com usuários. Esses trabalhos demonstram como a adoção de microsserviços pode ser uma estratégia eficaz para integrar sistemas, criar novos serviços e garantir escalabilidade, eficiência e independência operacional, reduzindo custos e otimizando processos.

Ao analisar os trabalhos correlacionados, nota-se a presença de características semelhantes, como a busca por modularidade e rapidez, além de diferenças relevantes que refletem as especificidades de cada contexto, na utilização de tecnologias. Essa abordagem destaca o papel dos microsserviços como uma solução versátil para atender às necessidades de diversos setores e envio de notificações para um propósito em comum, tendo como objetivo principal a melhoria da comunicação, divulgação e notificações.

### 2.2 ARQUITETURA BASEADA EM MICROSERVIÇO PARA GESTÃO ESCOLAR

Entre os trabalhos correlacionados, destaca-se o TCC de Jonas Ayres da Silva, intitulado "µIF: Arquitetura Baseada em Microsserviços para Gestão Escolar" (Silva, 2021). Esse trabalho apresenta uma solução inovadora para digitalizar e automatizar processos administrativos no campus Restinga do IFRS, abordando desafios como a lentidão e a suscetibilidade a erros em solicitações manuais. O autor

concentrou seus esforços em três tipos principais de solicitações: trancamento de matrículas, trancamento de disciplinas e justificativas de faltas.

A aplicação foi cuidadosamente projetada com os seguintes objetivos:

1. Automatizar processos manuais: Substituir práticas tradicionais por soluções digitais que reduzem a incidência de erros humanos e otimizam o tempo de resposta.

2. Garantir escalabilidade: Criar um sistema capaz de atender às demandas variáveis ao longo do tempo, especialmente em períodos de pico como inícios de semestres.

3. Reduzir custos operacionais: Minimizar o custo de desenvolvimento e consumo de recursos computacionais.

A solução adotou uma arquitetura baseada em microsserviços, na qual cada funcionalidade é tratada como um serviço independente que recebem chamadas via HTTP. Essa divisão permitiu a implementação de módulos especializados, como o envio de notificações automáticas utilizando a plataforma MailJet. Esse recurso garantiu que alunos e coordenadores recebessem informações em tempo hábil sobre o status de suas solicitações, melhorando a comunicação e a eficiência do processo.

O uso do MongoDB desempenhou um papel fundamental na solução proposta, servindo como o banco de dados principal para armazenar e gerenciar as informações de cada serviço. Por ser um banco de dados NoSQL, o MongoDB possibilitou uma alta flexibilidade na modelagem de dados, permitindo que cada serviço tivesse sua própria estrutura de armazenamento independente e adaptável. Além disso, sua capacidade de escalabilidade horizontal garantiu que o sistema pudesse lidar com grandes volumes de dados sem comprometer o desempenho, um aspecto crucial para períodos de alta demanda.

Figura 6 - MailJet, Plataforma de entrega e rastreamento de e-mails baseado em nuvem, utilizada na aplicação para envio de notificações para usuários.

```

app.use('/api/sendEmail', (req, res) => {
  const body = req.body;

  mailjet.post('send', { 'version': 'v3.1' }).request({
    'Messages': [
      {
        'From': { 'Email': 'adm.temporario.ifrs@gmail.com' },
        'To': [ { 'Email': body.to } ],
        'Subject': body.subject,
        'TextPart': body.content,
        'CustomID': 'IFRSEmailSender'
      }
    ]
  }).then((result) => {
    res.status(200).send();
  }).catch((err) => {
    res.status(500).json({ ApiStatusCode: err.statusCode });
  });
});

```

Fonte: Silva (2021).

Figura 7 - Estruturação MongoDB, exemplo de documento JSON.

```

  _id: ObjectId("5fb588b36d3b13002432c418")
  anexos: Array
  nomeCompleto: "Fulano da Silva"
  matricula: 10070100
  anoSemestreIngresso: "2019/1"
  curso: "ADS"
  justificativas: Array
  0: Object
    _id: ObjectId("5fb588b36d3b13002432c419")
    dataDaFalta: 2020-10-10T00:00:00.000+00:00
    disciplinas: Array
    0: Object
      avaliacaoSubstitutiva: false
      _id: ObjectId("5fb588b36d3b13002432c41a")
      nome: "Redes I"
      professor: "Esteves"
      motivo: "demaisCasos"
    dataEncaminhamento: 2020-11-18T20:48:51.848+00:00

```

Fonte: Silva (2021).

## 2.3 IMPLEMENTAÇÃO DE UMA PLATAFORMA BACKEND PARA INTEGRAÇÃO COM PROTOCOLO OCPP

No trabalho de José Mário Reisswitz, a implementação de uma plataforma backend para integração com o protocolo OCPP destacou o uso de tecnologias modernas para atender às necessidades da startup SAVE (Reisswitz, 2024). Entre essas tecnologias, o MongoDB foi utilizado no microserviço event-store para armazenar e gerenciar os eventos enviados pelo sistema.

O autor optou por essa solução devido à sua capacidade de lidar com dados não estruturados e à flexibilidade para armazenar eventos de diferentes formatos, o que é essencial em sistemas baseados em microsserviços que demandam alta escalabilidade e consultas rápidas. O uso do MongoDB permitiu que o sistema armazenasse cada evento como documentos independentes, com suporte para filtros avançados e agrupamento de dados, além de ser adequado para manipular grandes volumes de informações de forma eficiente.

Outro ponto importante do trabalho foi o envio de relatórios por e-mail, implementado com a biblioteca SMTPLIB da linguagem Python. Essa abordagem proporcionou uma integração simples e direta com servidores SMTP para envio de mensagens automatizadas. O autor utilizou a biblioteca para enviar relatórios de histórico de cargas em formato CSV aos responsáveis pelo sistema, configurando uma frequência periódica de envio.

Figura 8 - Estrutura SMTP.

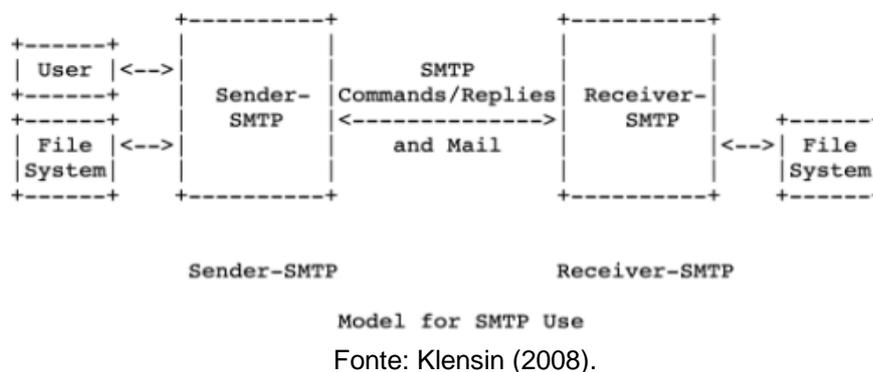


Figura 9 - Interface e-mail Sender.

```
class EmailSender(ABC):
    @abstractmethod
    def send_email(self, request: EmailSenderRequest):
        pass
```

Fonte: Reisswitz (2024).

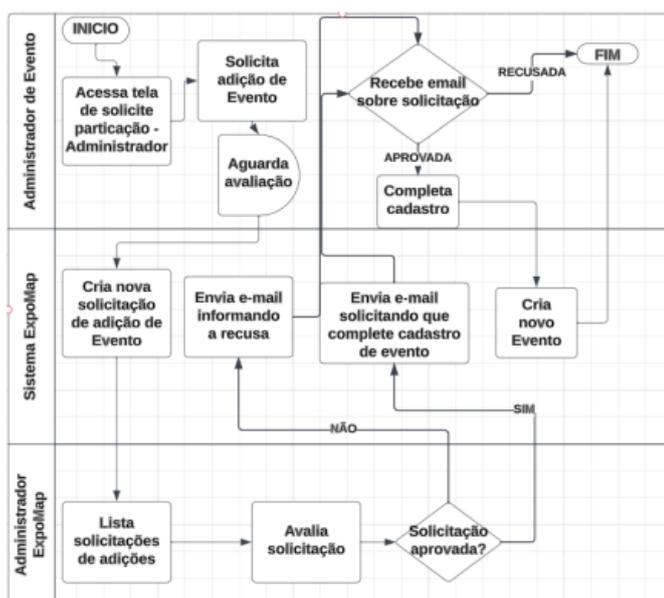
## 2.4 EXPOMAP: UMA APLICAÇÃO DE DIVULGAÇÃO DE EVENTOS AGROPECUÁRIOS NA REGIÃO NORDESTE

O trabalho feito Matheus Eduardo Rodrigues, trata-se de uma web aplicação voltada para divulgação de eventos agropecuários na região Nordeste do Brasil, tendo como principal objetivo criar um sistema de mapeamento unificado de eventos agropecuários, auxiliando a interação entre expositores e administradores de eventos (Silva, 2023).

No seu desenvolvimento foram utilizadas três camadas, frontend, backend e banco de dados, dentre as tecnologias utilizadas para o trabalho destaca-se a utilização do SMTP do Gmail, e a utilização da biblioteca Nodemailer, responsável por realizar o contato com os Administradores, informando se foi aprovado ou não a solicitação de participação dos Expositores.

Em determinados cenários os e-mails serão enviados, quando um expositor solicitar participação em um evento ou quando um administrador do EXPOMAP aceita ou recusa um evento. Ambos os casos de aceite ou recusa o e-mail será enviado para garantir um fluxo eficiente de comunicação entre administradores e expositores.

Figura 10 - Fluxograma da solicitação de adição de evento no sistema.



Fonte: Silva (2024).

Figura 11 - Tela para solicitar participação em evento cadastrado – Expositor.

The screenshot shows the ExpoMap website interface. At the top left is the ExpoMap logo. To the right are navigation links: 'Sobre', 'Eventos', and 'Participe'. Below the navigation is the heading 'Solicite sua participação'. Underneath, there is a 'Você é ?' section with two buttons: 'Expositor' (highlighted in green) and 'Administrador'. The main form area is titled 'Solicite sua participação' and contains the following fields:

- 'Qual evento deseja participar?': A dropdown menu with 'Expofeira Paraíba' selected.
- 'Nome': A text input field containing 'Matheus'.
- 'Email': A text input field containing 'matheuslineudo@hotmail.com'.
- 'Telefone': A text input field containing '83981602510'.
- 'Trabalhou na edição anterior?': A dropdown menu with 'Não' selected.
- 'Explique como deseja participar do evento': A text area containing the text: 'Posso uma barraca de 3 metros de largura por 3 metros de comprimento, onde vendo brinquedos diversos. Já trabalho em outros eventos de exposição, porém nunca trabalhei no seu evento e gostaria de poder fechar negócio e participar.'

At the bottom of the form is a green 'Solicitar' button.

Fonte: Silva (2024).

## 2.5 DESENVOLVIMENTO DE SISTEMA WEB PARA A ASSOCIAÇÃO DE TRANSPORTE URBANO DE VITÓRIA DA CONQUISTA (ATUV)

O problema abordado no trabalho de Adriano de Jesus Alves e Alexandro dos Santos Silva, é a falha na comunicação que gerava problemas como inconsistência de dados e dificuldade de acesso (Alves; Silva, 2022). Para resolver isso, o novo sistema propõe uma comunicação mais eficiente entre as instituições e a ATUV por meio de microsserviços e APIs REST.

A aplicação tem como função automatizar e agilizar o processo de atualização dos cadastros dos alunos, evitando problemas como a falta de padronização, dificuldade na busca de alunos e cadastramento incorreto de alunos com matrículas canceladas, agilizando também o recadastramento de cartões de passe estudantil, para uso do transporte público municipal de Vitória da Conquista.

O objetivo do artigo é apresentar uma solução para a consulta de alunos matriculados em outras instituições de ensino em Vitória da Conquista, de modo em que a (ATUV) consiga gerir os cartões e bilhetes.

O sistema possui cinco microsserviços, responsáveis por tarefas únicas, sendo eles:

1. Microsserviço de Autenticação e Autorização – Gerência login e segurança dos usuários utilizando JWT (JSON Web Token).
2. Microsserviço de API de Recepção de Dados – Recebe as listas de alunos matriculados enviadas pelas instituições de ensino.
3. Microsserviço da API de Requisição de Dados – Serviço responsável por consultar e importar alunos em sistemas externos mantidos por instituições de ensino.
4. Microsserviço de Consulta– Fornece um endpoint para buscar alunos pelo nome ou parte dele, e o código da instituição de ensino.
5. Microsserviço de Administração – Responsável pela manutenção dos cadastros que permitem o acesso aos demais microsserviços.

## 2.6 COMPARAÇÃO DE TECNOLOGIAS UTILIZADAS

A tabela abaixo tem como objetivo, visualizar as tecnologias utilizadas nos trabalhos correlatos como a utilização de microsserviços, docker e a utilização de algum serviço de mensageria.

Tabela 1 - Comparativos dos trabalhos relacionados.

Nome Autor	Microsserviços	Docker	MSG
Jonas	✓	✓	✓
Reisswitz	✓	✓	✓
Matheus	✗	✗	✓
Adriano e Alexandre	✓	✓	✓

Fonte: Elaborada pelo autor (2025)

O símbolo ✓ indica a utilização de tecnologia, enquanto o ✗ indica a não utilização.

## 2.7 CONSIDERAÇÕES FINAIS

Mesmo tendo temas e abordagens diferentes, principalmente em seus ramos distintos, sendo eles, agropecuário, escolar, transporte e integração com plataformas, ambos estudos tendem a solucionar problemas, buscando a melhoria da comunicação e a forma em que as requisições podem ser enviadas para o usuário.

Dentre os trabalhos é possível observar que mesmo tendo o mesmo objetivo, buscaram por formas diferentes de solucionar determinados problemas, utilizando métodos gratuitos ou pagos e que ambos métodos buscam pela qualidade de serviço, velocidade e escalabilidade.

Apesar do trabalho feito por Matheus Eduardo Rodrigues não fazer uso de microsserviços e dockerização, a utilização do recurso SMTP, para enviar e-mails conseguiu solucionar a comunicação entre os expositores e administradores dos eventos de forma gratuita, mesmo não fazendo uso de microsserviços nem do Docker conseguiu suprir a necessidade de notificar administradores sobre os expositores que querem participar dos eventos.

## 3 ARQUITETURA

Este capítulo aborda algumas etapas de desenvolvimento e tem como objetivo mostrar o planejamento e execução para a criação do sistema e seus microsserviços. Sendo possível visualizar a utilização dos conceitos ditos no referencial teórico, tais como dockerização, microsserviços e envio de e-mails, seja para lembretes ou com outros intuitos do estabelecimento.

### 3.1 CONSIDERAÇÕES INICIAIS

O projeto apresentado abaixo tem com intuito mostrar a utilização de microsserviços para manter a escalabilidade do sistema, melhorando a comunicação com os usuários de uma rede petshop e clínica de forma automatizada, enviando notificações como vacinação, remédios e imagens, afim de evitar zoonoses. Promovendo também outros serviços ofertados pelo estabelecimento.

A utilização da linguagem Python e do framework Flask foi feita pela sua simplicidade, minimalismo e flexibilidade, permitindo criar aplicações web, tanto no computador quanto no celular. Flask é um microframework leve e interage facilmente com o MongoDB, além de evitar sobrecarga desnecessária.

Ao escolher o serviço SMTP, a escolha ocorre devido à escalabilidade da aplicação e do número de microsserviços utilizados. A utilização do SMTPlib evita adicionar bibliotecas extras, reduzindo riscos de incompatibilidade, além de ser nativa do Python, permite criar filas de e-mails, enviando mensagens de forma assíncrona e com um maior controle sobre a aplicação.

Docker está sendo utilizado como forma de manter a individualidade de cada serviço, podendo ser controlado a ativação ou desativação, bem como seu funcionamento em qualquer dispositivo, através da imagem e containerização, ajudando a controlar também o custo operacional da máquina.

Os e-mails enviados para os usuários devem ocorrer conforme a programação dos dados, como vacinas anuais e prazo de remédios. Com o passar do

tempo ocorre o vencimento desses medicamentos, onde param de fazer efeito no animal, para vacinas de um ano, e para remédios cerca de noventa dias, tendo esse prazo vencido o animal ficará vulnerável.

Quando o usuário que possui a chave, enviar e-mails (todos) os usuários cadastrados presentes no banco de dados recebem, seja passando a informação sobre o prazo de vacinação estar vencido, remédios ou imagens.

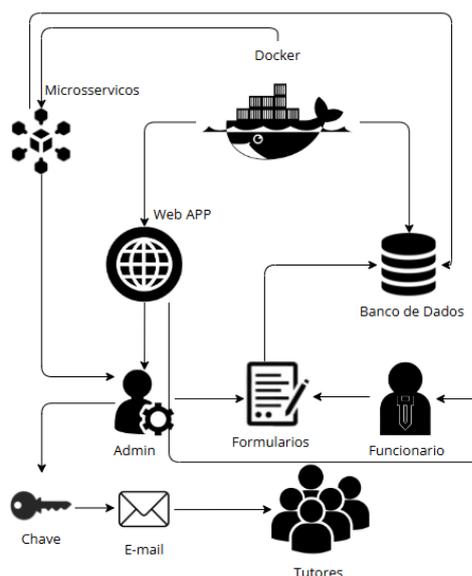
Cada prazo de vencimento é definido em seu único microserviço, onde é possível delimitar o envio do e-mail através de certas regras, evitando assim o envio de e-mail indesejado para os usuários, podendo ser de acordo com as datas (vacinação e remédio).

O estudo busca formas de evitar zoonoses e espalhar informação através de lembretes via e-mail de forma automatizada, onde o usuário pode escolher imagens para enviar para os usuários cadastrados no sistema de forma gratuita além de poder alternar os serviços em determinadas épocas do ano como notificações sobre a vacina do animal, remédios e imagens seja promovendo campanhas ou para benefício próprio do utilitário.

## 3.2 ARQUITETURA DO SISTEMA

A proposta da arquitetura é montar a web aplicação a fim de facilitar o entendimento do projeto, para que o usuário possa visualizar o projeto de forma simplificada. Segundo Crawley (2015) Arquitetura de sistemas é o estudo de tomada de decisões iniciais em sistemas complexos, com foco na escolha de arquitetura que atendam às necessidades das integrações.

Figura 12 - Arquitetura de Serviços.



Fonte: Elaborada pelo autor (2025)

A Figura 12 mostra como está a organização da aplicação, onde temos a Web APP, Microserviços e o banco de dados armazenados no Docker com todo o restante da aplicação e demais serviços incluindo as requisições via formulário. A utilização de containerização nos microserviços permite ao usuário ativar no momento que for usar e desativar quando não é necessário sem prejudicar a Web APP, por ficarem de fora da aplicação.

### 3.3 REQUISITOS FUNCIONAIS

Requisitos funcionais são aqueles que ditam o que o usuário pode fazer dentro do sistema, onde as declarações de serviço do usuário, reage a entradas específicas, gerando diferentes ações com a Web Aplicação.

Os RF são:

RF1: Login administrador e funcionário.

RF2: Validação de requisições e erros.

RF3: Após inserir a chave, os microserviços podem ser disparados.

RF4: Repetição de notificações por serviços.

RF5: Salvar imagens (PNG, JPG, JPEG).

RF6: Alteração de dados nativo na aplicação.

RF7: Envio de mensageria via microsserviços.

### 3.4 REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais têm um papel importante que determina a aceitação ou não de requisições feita pelo usuário, podendo ser interpretado como ditar o que o usuário não pode fazer no sistema

RNF presentes:

RNF1: Administrador não é criado via requisições.

RNF2: Funcionários não criam funcionários.

RNF3: Acesso à chave via banco de dados não disponível.

RNF4: Tutores não possuem login nem senha.

RNF5: Definição de mensagem via página não disponível.

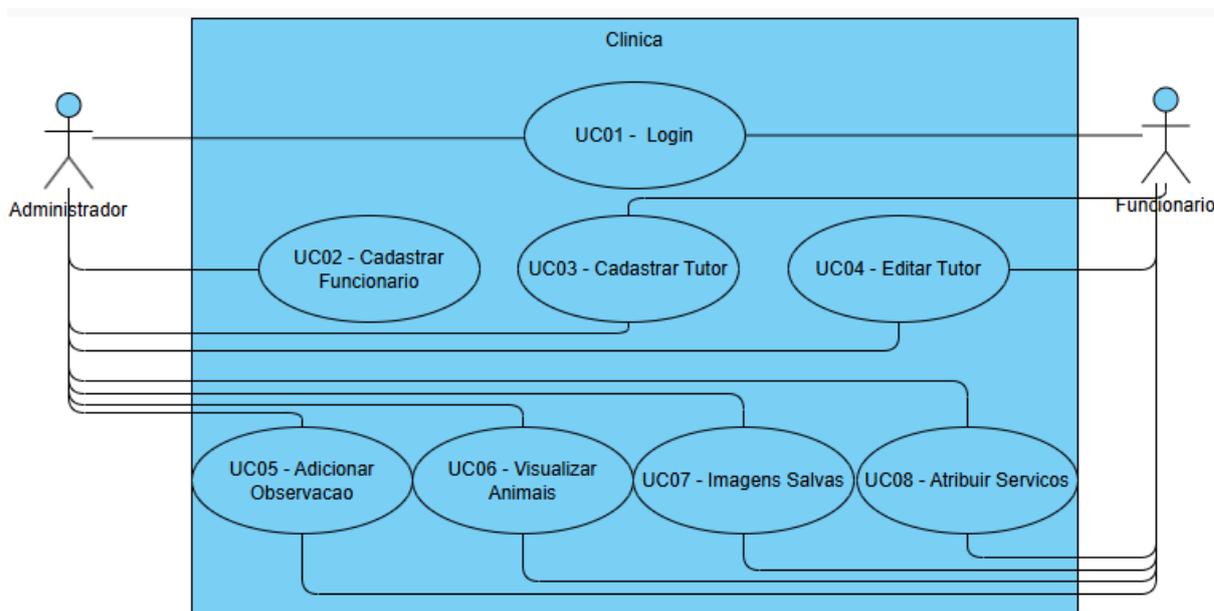
RNF6: Alteração de chave somente back-end.

RNF7: Enviar imagens fora do padrão estabelecido (PNG, JPG, JPEG).

### 3.5 CASO DE USO

Diagrama de caso de uso descreve como os diferentes tipos de usuário usam as funcionalidades do sistema para atingir objetivos, utilizado para entender o comportamento do sistema (Kautz, 2022).

Figura 13 - Caso de uso Clinica Pet.



Fonte: Elaborada pelo autor (2025).

Ao observar os diagramas é possível identificar as funcionalidades do sistema, como funciona e estão sendo utilizadas no projeto. Sendo cada UC apresentada uma página onde o usuário tem funcionalidades específicas.

Ambos microsserviços utilizados de fora da aplicação necessita de uma chave de envio, usada para enviar requisições, não sendo possível fazer o envio e alterar as mensagens diretamente pelos microsserviços ou aplicação. Para realizar a alteração de mensagem é necessário ter acesso ao back-end da aplicação.

### 3.5.1 Descrição dos Atores

Os respectivos diagramas possuem apenas 2 atores tendo apenas uma funcionalidade de criação dos funcionários como diferença, sendo eles Administrador e Funcionários. Ao utilizar qualquer microsserviço basta ter a chave para conseguir realizar disparos, podendo ser compartilhada entre o administrador que possui a chave para o funcionário desejado.

### 3.5.2 Descrição dos casos de uso

Tabela 2 - Descrição de caso de uso "Login".

<b>Caso de uso</b>	<i>UC01 – Login.</i>
<b>Ator Principal</b>	<i>Administrador e Funcionario</i>
<b>Descrição</b>	<i>O papel do Administrador é gerir a aplicação tendo acesso a todos os recursos de criação e modificação que o sistema oferece Funcionário não gera funcionário</i>
<b>Pré-Condição</b>	<i>Inexistente</i>
<b>Fluxo Normal</b>	<i>1. Criação de funcionários -Criação de tutores e animais -Alterações nos cadastros</i>
<b>Fluxos Excepcionais</b>	<i>Inexistente.</i>
<b>Pós-Condição</b>	<i>Painel Geral</i>

Fonte: Elaborada pelo autor (2025).

Tabela 3 - Descrição de caso de uso "Cadastrar Funcionario".

<b>Caso de uso</b>	<i>UC02 – Cadastrar Funcionario</i>
<b>Ator Principal</b>	<i>Administrador</i>
<b>Descrição</b>	<i>O usuário pode criar quantos funcionários desejar</i>
<b>Pré-Condição</b>	<i>Usuário Administrador</i>
<b>Fluxo Normal</b>	<i>1. Nome, senha, e-mail e número</i>
<b>Fluxos Excepcionais</b>	<i>Inexistente</i>
<b>Pós-Condição</b>	<i>Inexistente</i>

Fonte: Elaborada pelo autor (2025).

Tabela 4 - Descrição de caso de uso "Cadastrar Tutor".

<b>Caso de uso</b>	<i>UC03 - Cadastrar Tutor</i>
<b>Ator Principal</b>	<i>Administrador e Funcionário</i>
<b>Descrição</b>	<i>Podendo ser criado pelo computador ou celular, o usuário consegue cadastrar tutores e seus animais.</i>
<b>Pré-Condição</b>	<i>Data existente no calendário.</i>
<b>Fluxo Normal</b>	<i>1. Responder campos do formulário de forma completa.</i>
<b>Fluxos Excepcionais</b>	<i>Inexistente</i>
<b>Pós-Condição</b>	<i>UC04 – Editar Tutor e UC05 – Adicionar Observacao</i>

Fonte: Elaborada pelo autor (2025).

Tabela 5 - Descrição de caso de uso "Editar Tutor".

<b>Caso de uso</b>	<i>UC04 – Editar tutor</i>
<b>Ator Principal</b>	<i>Administrador ou Funcionário</i>
<b>Descrição</b>	<i>Capacidade de trocar dados, tanto de tutores quanto de seus respectivos animais.</i>
<b>Pré-Condição</b>	<i>Confirmação para alterar</i>
<b>Fluxo Normal</b>	<i>1. Após a confirmação os dados serão trocados, em caso de erro o usuário será informado</i>
<b>Fluxos Excepcionais</b>	<i>Inexistente</i>
<b>Pós-Condição</b>	<i>Inexistente</i>

Fonte: Elaborada pelo autor (2025).

Tabela 6 - Descrição de caso de uso "Adicionar Observação".

<b>Caso de uso</b>	<i>UC05 – Adicionar Observação</i>
<b>Ator Principal</b>	<i>Administrador e Funcionário</i>
<b>Descrição</b>	<i>Adicionar informações extras sobre o animal se necessário</i>
<b>Pré-Condição</b>	<i>Não obrigatório</i>
<b>Fluxo Normal</b>	<i>1. Ter cadastro completo no sistema</i>
<b>Fluxos Excepcionais</b>	<i>Inexistente</i>
<b>Pós-Condição</b>	<i>UC06 - Visualizar Animais</i>

Fonte: Elaborada pelo autor (2025).

Tabela 7 - Descrição de caso de uso "Visualizar Animais".

<b>Caso de uso</b>	<i>UC06 – Visualizar Animais</i>
<b>Ator Principal</b>	<i>Administrador e Funcionário</i>
<b>Descrição</b>	<i>Informações dos animais cadastradas em “cadastrar tutor” disponíveis para visualização e “Observação”.</i>
<b>Pré-Condição</b>	<i>Estar logado</i>
<b>Fluxo Normal</b>	<i>1. Ter cadastro completo no sistema</i>
<b>Fluxos Excepcionais</b>	<i>Inexistente</i>
<b>Pós-Condição</b>	<i>UC04 – Editar Tutor e UC05 – Adicionar Observacao</i>

Fonte: Elaborada pelo autor (2025).

Tabela 8 - Descrição de caso de uso "Imagens Salvas".

<b>Caso de uso</b>	<i>UC07 – Imagens Salvas</i>
<b>Ator Principal</b>	<i>Administrador e Funcionário</i>
<b>Descrição</b>	<i>O usuário pode salvar imagens que tenham intuito de divulgação, podendo adicionar e excluir</i>
<b>Pré-Condição</b>	<i>Imagens (PNG, JPG, JPEG)</i>
<b>Fluxo Normal</b>	<i>1. Definição de formato</i>
<b>Fluxos Excepcionais</b>	<i>Salvamento em grupos</i>
<b>Pós-Condição</b>	<i>Inexistente</i>

Fonte: Elaborada pelo autor (2025).

Tabela 9 - Descrição de caso de uso "Atribuir Servicos".

<b>Caso de uso</b>	<i>UC08 – Atribuir Servicos</i>
<b>Ator Principal</b>	<i>Administrador e Funcionário</i>
<b>Descrição</b>	<i>O usuário pode salvar serviços que serão utilizados pelos tutores</i>
<b>Pré-Condição</b>	<i>Selecionar Serviço</i>
<b>Fluxo Normal</b>	<i>1. Ter cadastro completo no sistema</i>
<b>Fluxos Excepcionais</b>	<i>Inexistente</i>
<b>Pós-Condição</b>	<i>Inexistente</i>

Fonte: Elaborada pelo autor (2025).

### 3.6 CONSIDERAÇÕES FINAIS

Neste capítulo foi demonstrado como as UC se comportam no funcionamento do sistema indicando o que cada usuário pode fazer, e como podem ser utilizadas, juntamente com suas regras. Onde cada condição pode afetar no funcionamento, indicando algum possível erro caso ela não seja cumprida.

## 4 IMPLEMENTAÇÃO

Neste capítulo será abordado o funcionamento de sistema, tecnologias que auxiliam no processo de desenvolvimento e a comunicação dos microsserviços com a aplicação.

### 4.1 PYTHON

Como plataforma de operação do protótipo foi definido que seria uma aplicação web, por facilitar o acesso e escalabilidade da aplicação, sendo assim foi escolhida uma linguagem de programação web para o back-end, sendo ela o Python em sua versão 3.12.2.

Além de facilitar a escalabilidade, a escolha da linguagem se baseou em outros fatores, como o Flask, o banco de dados utilizado (MongoDB) e microsserviços para mensageria via SMTP. Facilitando também a integração com o Docker já que a proposta de encapsulamento se encaixa com a ideia de microsserviços serem isolados do restante da aplicação.

A utilização da linguagem também é pensada para a criação de novas implementações no futuro, como a possibilidade de integrações com outras ferramentas e a utilização de automações durante os processos, garantindo assim a evolução do código por ser possível desenvolver outras funções de forma eficiente.

### 4.2 VISUAL STUDIO CODE

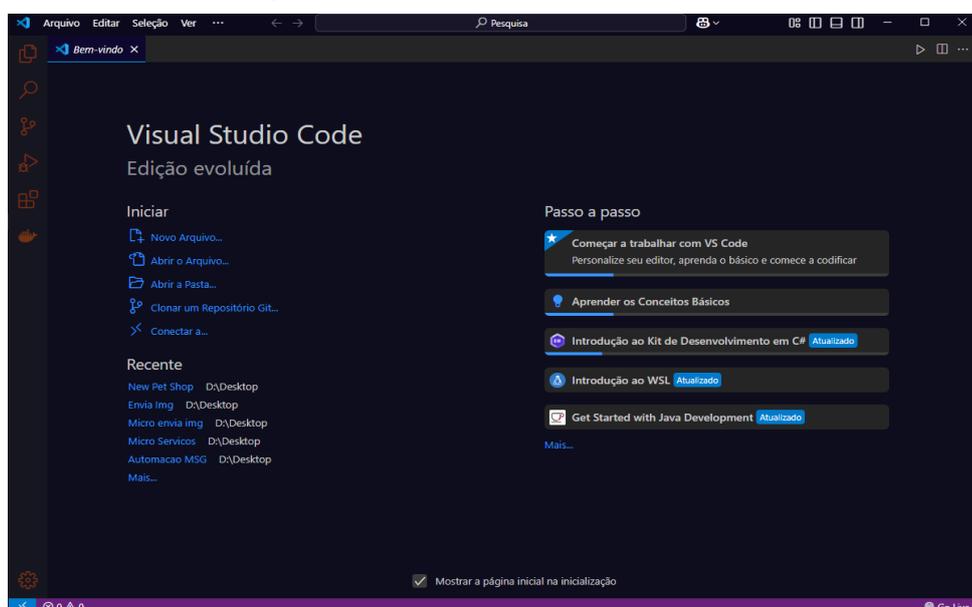
O Visual Studio Code é o editor de código escolhido para o desenvolvimento da web aplicação, pela sua leveza e velocidade em comparação com IDE mais pesadas e complexas como o próprio Visual Studio, pertencente à mesma

empresa Microsoft. Entre suas características se destaca ser gratuito, leve e poderoso, sendo altamente personalizável ao utilizar extensões.

A capacidade de personalização do VS Code possibilita também uma maior facilidade para visualizar erros, seja de indentação ou de sintaxe presentes no código do desenvolvedor, acelerando o processo de desenvolvimento da aplicação e resolução de problemas, fornecendo maior adaptação para seus usuários.

Durante o desenvolvimento a versão utilizada foi a 1.97.2. Outras vantagens de se utilizar VS Code como editor é a compatibilidade com diversas linguagens além das utilizadas no projeto como JavaScript, Java, C++, PHP, Dart entre outras linguagens de programação.

Figura 14 - Tela inicial Visual Studio Code.



Fonte: Elaborada pelo autor (2025).

## 4.3 HTML E CSS

A aplicação web utiliza como base a criação de páginas, utilizando linguagem de marcação, definindo elementos e a organização do conteúdo como textos, imagens, links, tabelas, imagens e todos atributos que fazem parte da página. A versão utilizada no trabalho é a mais atual HTML5.

Ao utilizar o HTML5 para criar as páginas junto com o ambiente visível para o usuário em sua forma bruta, pode-se utilizar o CSS para torná-los mais agradáveis, sendo possível controlar o layout, cores, formatos, fonte de texto, tamanho e ícones, juntamente com animações e efeitos.

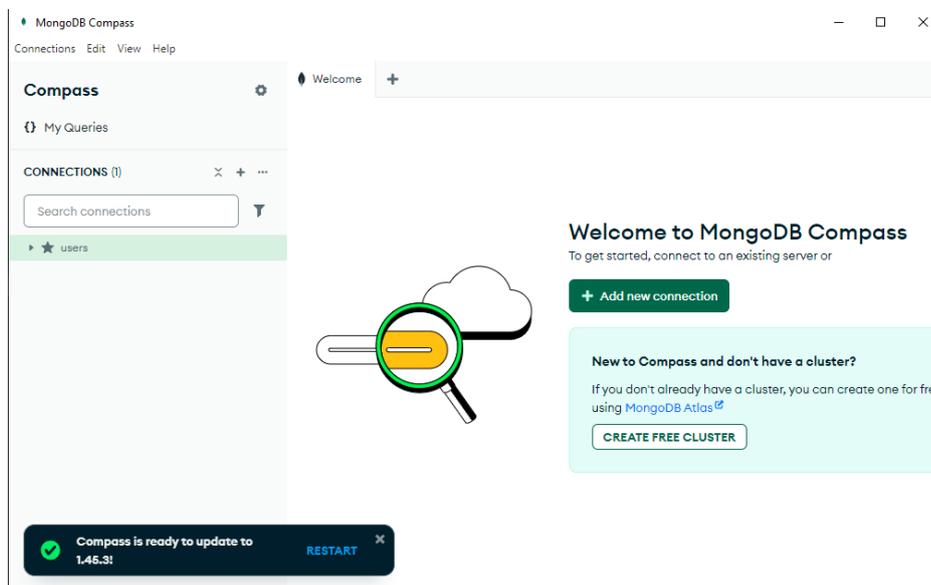
Quando um dos requisitos de um sistema é responsividade, a utilização das tecnologias acima pode ser uma boa opção, podendo definir padrões para ambos dispositivos, fazendo com que a página se adeque a diferentes tamanhos de tela, garantindo a usabilidade em diferentes dispositivos.

## 4.4 MONGODB

Segundo Rathore (2024), o MongoDB é um ator fundamental no gerenciamento moderno de dados. Essa afirmação se dá por conta do alto desempenho oferecido ao desafiar banco de dados relacionais tradicionais e influenciar a adoção do NoSQL de forma global, mostrando como esse banco de dados permite criar aplicativos escaláveis.

A escolha de se utilizar um Banco NoSQL além do fator escalabilidade, é sua fácil integração com o python, onde o desenvolvimento fica simples e de forma direta. O MongoDB armazena dados em formato BSON (extensão para JSON), facilitando o mapeamento de dados entre o banco e o back-end em Python, ajudando também na evolução do código, com a versão utilizada no trabalho sendo 1.45.3.

Figura 15 - Tela inicial MongoDB Compass.



Fonte: Elaborada pelo autor (2025).

## 4.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi possível passar parte do conhecimento utilizado na aplicação de forma simplificada, mostrando as principais tecnologias, suas versões e como se portam na aplicação, assim como o motivo de escolha de cada uma delas. Como pode ser observado, em ambos os casos o fator escalabilidade e aprimoramento do código, foi levado em consideração durante todo processo.

## 5 FUNCIONAMENTO DO SISTEMA

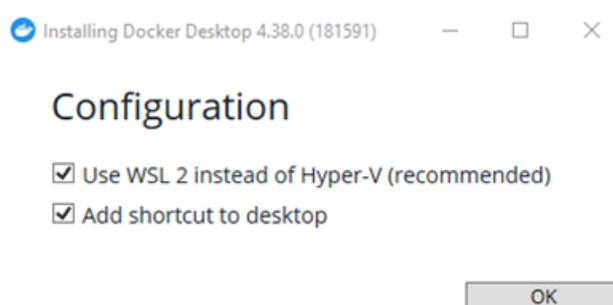
Neste capítulo será abordado o passo a passo da configuração do ambiente, para utilizar o sistema de forma correta com as instalações necessárias para rodar o programa.

### 5.1 PREPARAÇÃO DO AMBIENTE

A instalação do Docker é fundamental para a aplicação podendo ser instalada via link <https://www.docker.com/products/docker-desktop/>. Abaixo o passo a passo de instalação da ferramenta e suas configurações recomendadas.

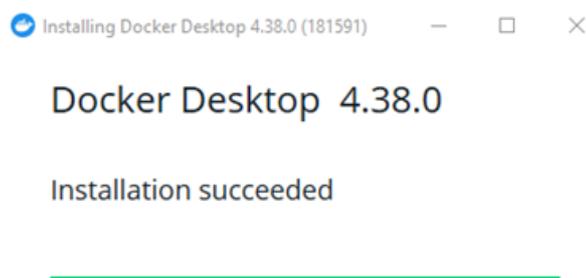
Para que a ferramenta funcione corretamente, como podemos observar na imagem abaixo precisa do WSL 2, além de aceitar a configuração recomendada o computador deve ter acesso a virtualização, que pode ser ativada diretamente na BIOS da placa mãe.

Figura 16 - Instalação Docker passo a passo.



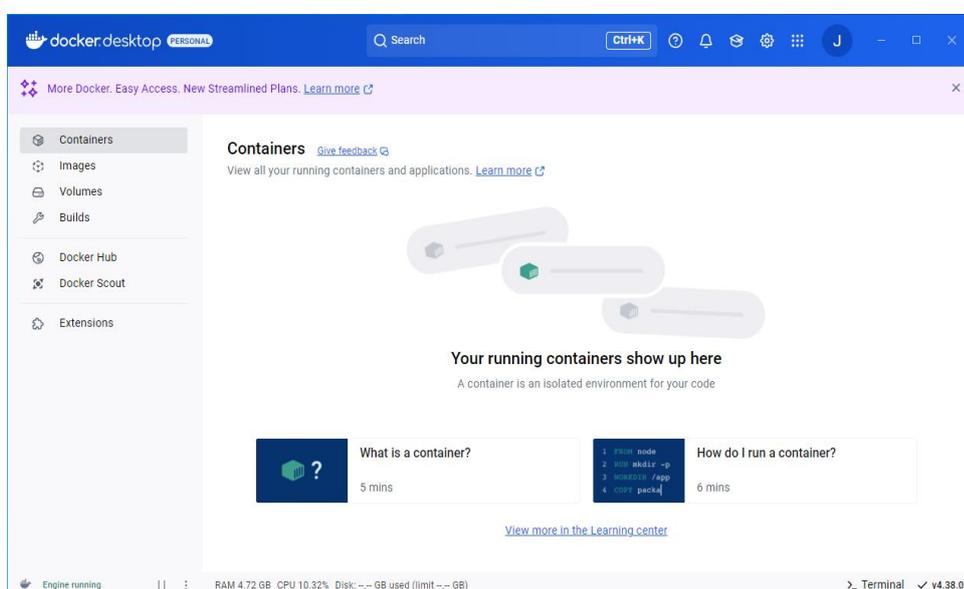
Fonte: Elaborada pelo autor (2025).

Figura 17 - Instalação Docker finalizada.



Fonte: Elaborada pelo autor (2025).

Figura 18 - Página inicial do Docker.



Fonte: Elaborada pelo autor (2025).

## 5.2 COMANDOS PARA INSTALAÇÃO DA WEB APLICAÇÃO

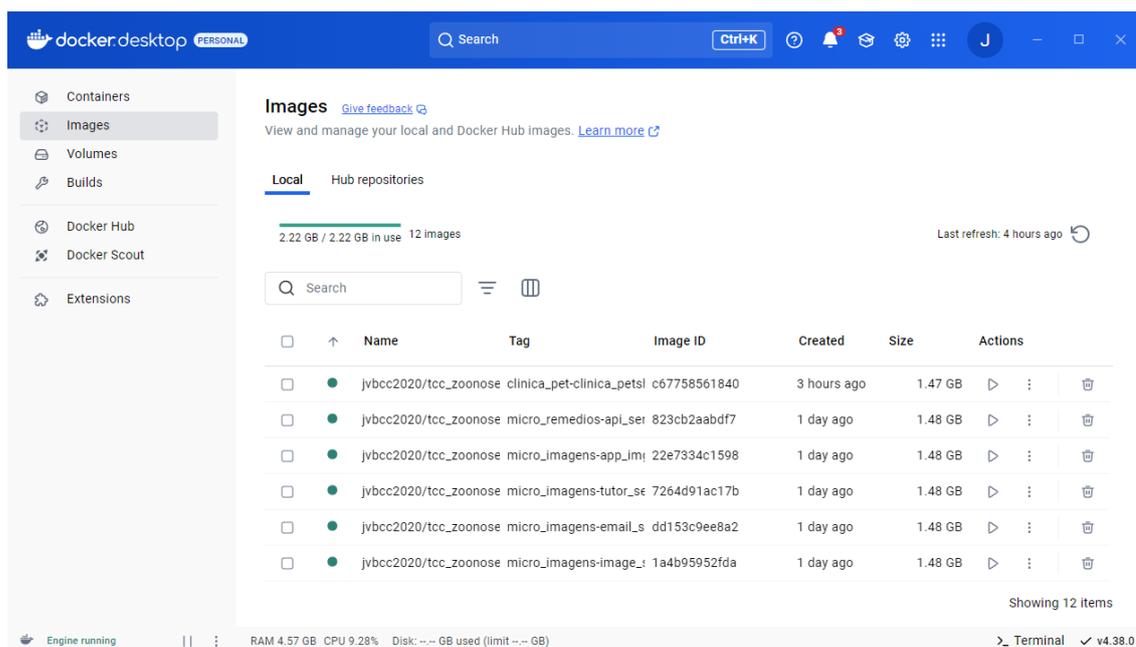
A tabela abaixo tem o intuito de passar como a instalação deve ser feita passo a passo, para fazer download das imagens que estão na conta Docker hub. Utilizando apenas o CMD para inserir os comandos abaixo, e as instalações mostradas anteriormente bastando apenas copiar e colar os comandos da tabela abaixo.

Tabela 10 - Comandos para instalar as imagens salvas no Docker Hub.

<code>docker login</code>
<code>docker pull jvbcc2020/tcc_zoonoses:mongo</code>
<code>docker pull jvbcc2020/tcc_zoonoses:clinica_pet-clinica_petshop</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_imagens-tutor_servicei</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_imagens-image_servicei</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_imagens-email_servicei</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_imagens-app_img</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_remedios-db_servicer</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_remedios-email_servicer</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_remedios-api_servicer</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_vacinas-db_servicev</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_vacinas-email_servicev</code>
<code>docker pull jvbcc2020/tcc_zoonoses:micro_vacinas-api_servicev</code>

Fonte: Elaborada pelo autor (2025).

Figura 19 - Docker pós instalação das imagens.



Fonte: Elaborada pelo autor (2025).

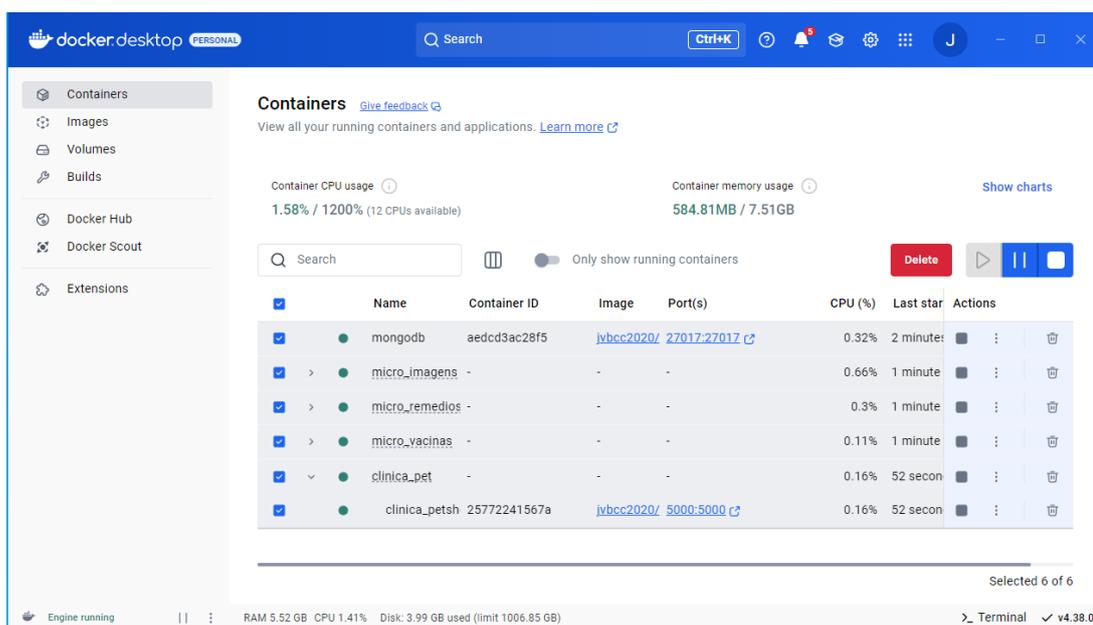
Após a instalação das imagens é preciso especificar em qual porta cada contêiner deve rodar. Para que ocorra a comunicação dos containers mesmo estando em portas diferentes é necessário criar uma rede em comum, a definição dos nomes, criação da rede e portas estão na tabela abaixo a fim de facilitar a instalação.

Tabela 11 - Iniciando Containers.

<code>docker network create pet_network</code>
<code>docker run -d --name mongodb --network pet_network -p 27017:27017 -v C:\tmp\mongodb:/data/db jvbcc2020/tcc_zoonoses:mongo</code>
<code>docker run -d --name clinica_petshop --network pet_network -p 5000:5000 --link mongodb jvbcc2020/tcc_zoonoses:clinica_pet-clinica_petshop</code>
<code>docker run -d --name email_servicev --network pet_network -p 5010:5010 jvbcc2020/tcc_zoonoses:micro_vacinas-email_servicev</code>
<code>docker run -d --name db_servicev --network pet_network -p 5011:5011 jvbcc2020/tcc_zoonoses:micro_vacinas-db_servicev</code>
<code>docker run -d --name api_servicev --network pet_network -p 5001:5001 --link email_servicev --link db_servicev jvbcc2020/tcc_zoonoses:micro_vacinas-api_servicev</code>
<code>docker run -d --name email_servicer --network pet_network -p 5012:5012 jvbcc2020/tcc_zoonoses:micro_remedios-email_servicer</code>
<code>docker run -d --name db_servicer --network pet_network -p 5013:5013 jvbcc2020/tcc_zoonoses:micro_remedios-db_servicer</code>
<code>docker run -d --name api_servicer --network pet_network -p 5002:5002 --link email_servicer --link db_servicer jvbcc2020/tcc_zoonoses:micro_remedios-api_servicer</code>
<code>docker run -d --name tutor_servicei --network pet_network -p 5014:5014 jvbcc2020/tcc_zoonoses:micro_imagens-tutor_servicei</code>
<code>docker run -d --name image_servicei --network pet_network -p 5015:5015 jvbcc2020/tcc_zoonoses:micro_imagens-image_servicei</code>
<code>docker run -d --name email_servicei --network pet_network -p 5016:5016 --link tutor_servicei --link image_servicei jvbcc2020/tcc_zoonoses:micro_imagens-email_servicei</code>
<code>docker run -d --name app_img --network pet_network -p 5003:5003 --link tutor_servicei --link image_servicei --link email_servicei jvbcc2020/tcc_zoonoses:micro_imagens-app_img</code>

Fonte: Elaborada pelo autor (2025).

Figura 20 - Docker pós definição de portas.



Fonte: Elaborada pelo autor (2025).

## 5.2.1 Funcionamento em rede local

Para o usuário conseguir acessar a aplicação web no celular ou no computador é preciso encontrar o IP da máquina hospedeira, responsável por rodar a aplicação em rede local. A imagem abaixo tem o intuito de mostrar como encontrar o IP da máquina via CMD, através do comando “ipconfig” utilizado para exibir as configurações de IP.

Figura 21 - CMD comando ipconfig.

```
Prompt de Comando
C:\Users\Joao Vitor>ipconfig

Configuração de IP do Windows

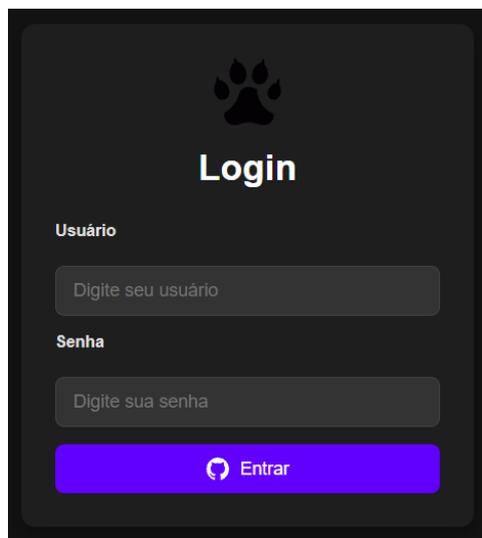
Adaptador Ethernet Ethernet:

    Sufixo DNS específico de conexão. . . . . :
    Endereço IPv4. . . . . : 192.168.100.30
    Máscara de Sub-rede . . . . . : 255.255.255.0
    Gateway Padrão. . . . . : 192.168.100.1
```

Fonte: Elaborada pelo autor (2025).

Após encontrar o endereço IPV4 da máquina, basta acessar utilizando o endereço encontrando como no caso acima, e a porta definida, como no caso abaixo da aplicação web principal.

Figura 22 - Acesso local 192.168.100.30:5000.

A imagem mostra uma interface de login com um fundo escuro. No topo, há um ícone de pata preta. Abaixo dele, o título "Login" está em branco. Seguem os campos de entrada: "Usuário" com o placeholder "Digite seu usuário" e "Senha" com o placeholder "Digite sua senha". Ambos os campos são retangulares com cantos arredondados e um fundo cinza claro. Abaixo dos campos, há um botão azul com o texto "Entrar" e um ícone de seta para a direita.

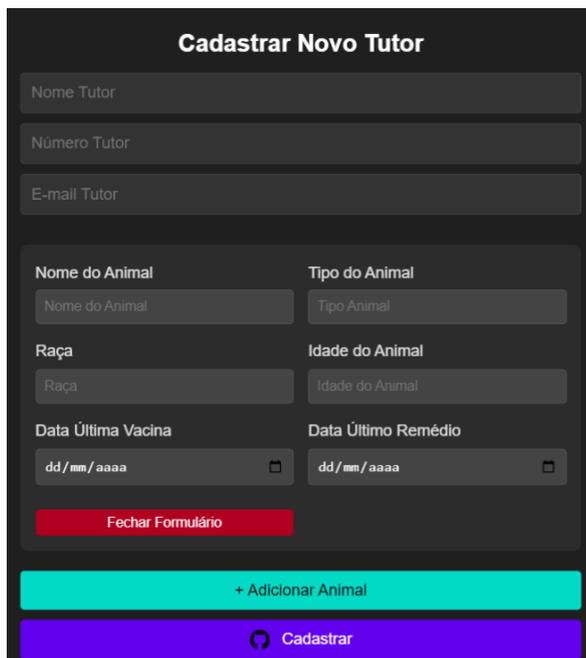
Fonte: Elaborada pelo autor (2025).

## 5.2.2 Formulários de criação

Na tela painel, temos dois formulários onde é possível a criação de funcionários e tutores, ao preencher as informações corretamente dos tutores, o usuário garante o funcionamento do envio de mensagens, ao utilizar os microsserviços da aplicação.

Conforme a imagem abaixo é possível visualizar como é feito o cadastro dos tutores e de seus animais, precisando inserir corretamente o e-mail utilizado pelo tutor, a data correta da última vacina e último remédio para que as notificações via e-mail cheguem de forma efetiva ao tutor cadastrado.

Figura 23 - Formulário de cadastro Tutores e Animais.



O formulário, intitulado "Cadastrar Novo Tutor", é apresentado em um tema escuro. Ele contém os seguintes campos de entrada:

- Nome Tutor
- Número Tutor
- E-mail Tutor
- Nome do Animal (campo dividido)
- Tipo do Animal (campo dividido)
- Raça (campo dividido)
- Idade do Animal (campo dividido)
- Data Última Vacina (campo dividido com ícone de calendário)
- Data Último Remédio (campo dividido com ícone de calendário)

Na base do formulário, há um botão vermelho "Fechar Formulário". Abaixo dele, há um botão ciano "+ Adicionar Animal" e, no rodapé, um botão roxo "Cadastrar" com um ícone de seta circular.

Fonte: Elaborada pelo autor (2025).

Caso o usuário note alguma informação errada ou que mudou do cliente, ele pode fazer alterações através de uma outra página chamada editar tutor, onde todos os dados cadastrados dele e de seus animais estão.

Como a aplicação roda de forma local, ela está disponível tanto nos computadores que acessam a rede e nos celulares, como é demonstrado na imagem abaixo utilizando um aparelho móvel.

Figura 24 - Tela para editar tutores e animais.



A imagem mostra a interface de um celular com o seguinte conteúdo:

- Barra de status: 20:43, 168.100.30:5000, 73% de bateria.
- Título: Escolha o Tutor para Editar
- Subtítulo: Selecione o Tutor:
- Menu suspenso: Escolha um tutor...
- Botão: Editar
- Título: Editar Tutor: Joao Vítor Mendes Borges
- Nome do Tutor: Joao Vítor Mendes Borges
- Número de telefone: (64) 99214-1203
- E-mail: joaovitormendesborges@gmail.com
- Seção: Animais:

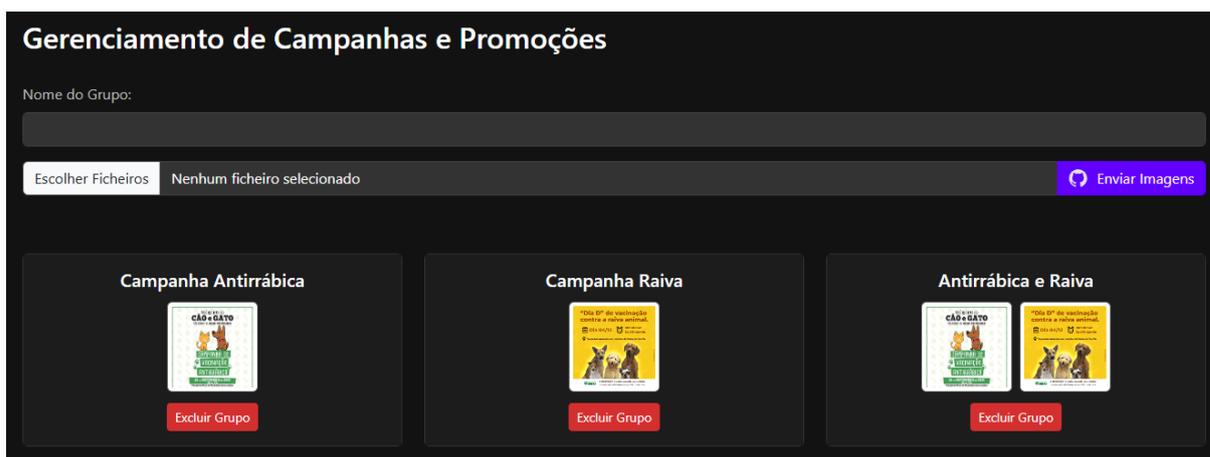
Fonte: Elaborada pelo autor (2025).

### 5.2.3 Gerenciar imagens

A página de gerenciamento de imagens tem como objetivo o usuário promover campanhas e promoções para os tutores cadastrados, podendo notificar os usuários tanto para o bem estar do animal promovendo campanhas gratuitas, até mesmo divulgar produtos vendidos pelo estabelecimento e serviços.

Imagens salvas em um único grupo serão enviadas ao mesmo tempo para os tutores. Como as imagens estão sendo salvas em “Blocos” ao salvar uma imagem junta a outra, ela pertence a um único grupo, caso o usuário exclua o grupo selecionado, as imagens que fazem parte serão deletadas do banco de dados.

Figura 25 - Imagens Salvas.



Fonte: Elaborada pelo autor (2025).

## 5.3 MICROSERVIÇOS

Após a coleta de dados via formulário de tutores e animais, será apresentado a utilização dos microsserviços, onde os dados cadastrados presente no banco de dados serão utilizados para o envio de mensageria, enviando notificações diretamente para os e-mails dos tutores.

Figura 26 - Micro Vacinas 192.168.100.30:5001.

Fonte: Elaborada pelo autor (2025).

Figura 27 - Micro Remedios 192.168.100.30:5002.

Fonte: Elaborada pelo autor (2025).

Figura 28 - Micro Envia Imagens 192.168.100.30:5003.

Micro Envia Imagens		
Nome do Grupo	Imagens	Ações
Campanha Antirrábica		Digite a senha: <input type="password"/> <input type="button" value="Enviar por E-mail"/>
Campanha Raiva		Digite a senha: <input type="password"/> <input type="button" value="Enviar por E-mail"/>
Antirrábica e Raiva	 	Digite a senha: <input type="password"/> <input type="button" value="Enviar por E-mail"/>

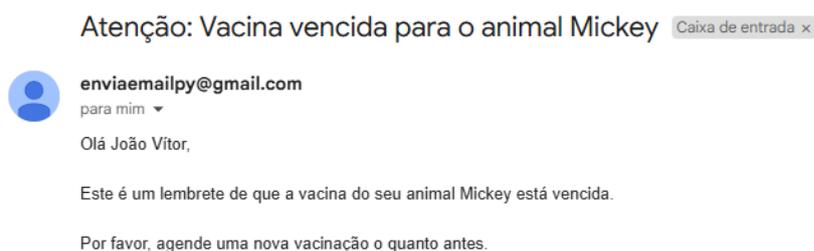
Fonte: Elaborada pelo autor (2025)

## 5.4 ENVIO DA MENSAGEM

Após o usuário digitar a senha e apertar no botão a mensagem será disparada para os e-mails dos usuários que satisfaçam as condições definidas no código como ter o remédio ou a vacinas vencidas a um determinado tempo, sendo 90 dias para remédios e 365 para a vacina. As imagens serão disparadas para qualquer usuário que esteja presente no banco de dados e satisfaça a condição.

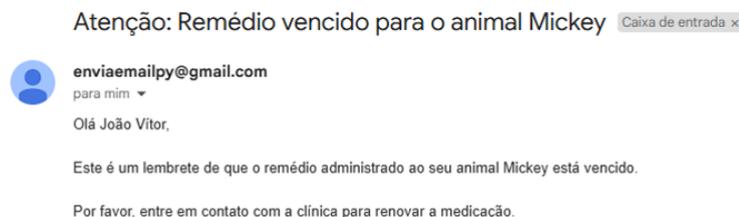
Como mecanismo de segurança, após efetuar os disparos o usuário não terá acesso a aplicação novamente, tendo que fazer login para ter acesso novamente a porta 5000 (Responsável pela aplicação principal).

Figura 29 - Disparo Vacina.



Fonte: Elaborada pelo autor (2025).

Figura 30 - Disparo Remédio.



Fonte: Elaborada pelo autor (2025).

Figura 31 - Disparo Imagens.



Fonte: Elaborada pelo autor (2025).

## 5.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi mostrado como instalar e visualizar a aplicação juntamente com suas funcionalidades e serviços, tanto no Docker quanto nas páginas web.

## 6 TESTES DO SISTEMA

### 6.1 CONSIDERAÇÕES INICIAIS

Os testes do sistema serão realizados através do método teste de caixa preta, na qual tem como foco apenas testes e funcionalidades, desconsiderando o código da aplicação. Testes escolhidos para apresentar funcionalidades referem-se aos das portas iniciais 5000 (UC03, UC08), 5001, 5002 e 5003, onde temos a aplicação e os microsserviços que contemplam as principais funcionalidades do projeto.

## 6.2 TESTE DE CAIXA PRETA

Tabela 12 - UC03: Cadastrar Tutor.

Caso de uso	Condição de entrada	Classes válidas	Classes inválidas
UC03 – Cadastrar Tutor	Estar logado no sistema	1. Após realizar o login aparece o painel para preencher os formulários "Cadastro tutores"	2. Não realizar o cadastro completo.
			3. Inserir datas além do tempo atual ou não inserir @ no e-mail.
			4. Não preencher nenhum dado.

Fonte: Elaborada pelo autor (2025).

Tabela 13 - Resultados UC03: Cadastrar Tutor.

Condição de Entrada	Classes	Resposta
Preencher o formulário completo	1	O cadastro ocorreu com sucesso
	2	Não foi possível enviar para o banco de dados
	3	Retorna erros para o usuário
	4	Retorna erros para o usuário

Fonte: Elaborada pelo autor (2025).

Tabela 14 - UC08: Atribuir Serviços.

Caso de uso	Condição de entrada	Classes válidas	Classes inválidas
UC08 - Atribuir Serviços	Atribuir serviços predefinidos	1 Navegue até o menu do painel para página "Atribuir Serviços"	3 Não escolher nenhum serviço
		2 Escolher mais de um serviço	

Fonte: Elaborada pelo autor (2025).

Tabela 15 - Resultados UC08: Atribuir Serviços.

Condição de Entrada	Classes	Resposta
Selecionar as opções	1	Acessa a página como o esperado
	2	Envia para o banco de dados
	3	Não retorna erros

Fonte: Elaborada pelo autor (2025).

Tabela 16 - Porta 5001: Micro Vacinas.

Porta	Condição de entrada	Classes válidas	Classes inválidas
5001- Micro Vacinas	Inserir a chave correta para disparar os lembretes	1. Caso o usuário tenha acesso a chave o serviço efetuará disparos	2. Digitar qualquer caractere
			3. Tentativa de envio sem chave
			4. Utilizar a chave com caracteres minúsculos

Fonte: Elaborada pelo autor (2025).

Tabela 17 - Resultados Porta 5001: Micro Vacinas.

Condição de Entrada	Classes	Resposta
Inserir a chave correta	1	Serviço funcionando como esperado
	2	Retorna que a senha está errada
	3	Pede para digitar a senha
	4	Retorna que a senha está errada

Fonte: Elaborada pelo autor (2025).

Tabela 18 - Porta 5002: Micro Remédios.

Porta	Condição de entrada	Classes validas	Classes inválidas
5002- Micro Remédios	Inserir a chave correta para disparar os lembretes	1. Caso o usuário tenha acesso a chave os e-mails serão disparados	2. Digitar qualquer caractere
			3. Tentativa de envio sem chave
			4. Utiliza a chave com caracteres minúsculos

Fonte: Elaborada pelo autor (2025).

Tabela 19 - Resultados Porta 5002: Micro Remédios.

Condição de Entrada	Classes	Resposta
Inserir a chave correta	1	Serviço funcionando como esperado
	2	Retorna que a senha está errada
	3	Pede para digitar a senha
	4	Retorna que a senha está errada

Fonte: Elaborada pelo autor (2025).

Tabela 20 - Porta 5003: Micro Imagens.

Porta	Condição de entrada	Classes validas	Classes invalidas
5003- Micro Imagens	Inserir a chave correta para disparar os lembretes	1. Navegar até o grupo de imagens e inserir a chave	2. Digitar qualquer carácter.
			3. Tentativa de envio sem chave.
			4. Utiliza a chave com caracteres minúsculos
			5. Efetuar o disparo utilizando mais grupos

Fonte: Elaborada pelo autor (2025).

Tabela 21 - Resultados porta 5003: Micro Imagens.

Condição de Entrada	Classes	Resposta
Escolher o grupo desejado e inserir a chave corretamente	1	Serviço funcionando como esperado
	2	Retorna senha incorreta
	3	Pede para digitar a senha
	4	Retorna que a senha está errada
	5	Erro ao tentar disparar mais de um grupo

Fonte: Elaborada pelo autor (2025).

## 7. CONCLUSÕES E TRABALHOS FUTUROS

### 7.1 CONCLUSÕES

Tendo em vista o objetivo principal do projeto que é enviar mensagens para os tutores a fim de prevenir zoonoses e divulgar outros serviços disponibilizados por uma clínica ou petshop, pode-se concluir que os serviços de mensageria ofertados como, lembrete de vacinas, remédios e imagens estão sendo entregues corretamente.

Devido à falta de implementação em testes reais e ao tempo que estes testes poderiam levar, não foi possível realizá-los de forma eficaz a comprovar limitações e benefícios da aplicação e seus serviços, por ser necessário ter um cadastro completo e dados anteriores e pós aplicação.

Se tratando do envio de vacinas e remédios vencidos, uma conta comum no e-mail consegue enviar até 500 e-mails por dia, podendo suprir uma pequena e média empresa tranquilamente. Também é possível adequar os serviços de acordo com a usabilidade de outros a fim de retirar tutores que não frequentam mais o local.

Por outro lado, ao enviar imagens para todos os tutores cadastrados no banco de dados, mesmo avaliando o tempo de serviço utilizado pelos usuários, ainda sim é necessário adequar o serviço pois quando passa do limite de conta comum, outros usuários não receberam as imagens disparadas.

Para solucionar o problema de alguns usuários receberem e outros não, o estabelecimento pode utilizar uma conta empresa no Gmail, onde o limite passa a ser de 2000 mensagens diárias. Caso ainda assim não supra a demanda, pode-se refinar ainda mais os microsserviços.

## 7.2 TRABALHOS FUTUROS

O trabalho visa melhorar a comunicação entre os serviços e os tutores, a fim de ser mais um método além dos convencionais para prevenção de zoonoses, mas buscando melhorias na comunicação e divulgação, existem outros meios que podem ser mais adequados em termos de usabilidade.

Primeiramente a implementação de um novo serviço de mensageria via Whatsapp, por ser uma rede mais usada pelos usuários, sendo uma das redes sociais mais utilizadas no mundo, tendo um custo para realizar o envio de mensagens.

Apesar da mensagem ser enviada via e-mail para o usuário, é necessário adequar essa mensagem dependendo do estabelecimento que aderir o uso dos microsserviços a fim de ser mais social e chamar a atenção do usuário.

Sabendo das limitações impostas nas contas de e-mail é necessário refinar os microsserviços, podendo monitorar os envios através de planilhas que podem ser geradas para verificar quem recebeu ou não a mensagem. Com isso pode-se criar um contador para recomençar o enviar das mensagens de onde parou um outro dia.

Realizar testes de forma online, utilizando outras formas além da rede local como AWS ou outros provedores que possuem suporte ao Docker. Juntamente com parceiros que tenham interesse em testar a aplicação para comprovar a eficácia do envio de notificações aos usuários.

Quanto à parte de atribuição de serviços (UC08), podem ser feitos microsserviços a fim de lembrar os tutores de reutilizar algum serviço que já não utilizam há algum tempo.

Mesmo se o usuário optar por utilizar de forma local, é necessário definir um serviço de backup regular e automático para que não tenha nenhum tipo de perda de dados. Sendo também necessário revisar a segurança da aplicação através de testes para evitar furto de dados ou inserção.

## REFERÊNCIAS

Alves, A. De J.; Silva, A. Dos S. **Desenvolvimento de Sistema Web para a Associação de Transporte Urbano de Vitória da Conquista (ATUV)**. Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas de Informação (SBSI Estendido 2022). Anais...Sociedade Brasileira de Computação (SBC), 2022. Disponível em: [https://sol.sbc.org.br/index.php/sbsi\\_estendido/article/view/21601](https://sol.sbc.org.br/index.php/sbsi_estendido/article/view/21601). Acesso em: 9 jan. 2025

Anjos, T. E. G. D. ., Silva, L. P. de F. e ., Silva, E. A. T. D. ., Correia, L. F. S. ., Da Silva, C. F. ., Ostermayer, D., Procópio, D. L. ., Paiva, I. de V. ., Cabral, L. M. R. ., Santos, M. R. ., & Silva, R. E. . (2025). **COBERTURA VACINAL DE CÃES E GATOS E SEUS IMPACTOS NA SAÚDE PÚBLICA**. *Revista CPAQV - Centro De Pesquisas Avançadas Em Qualidade De Vida* , 17(1). Disponível em: <https://doi.org/10.36692/V17N1-27R>. Acesso em: 20 jan. 2025.

AWS. **Qual a diferença entre Docker e VM?** 2024. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-docker-vm/>. Acesso em: 22 fev. 2025.

Butantan. **Mortes por raiva humana no país reforçam importância da vacina antirrábica distribuída pelo Butantan**. 2024. Disponível em: <https://butantan.gov.br/noticias/mortes-por-raiva-humana-no-pais-reforcam-importancia-da-vacina-antirrabica-distribuida-pelo-butantan>. Acesso em: 4 fev. 2025.

Castro Alves, A. **Febre Maculosa tem alto índice de letalidade - Fundação Hospitalar do Estado de Minas Gerais - FHEMIG**. Disponível em: <https://fhemig.mg.gov.br/noticias/2664-febre-maculosa-tem-alto-indice-de-letalidade>. Acesso em: 20 jan. 2025.

Costa, Holliver de Oliveira (ES). **Microserviços e orquestração de contêineres: uma abordagem prática**. 2023. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/30478>. Acesso em: 14 out. 2024.

Donis, J. L., e Silva, E. R., & Afonso, M. V. R. (2023). **Avaliação de cuidados básicos realizados em cães e gatos domiciliados**. *CONTRIBUCIONES A LAS CIENCIAS SOCIALES*, 16(11), 28322–28336. Disponível em: <https://doi.org/10.55905/revconv.16n.11-215>. Acesso em: 23 fev. 2025.

E. Crawley, B. Cameron, D Selva. " **System architecture: strategy and product development for complex systems**."(2015).

Ferreira, Lúria Stéfany Alves; Pereira, Saulo Gonçalves; Vieira, Adriele Laurinda Silva. **Análise da percepção da importância da vacinação contra doenças virais pelos tutores de cães em uma clínica veterinária no município de Patos de Minas-MG.** GETEC – Gestão, Tecnologia e Ciências. 2023. Disponível em: <https://www.revistas.fucamp.edu.br/index.php/getec/article/view/3004>. Acesso em: 08 jan. 2025.

Fowler, Martin. **Microservices.** 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acesso em: 14 out. 2024.

Freire, M. N. da C., Vêras, M. L. M., Silva, A. R. da, Carvalho, K. da S., & Silva, T. H. da. (2024). **Transmissão de zoonoses e os fatores de riscos na Região de Sousa-PB.** REVISTA DELOS, 17(61). <https://doi.org/10.55905/rdelosv17.n61-124>. Acesso em: 8 jan. 2025.

Garcia, W.; Pereira, F. C. Docker – **CONTAINERS NÃO SÃO VM's. SEMINÁRIO DE TECNOLOGIA GESTÃO E EDUCAÇÃO**, v. 1, n. 2, p. 5–10, 2019. Disponível em: <https://raam.alcidesmaya.com.br/index.php/SGTE/article/view/21>. Acesso em 14 mar. 2025

Hubálek, Zdenek. Emerging human infectious diseases: anthroponoses, zoonoses, and sapronoses. *Emerging Infectious Diseases*, v. 9, n. 3, p. 403-404, 2003. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC2958532/pdf/02-0208.pdf>. Acesso em: 14 mar. 2025

Kautz, O., Rumpe, B., & Wachtmeister, L. (2022). **Diferenciação semântica de diagramas de casos de uso.** *J. Object Technol.* , 21, 3:1-14. <https://doi.org/10.5381/jot.2022.21.3.a5>. Acesso em: 24 mar. 2025

Klensin, J. **Estrutura SMTP.** 2008. Disponível em: [https://www.researchgate.net/figure/SMTP-model-Klensin-J-2008\\_fig4\\_282817937](https://www.researchgate.net/figure/SMTP-model-Klensin-J-2008_fig4_282817937). Acesso em: 08 jan. de 2025.

Lucena, Adelmo. **Pesquisa revela que 87% dos tutores não vacinam animais domésticos no Nordeste.** *Diário de Pernambuco*, 4 nov. 2024. Disponível em: <https://www.diariodepernambuco.com.br/noticia/vidaurbana/2024/11/pesquisa-revela-que-87-dos-tutores-nao-vacinam-pets-no-nordeste.html>. Acesso em: 12 jan. de 2025.

Marques, S. **O que é escalabilidade de software e por que ela é tão importante?** Disponível em: <https://uds.com.br/blog/o-que-e-escalabilidade-de-software/>. Acesso em: 23 fev. 2025.

MINISTÉRIO DA SAÚDE. **VIGILÂNCIA, PREVENÇÃO E CONTROLE DE ZOOSE NORMAS TÉCNICAS E OPERACIONAIS** MINISTÉRIO DA SAÚDE. 2016. Disponível em: [https://bvsms.saude.gov.br/bvs/publicacoes/manual\\_vigilancia\\_prevencao\\_controle\\_zoonoses.pdf](https://bvsms.saude.gov.br/bvs/publicacoes/manual_vigilancia_prevencao_controle_zoonoses.pdf) Acesso em: 15 fev. 2025

Morais, Gabriela Duarte de. **Percepção dos tutores de cães e gatos atendidos no Hospital Veterinário da Universidade Federal da Paraíba quanto à importância da vacinação.** 2022. Trabalho de Conclusão de Curso (TCC) – Universidade Federal da Paraíba, Areia, 2022. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/25657>. Acesso em: 08 jan. 2025.

Newman, S. **Criando Microsserviços – 2ª edição.** [S. l.]: Novatec Editora, 2022. Oliveira Gomes, L. G.; Oliveira Gomes, G.; Drielle Fodra, J.; Massabni, A. C. **Zoonoses: as doenças transmitidas por animais.** Revista Brasileira Multidisciplinar, v. 25, n. 2, p. 158-174, 2022. DOI: 10.25061/2527-2675/ReBraM/2022.v25i2.1261. Acesso em: 08 jan. de 2025

Oracle. **O que é Docker? 2024.** Disponível em: <https://www.oracle.com/br/cloud/cloud-native/container-registry/what-is-docker/>. Acesso em: 9 fev. de 2025.

Pimenta, M. L. **Remédio para carrapato dura quanto tempo?** Disponível em: <https://www.patasdacasa.com.br/noticia/remedio-para-carrapato-dura-quanto-tempo>. Acesso em: 5 jan. 2025.

Pires, Felipe Alves De Oliveira; Corrêa, Fabrício Gonçalves. **Relevância e alcance dos protocolos de vacinação em cães: estudo de caso da incidência de doenças infecciosas em cães no hvu-unicep: cinomose, parvovirose e leptospirose.** Disponível em: [http://faef.revista.inf.br/imagens\\_arquivos/arquivos\\_destaque/DzG4NHIXfcqNtuJ\\_2020-6-18-8-54-2.pdf](http://faef.revista.inf.br/imagens_arquivos/arquivos_destaque/DzG4NHIXfcqNtuJ_2020-6-18-8-54-2.pdf). Acesso em: 08 de fev. 2025.

Rahman, M., Sobur, M., Islam, M., Levy, S., Hossain, M., Zowalaty, M., Rahman, A., & Ashour, H. (2020). **Zoonotic Diseases: Etiology, Impact, and Control. Microorganisms**, 8. . Disponível em: <https://doi.org/10.3390/microorganisms8091405>. Acesso em: 8 jan. 2025.

Rathore, M., & Bagui, S. (2024). **MongoDB: Meeting the Dynamic Needs of Modern Applications.** Encyclopedia. Disponível em: <https://doi.org/10.3390/encyclopedia4040093>. Acesso em: 24 mar. 2025.

Reisswitz, José Mário. **Implementação de uma plataforma backend para integração com protocolo OCPP.** 2024. Monografia (Bacharelado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Instituto de Informática, Porto Alegre, 2024. Disponível em: <https://lume.ufrgs.br/handle/10183/279133>. Acesso em: 08 jan. 2025.

Richardson, Chris. **Microservices:Decomposição de Aplicações para Implantação e Escalabilidade.** 2014. Disponível em: <https://www.infoq.com/br/articles/microservices-intro>. Acesso em: 15 out. 2024.

Sartori, Luciana. **Desafios e estratégias no combate à resistência antimicrobiana na clínica veterinária: impactos na saúde animal e humana.** Revista CFMV, 27 set. 2024. Disponível em: <https://revista.cfmv.gov.br/desafios-e-estrategias-no-combate-a-resistencia-antimicrobiana-na-clinica-veterinaria-impactos-na-saude-animal-e-humana/>. Acesso em: 4 jan. 2025.

Silva, Jonas Ayres da. μIF: **Arquitetura baseada em microsserviços para gestão escolar.** 2021. Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Restinga, 2021. Disponível em: <https://dspace.ifrs.edu.br/xmlui/handle/123456789/1369>. Acesso em: 08 fev. 2025.

Silva, Marcelo Vinícius Pereira; Lima, Mariana Aisla Pereira; Rocha, Isaac Manoel; Santos, Soraia Maria Pereira Gualda; Aguirre, André De Abreu Rangel; Terassini, Flávio Aparecido. **Febre maculosa no século XXI: panorama da mortalidade em um contexto nacional.** Acervo Mais, 2024. Disponível em: <https://acervomais.com.br/index.php/saude/article/view/18307/9447>. Acesso em: 4 jan. 2025.

Silva, Matheus Eduardo Rodrigues da. **EXPOMAP: Uma aplicação de divulgação de eventos agropecuários na região Nordeste.** 2023. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal de Campina Grande, Campina Grande, 2023. Acesso em: 08 jan. 2025.

Ufla. **Parvovirose: pesquisa da UFLA detecta alteração cardíaca em cães com mais precisão.** Disponível em: <https://ufla.br/noticias/pesquisa/12632-parvovirose-pesquisa-da-ufla-detecta-alteracao-cardiaca-em-caes-com-mais-precisao>. Acesso em: 8 jan. de 2025.

Varella, Dr. Dráuzio; Fundação Oswaldo Cruz. **Febre maculosa brasileira.** 2018. Disponível em: <https://bvsmms.saude.gov.br/febre-maculosa-brasileira/>. Acesso em: 13 fev. 2025.

4LINUX. **O que é Docker?** 2024. Disponível em: <https://4linux.com.br/o-que-e-docker/#:~:text=O%20Docker%20foi%20desenvolvido%20em,grande%20demanda%20de%20m%C3%A1quinas%20virtuais>. Acesso em: 23 fev. 2025.

4LINUX. **O que é Container?** 2024. Disponível em: <https://4linux.com.br/o-que-e-container/> Acesso em: 23 fev. 2025.