

FACULDADE DE TECNOLOGIA DE ARAÇATUBA
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
FELIPE MURAMATSU GOBBO TEIXEIRA
GUSTAVO FREITAS OLIVEIRA

SISTEMA DE GESTÃO SILI PROJETADO PARA MONITORAR E
REGISTRAR DADOS SOBRE CASOS E FATORES DE LEISHMANIOSE EM
ARAÇATUBA-SP.

ARAÇATUBA - SP

2024

FACULDADE DE TECNOLOGIA DE ARAÇATUBA
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
FELIPE MURAMATSU GOBBO TEIXEIRA
GUSTAVO FREITAS OLIVEIRA

**SISTEMA DE GESTÃO SILI PROJETADO PARA MONITORAR E
REGISTRAR DADOS SOBRE CASOS E FATORES DE LEISHMANIOSE EM
ARAÇATUBA-SP.**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Araçatuba, do Centro Estadual de Educação Tecnológica Paula Souza, como requisito parcial para conclusão do curso de Tecnologia em Análise e Desenvolvimento de Sistemas sob a orientação da Prof. Me. Alexandre Marcelino da Silva.

ARAÇATUBA- SP

2024

FACULDADE DE TECNOLOGIA DE ARAÇATUBA
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
FELIPE MURAMATSU GOBBO TEIXEIRA
GUSTAVO FREITAS OLIVEIRA

**SISTEMA DE GESTÃO SILI PROJETADO PARA MONITORAR E
REGISTRAR DADOS SOBRE CASOS E FATORES DE LEISHMANIOSE EM
ARAÇATUBA-SP.**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Araçatuba, do Centro Estadual de Educação Tecnológica Paula Souza, como requisito parcial para conclusão do curso de Tecnologia em Análise e Desenvolvimento de Sistemas avaliado pela banca examinadora.

Banca composta pelos professores:

Prof. Me. Alexandre Marcelino da Silva
Orientador-Fatec-Araçatuba

Profa. Me. Lucilena de Lima
Avaliador 1-Fatec-Araçatuba

Prof. Saulo Felício Fernandes Zambotti
Avaliador 2-Fatec-Araçatuba

ARAÇATUBA- SP

2024

Dedicamos à nossa família, amigos e a todos os professores.

AGRADECIMENTOS

A primeiramente a Deus, por ter nos dado saúde e força para superar as dificuldades.

A universidade, que nos desafia e nos ajudou nessa caminhada, a nos tornar pessoas melhores e profissionais qualificados.

Para nossos familiares, agradecemos pela motivação, encorajamento, apoio e compreensão pela ausência enquanto nos dedicamos em nossos estudos.

É com enorme gratidão, agradecemos a todos os professores que nos ajudaram nessa jornada, em especial ao nosso professor e orientador, Alexandre Marcelino, o nosso professor, Euclides Teixeira e por fim, Renata de Freitas, obrigado por todo apoio, orientação, paciência e disposição, vocês foram essenciais para concluir a nossa trajetória de estudos na faculdade, e concluir esse projeto.

Aos nossos amigos do grupo de estudo, nossa gratidão pelos momentos felizes, pelo constante suporte e pelo apoio mútuo ao longo desta jornada, em especial a Ana Lúcia Zago da Silveira Berte, Fernando Garcia dos Santos, Marjory Mineko Washio e Simone Leite da Silva, que contribuíram com esse projeto.

Nossa enorme gratidão, a toda equipe do Centro de Controle de Zoonoses de Araçatuba-sp, foram fundamentais para criação desse projeto, nosso muito obrigado.

“Porque dEle e por Ele, e para Ele são todas as coisas; glória, pois, a Ele eternamente. Amém”.

Romanos 11:36

RESUMO

A leishmaniose é uma doença causada pela infecção por protozoários do gênero *Leishmania*, que é transmitido para o ser humano pela picada de insetos vetores, conhecidos pelo nome flebotomíneos ou popularmente “mosquito-palha”. Sua transmissão pode ocorrer de duas formas: Leishmaniose cutânea e Leishmaniose visceral. A leishmaniose visceral é a mais perigosa, pois afeta órgãos internos e que pode levar à morte em 90% dos casos, se não for tratada. A doença afeta, em média, cerca de 3500 pessoas anualmente no Brasil e nos últimos anos, a letalidade vem aumentando, em 2010 era de 6,9% e em 2020 atingiu 9,5%. A região de Araçatuba-sp, vem registrando um aumento de mortes, em 2023, Araçatuba registrou mais da metade de casos de mortes no estado de São Paulo, de sete mortes, quatro foram na região noroeste paulista, segundo dados do Centro de Vigilância Epidemiológica (CVE) estadual. Este estudo teve como objetivo desenvolver um sistema mobile para gerenciar os registros de casos de leishmaniose humana entre profissionais epidemiológicos do centro de controle de zoonoses de Araçatuba, São Paulo, oferecendo suporte e implementação ao aplicativo SILI. A metodologia deste estudo envolveu uma revisão bibliográfica detalhada sobre a leishmaniose, além do uso de ferramentas de desenvolvimento mobile. O objetivo foi desenvolver um aplicativo que facilite a visualização e o controle das informações fornecidas pelos agentes epidemiológicos que atuam na detecção e tratamento da doença. O desenvolvimento do sistema foi realizado utilizando a linguagem JavaScript e o framework React Native. A interface do aplicativo usa como referência a ficha que os agentes usam no dia a dia, com intuito de ajudar na eficiência dos registros dos dados. Ao final do preenchimento, será gerado um relatório em PDF com todas as informações coletadas. Com essa plataforma digital, espera-se que os profissionais, ao adotarem o aplicativo móvel SILI, gerenciem as informações coletadas na região com mais precisão, contribuindo para a saúde pública.

Palavras Chaves: Leishmaniose; Zoonoses; Desenvolvimento web; Saúde pública.

ABSTRACT

Leishmaniasis is a disease caused by infection with protozoa of the genus *Leishmania*, transmitted to humans through the bite of vector insects known as sandflies or, popularly, “straw mosquitoes.” Transmission can occur in two forms: Cutaneous Leishmaniasis and Visceral Leishmaniasis. Visceral leishmaniasis is the more dangerous form, as it affects internal organs and can lead to death in 90% of cases if untreated. The disease affects an average of about 3,500 people annually in Brazil, and in recent years, the lethality rate has been increasing—from 6.9% in 2010 to 9.5% in 2020. The Araçatuba region in São Paulo state has been recording a rise in deaths, with more than half of São Paulo state's leishmaniasis-related deaths occurring there in 2023. Of the seven deaths, four were in the northwest region, according to data from the State Center for Epidemiological Surveillance (CVE). This study aimed to develop a mobile system to manage human leishmaniasis case records among epidemiological professionals at the zoonosis control center in Araçatuba, São Paulo, providing support and implementation for the SILI application. The methodology included a detailed literature review on leishmaniasis, as well as the use of mobile development tools. The goal was to create an application that facilitates the visualization and control of information provided by epidemiological agents involved in the detection and treatment of the disease. The system was developed using JavaScript and the React Native framework. The app's interface is based on the daily-use form filled out by agents, aiming to improve the efficiency of data recording. At the end of data entry, a PDF report with all collected information is generated. This digital platform is expected to enable professionals to manage the collected data more accurately when adopting the SILI mobile application, contributing to public health.

Keywords: Leishmaniasis; Zoonoses, Mobile development; Public health.

LISTA DE FIGURAS

Figura 1: Diagrama de Caso de Uso Interface Inicial	29
Figura 2: Diagrama de Caso de Uso Interface Relatório de Casos de Leishmaniose	30
Figura 3: Diagrama de Caso de Uso Função de Compartilhamento e Impressão	31
Figura 4: Diagrama de Contexto	32
Figura 5: Diagrama de Seqüência do Caso de Uso Interface Inicial	32
Figura 6: Diagrama de Seqüência do Caso de Uso Interface Relatório de Casos de Leishmaniose	33
Figura 7: Diagrama de Seqüência do Caso de Uso Função de Compartilhamento e Impressão	33
Figura 8: Diagrama de Atividades do Caso de Uso Interface Inicial	34
Figura 9: Diagrama de Atividades do Caso de Uso Interface Relatório de Casos de Leishmaniose	35
Figura 10: Diagrama de Atividades do Caso de Uso Função de Compartilhamento e Impressão	36
Figura 11: Diagrama de classes	37
Figura 12: Diagrama de Estrutura de Dados (DED)	38
Figura 13: Diagrama Entidade Relacionamento (DER)	38
Figura 14: Dicionário de Dados – HomeScreen.js	39
Figura 15: Dicionário de Dados – RelatorioDeCasos.js	40
Figura 16: Dicionário de Dados – PdfCode.js	41
Figura 17: Tela de Login	42
Figura 18: Relatório de Cadastro	43
Figura 19: Tela do PDF com os dados inseridos	46
Figura 20: Tela de Código Home	48
Figura 21: Tela de Dados do Grupo	49
Figura 22: Tela de Código Dados do Morador	50

Figura 23: Tela de Código Dados Gerais do Animal	51
Figura 24: Tela de Código Dados de Coleta do Animal	52
Figura 25: Tela de Código Definição de Data e Hora Automáticas no Arquivo PDF	53
Figura 26: Tela de Código Gerar Arquivo PDF	54

LISTA DE TABELAS

Tabela 1. RF 1 – Exibir Interface do Usuário	22
Tabela 2. RF 2 – Gerar Relatório	24
Tabela 3. RF 3 – Navegar entre Interfaces	25
Tabela 4. RF 4 – Validar Dados	25
Tabela 5. RNF 01 – Usabilidade Satisfatória	26
Tabela 6. RNF 02 – Manutenção Eficiente	26
Tabela 7. RNF 03 – Disponibilidade	26
Tabela 8. RNF 04 – Compatibilidade	27
Tabela 9. Lista de Casos de Uso	28
Tabela 10. Saídas de Caso de Uso	28

SUMÁRIO

1. INTRODUÇÃO	12
2. REVISÃO DA LITERATURA	14
2.1 CONTEXTUALIZAÇÃO	14
2.2 Tecnologias utilizadas	16
2.2.1 Conceitos sobre a linguagem JavaScript	16
2.2.2 Conceitos o framework React Native	19
3. METODOLOGIA DA PESQUISA	20
3.1 LEVANTAMENTO DE REQUISITOS	21
3.1.1 Requisitos Funcionais:	21
3.2 ANÁLISE E PROJETO DO SISTEMA	26
3.2.1 Lista de Casos de Uso	26
3.2.2 Diagramas de Casos de Uso	28
3.2.3 Diagramas de Contexto	30
3.2.4 Diagrama de Seqüência	31
3.2.5 Diagrama de Atividades	33
3.2.6. Diagrama de Classes	36
3.2.7. DED	36
3.2.8. DER	37
3.2.9 Dicionário de Dados do Aplicativo	38
3.3 DESENVOLVIMENTO DO PROJETO	40
3.3.1 Protótipo	40
3.3.2 Código Mobile	46

4.CONCLUSÃO	53
. TRABALHOS FUTUROS	53

1. INTRODUÇÃO

Com o aumento dos desafios no controle de casos de leishmaniose em Araçatuba-SP, a Faculdade de Tecnologia do Estado de São Paulo (Fatec), em parceria com o Centro de Controle de Zoonoses do município, deu início a uma parceria de projeto tecnológico: um aplicativo móvel que promete transformar a forma como os agentes epidemiológico registram e monitoram dados no campo.

O propósito deste projeto é ajudar nas atividades, registro e fácil controle da doença, para isso, o desenvolvimento de um aplicativo móvel. O intuito, como já mencionado, é facilitar. Conquanto, foi utilizado como referência, a ficha de coleta de dados, uma vez que já faz-se uso pelos agentes de saúde.

Com isso, a ficha física deixará de ser utilizada, passando-se a utilização apenas do aplicativo, tornando desnecessário que os agentes escrevam, dessa forma, passando apenas a digitar as informações específicas, tais como o nome do morador, e também, selecionar opções pré-determinadas que são padronizadas para o controle da leishmaniose, evitando duplicidade de informações e agilizando as atividades de campo.

Uma solução digital otimiza processos e tem um ganho alarmante na produtividade, além de fácil controle dos dados e gerenciamento das informações coletadas. Nesse viés, o aplicativo permite o registro de informações e a geração de relatórios em PDF a partir de dados preenchidos pelo usuário. Essa funcionalidade promove a praticidade na coleta, documentação e compartilhamento dos dados de maneira eficiente.

A interface do aplicativo inclui um formulário intuitivo para o usuário registrar dados essenciais de cada visita, para o monitoramento e controle da leishmaniose, como nome do animal, número de registro, idade, sintomas, e histórico clínico. Esse método tem se mostrado eficaz para sistematizar dados e promover o controle epidemiológico da leishmaniose (SANTOS et al., 2020).

Ao concluir o preenchimento das informações, o usuário pode revisar todos os detalhes antes de finalizar. A coleta de dados por aplicativos móveis é amplamente reconhecida por sua capacidade de reduzir erros e permitir a padronização de dados em saúde pública (GARCIA et al., 2021). O uso de PDFs para registro e compartilhamento de informações padronizadas é amplamente utilizado em ambientes de saúde pública (JONES, 2019).

A funcionalidade implementada, organiza automaticamente as informações inseridas no formulário em um layout acessível e estruturado. Após a geração do relatório, o documento PDF pode ser salvo no próprio dispositivo móvel ou compartilhado diretamente pelo aplicativo, promovendo a integração com outras plataformas e agilizando a continuidade do acompanhamento clínico (PEREIRA e LIMA, 2020).

O aplicativo se mostra um recurso útil para o controle da leishmaniose, permitindo uma coleta e uma análise simplificada dos dados registrados. O uso de tecnologias digitais, especialmente em zoonoses, é incentivado pela Organização Mundial da Saúde (OMS), considerando o impacto positivo que pode ter na precisão e na acessibilidade dos dados (WHO, 2022).

Pensando em primeira instância, foi tomada a decisão de não fazer mudanças das quais precisam de um determinado tempo de adaptação e treinamento de alguns setores do Centro de Controle de Zoonoses de Araçatuba. Dessa forma, o protótipo visa melhorar a coleta de dados dos agentes epidemiológicos, que visitam as casas dos moradores, para com o objetivo de detectar a doença em animais e coletar dados, pois, ao passar alguns dias de experiência em campo, com esses agentes, foi identificado a dificuldade do acesso à internet em alguns locais e da complexidade do preenchimento da ficha em papel.

Tal aplicativo foi desenvolvido para que os agentes tenham familiaridade com o fluxo de preenchimento, que de certa maneira, é como se os mesmos estivessem anotando no papel, mas de uma forma mais ágil e

prática. Pensando nos desafios dos agentes epidemiológicos e na introdução por etapas na melhoria do processo e na coleta de dados de ponta a ponta, o aplicativo não necessita de acesso a internet, funcionando offline, com isso, em locais onde a conexão é instável ou inexistente, não irá limitar o trabalho, de modo que garanta a continuidade do monitoramento.

Faz-se imprescindível ressaltar que o protótipo foi pensado, para que não dependesse de uma infraestrutura de banco de dados nesse momento, diminuindo custos de implementação, uma vez que também, serão trabalhados dados efêmeros processados em memória, ou seja, dados transitórios. Em relação aos arquivos PDFs que serão gerados após o preenchimento do formulário, ambos poderão ser armazenados localmente no dispositivo móvel ou em soluções de nuvem (ao exemplo do Google Drive), com a também possibilidade de serem impressos em momento oportuno (quando o dispositivo estiver conectado a uma impressora), ou até mesmo, compartilhados por intermédio de aplicativos de mensagem (ao exemplo do WhatsApp), para tornar-se possível a posterior visualização ao acessar a internet na própria sede do Centro de Controle de Zoonoses de Araçatuba, assim proporcionando uma gestão mais eficaz.

Pautando-se em um propósito específico, busca-se obter informações por meio de reuniões com os profissionais do Centro de Controle de Zoonoses do município, com o suporte de outros estudantes da faculdade e de professores universitários, que terão o papel de auxiliar na definição das necessidades que o sistema de gerenciamento deve atender para o usuário.

O principal objetivo deste estudo é desenvolver um aplicativo utilizando JavaScript, o framework React Native e o salvamento dos dados coletados a partir da geração de um documento em formato PDF. Esse trabalho, por sua vez, buscou ser objetivo, de fácil aplicação e alta praticidade, pensando em melhorias por etapas e aprimoramento conforme a necessidade dos stakeholders para o controle de casos de leishmaniose no município.

A abordagem adotada foi qualitativa, focada no desenvolvimento de uma plataforma online que atenda às necessidades dos profissionais do Centro de Controle de Zoonoses de Araçatuba, São Paulo. Para atingir seus objetivos, optou-se pela pesquisa exploratória, que busca proporcionar uma maior compreensão do problema. Para o entendimento da dificuldade dos agentes, também foi realizada pesquisa em campo, de modo, a entender os principais desafios de quem de fato fará uso da solução. Os procedimentos metodológicos começaram com uma pesquisa bibliográfica em artigos e materiais didáticos.

O estudo está organizado em cinco partes: Introdução, Revisão da Literatura, Metodologia da Pesquisa, Conclusão, Trabalhos Futuros e Referências.

2. REVISÃO DA LITERATURA

2.1 CONTEXTUALIZAÇÃO

A leishmaniose é uma doença infecciosa ocasionada pelo protozoário do gênero *Leishmania*. O homem adquire a doença através da picada de insetos, conhecidos como vetores ou flebotomíneos, e popularmente chamados de “mosquito-palha”. Dois modos de transmissão são distinguidos – cutâneo e visceral. O visceral é o mais perigoso. É uma doença endêmica em cerca de 98 países, com mais de 350 milhões de pessoas em risco de contrair a doença. Segundo a Organização Mundial da Saúde (OMS), estima-se que haja cerca de 1,5 a 2 milhões de novos casos por ano, com aproximadamente 20 a 30 mil mortes.

No Brasil, o principal vetor é a *Lutzomyia longipalpis*, popularmente conhecido como mosquito-palha, dominante em regiões tropicais e subtropicais, com uma crescente taxa de letalidade (Prado et al., 2024).

A enfermidade tem algumas diferentes evoluções clínicas, tais como, a cura natural, formas oligossintomáticas, assintomáticas, até manifestações graves, podendo alcançar letalidade entre 10% e 98%.

Conforme o Centro Estadual de Vigilância em Saúde (2023), “Nas Américas, é endêmica em 13 países, sendo o Brasil responsável pelo maior número de notificações, cerca de 97%, em 2020” (p. 1). Estima-se que a leishmaniose visceral esteja presente em todas as regiões do território brasileiro, tendo se adaptado ao contexto das zonas urbanas e periurbanas em resposta ao crescimento desordenado da presença humana e às condições de vida inadequadas. Isso traz consequências significativas para a dinâmica e a expansão geográfica dessa enfermidade (Vinicius Oliveira et al., 2022).

No estado de São Paulo, o registro histórico de casos de leishmaniose visceral começa em 1999, com a entrada da doença pela região oeste, mais especificamente no município de Araçatuba e sua subsequente expansão para outras áreas. Uma compreensão mais aprofundada desse processo e a identificação dos fatores determinantes podem facilitar a implementação de ações de vigilância e controle eficazes, evitando que a leishmaniose visceral se dissemine por todo o território paulista ou, pelo menos, reduzindo a velocidade de sua expansão.

No período de 2014 a 2019, a região de Araçatuba registrou 23% dos casos notificados em todo o estado, além disso, o município de Araçatuba registrou 7% destes casos. No ano de 2023, de acordo com os dados do Centro de Vigilância Epidemiológica (CVE) estadual, a região de Araçatuba registrou 4 mortes por leishmaniose, mais da metade de todo o estado de São Paulo.

Considerando esses pontos, o uso da tecnologia tem um papel fundamental no enfrentamento a essa doença, contribuindo de forma eficiente na coleta de dados e monitoramento, de forma que permite acompanhar a evolução da doença, identificando áreas de riscos e tomar medidas estratégicas com base em uma coleta de dados mais ágil.

Com isso, a tecnologia pode contribuir para a prevenção, o controle e a eliminação das doenças que afetam a população da região (Fredy Ovallos et al, 2019).

A coleta de dados e monitoramento da leishmaniose é essencial para

compreender a evolução da doença e adotar medidas de controle eficazes, contribuindo para a saúde pública e o bem-estar da população. (Severino, 2023)

Neste cenário, foi criado o SILI, o protótipo do aplicativo para auxiliar na coleta de dados e monitoramento de casos de leishmaniose no município de Araçatuba-SP, que por ventura, já vem passando por um cenário de desenvolvimento há alguns anos e por conseguinte, passou por transformações e adequações para o quadro atual, levando em consideração reuniões com o intuito de conhecer os determinados requerimentos dos stakeholder e também, a necessidades dos envolvidos. Esse sistema proporciona suporte eficaz aos profissionais de saúde que atuam no Centro de Controle de Zoonoses, viabilizando o monitoramento e o controle da leishmaniose animal. O aplicativo móvel desenvolvido como parte desse projeto representa uma valiosa ferramenta para os agentes epidemiológicos, permitindo assim, uma gestão mais precisa e prática da doença quando empregado pela instituição de saúde.

2.2 Tecnologias utilizadas

2.2.1 Conceitos sobre a linguagem JavaScript

O JavaScript é uma linguagem de programação de alto nível, muito popular na área do desenvolvimento, interpretada e orientada a objetos amplamente utilizada para criar páginas da web dinâmicas e interativas. Ela foi criada inicialmente pela Netscape em 1995 e atualmente é suportada por todos os principais navegadores. Além de seu uso no desenvolvimento web, o JavaScript também é amplamente utilizado para desenvolvimento back-end, aplicativos móveis e até mesmo para programação de dispositivos de Internet das Coisas (IoT) (Cunha, 2023). De acordo com a Pesquisa de Desenvolvedores do Stack Overflow de 2023, a linguagem é a mais popular no mundo entre os desenvolvedores.

A história do JavaScript remonta a 1995, quando Brendan Eich, então funcionário da Netscape, foi encarregado de desenvolver uma linguagem de script para a plataforma Netscape Navigator. Inicialmente chamada de Mocha, a linguagem foi renomeada para Live Script e finalmente para JavaScript. Em 1996, a Microsoft lançou seu próprio dialeto chamado JScript, enquanto a Netscape submeteu o JavaScript à Ecma International para padronização, resultando na especificação ECMAScript. Desde então o JavaScript passou por várias atualizações e evoluções, culminando na versão moderna conhecida como ECMAScript 6, que introduziu muitos recursos avançados à linguagem.

O JavaScript é uma linguagem de tipagem dinâmica, o que quer dizer que as variáveis não precisa ter um tipo específico declarado e podem mudar de tipo durante a execução do programa. O JavaScript é assíncrono por natureza, permitindo que operações demoradas, como requisições de rede, sejam executadas em segundo plano sem bloquear a interface do usuário. Outra característica importante é a capacidade de manipular o Document Object Model (DOM), permitindo a interação com os elementos de uma página web de forma dinâmica. Além de suportar a programação assíncrona, o JavaScript viabiliza operações como requisições de rede sem interromper a fluidez da execução. A linguagem disponibiliza recursos para a geração de funções, objetos, matrizes e gerenciamento de eventos, consolidando-se como uma base essencial no contexto do desenvolvimento web. (Justus Bogner e Manuel Merkel, 2022). A assincronia em JavaScript é fundamental para lidar com operações demoradas que podem bloquear a execução do código. Por meio de técnicas assíncronas, é possível realizar tarefas sem interromper o fluxo principal do programa. Isso é especialmente útil em casos como requisições de rede, operações de leitura/gravação de arquivos e outras atividades que demandam tempo. A capacidade de lidar com tarefas em segundo plano aumenta significativamente a responsividade e a eficiência das aplicações web, melhorando a experiência do usuário (Endler, 2023).

Uma das técnicas mais antigas para lidar com assincronia em JavaScript são os callbacks. Eles consistem em passar uma função como argumento para outra função, que será chamada quando a operação assíncrona for concluída. Embora sejam eficazes, os callbacks podem levar a problemas de legibilidade e a um fenômeno conhecido como callback hell, devido ao aninhamento excessivo de funções. Apesar disso, é importante compreender o funcionamento dos callbacks para entender boa parte dos conceitos assíncronos em JavaScript (Rocha, 2022). A evolução para técnicas como Promises e `async/await` contribuiu para tornar o código mais legível e gerenciável, mas o entendimento dos callbacks permanece relevante para uma base sólida de conhecimentos em JavaScript.

Outra abordagem importante no desenvolvimento em JavaScript é o uso de módulos. Em JavaScript, os módulos são uma forma de organizar o código em arquivos separados, permitindo a importação e exportação de funcionalidades entre eles. A declaração `'export'` é usada para disponibilizar uma função, classe ou variável para importada em outros arquivos, enquanto a declaração `'import'` é utilizada para trazer essas funcionalidades para o arquivo atual. Isso facilita a manutenção do código, evita a

repetição de lógica e promove a reutilização de componentes em diferentes partes de um projeto. Além disso, o uso de módulos ajuda a manter a clareza e organização do código, tornando-o mais escalável e legível (Buss, 2023).

Para garantir que todo o processo de desenvolvimento seja eficiente e produtivo, o uso de ferramentas de desenvolvimento em JavaScript é essencial. Ferramentas como editores de código, por exemplo, o Visual Studio Code, e sistemas de versionamento como o Git, desempenham papéis cruciais. Além disso, bundlers como o Webpack são utilizados para empacotar o código em um formato que o navegador possa entender. Essas ferramentas são fundamentais para aumentar a produtividade e a qualidade do desenvolvimento em JavaScript (Soares e Dadalto, 2024).

Faz-se também imperioso destacar ao desenvolvimento em JavaScript, a depuração de código. A depuração é essencial para identificar e corrigir erros, permitindo uma inspeção detalhada do código, incluindo a verificação de variáveis a pilha de chamadas e os breakpoints. Ferramentas como o debugger do Chrome e o Node.js Inspector são comumente utilizadas para depurar código JavaScript, auxiliando os desenvolvedores na identificação e correção de problemas (Faria, 2023). O uso dessas ferramentas com boas práticas de programação contribui significativamente para a criação de um código mais robusto e confiável.

Ao programar em JavaScript e seguir práticas de boa programação é uma recomendação fundamental para garantir que o código seja limpo, legível e de fácil manutenção. Isso inclui a utilização de nomes significativos para variáveis, funções e classes, e o seguimento de convenções de codificação como camelCase ou snakecase. A consistência no estilo de codificação em todo o projeto facilita a colaboração e compreensão do código por outros desenvolvedores (Buss, 2023). Além disso, essas práticas ajudam a manter a uniformidade e a padronização do projeto, evitando confusões e aumentando a eficiência do trabalho em equipe.

Para finalizar, a documentação do código JavaScript. Comentários bem escritos explicam a lógica por trás do código e evitam confusão para outros desenvolvedores que venham a trabalhar no mesmo projeto. A documentação adequada das funções, classes e bibliotecas utilizadas é fundamental para facilitar a manutenção e a colaboração. Ferramentas como o JSDoc podem ser extremamente úteis para garantir uma boa documentação do código, mantendo a clareza e compreensão (Siqueira et al., 2023). Uma documentação completa e bem estruturada complementa as práticas de codificação e fortalece o entendimento coletivo do projeto.

2.2.2 Conceitos o framework React Native

O React Native é um framework de código aberto desenvolvido pelo Facebook que permite o desenvolvimento de aplicativos móveis entre várias plataformas, utilizando a linguagem JavaScript. Com o React Native, é possível criar aplicativos nativos para iOS e Android a partir de um único código-fonte, proporcionando uma maior eficiência e economia de tempo para os desenvolvedores. O framework utiliza uma abordagem de desenvolvimento baseada em componentes, fornecendo uma interface de usuário responsiva e de alto desempenho (FREITAS, 2023).

O React Native é uma estrutura de desenvolvimento de aplicativos móveis que permite a criação de aplicativos nativos para iOS e Android usando JavaScript como linguagem de programação. Ele permite aos desenvolvedores compartilhar até 90% do código entre diferentes plataformas, proporcionando uma experiência de desenvolvimento eficiente e permitindo alcançar um público mais amplo. Com o React Native, é possível criar interfaces de usuário altamente responsivas e oferecer uma experiência de usuário mais suave.

O React Native oferece diversas vantagens em relação a outros frameworks de desenvolvimento mobile, como a capacidade de criar aplicativos multiplataforma com uma única base de código, o que proporciona economia de tempo e recursos. Além disso, o framework possui uma vasta comunidade de desenvolvedores ativos, o que significa amplo suporte e documentação disponível. Outro destaque é a facilidade de integração com bibliotecas e APIs externas, o que amplia as possibilidades de funcionalidades nas aplicações desenvolvidas com React Native (SANTANA JÚNIOR, 2024).

Uma das principais características do React Native é seu desempenho e performance superiores, possibilitando a construção de aplicativos rápidos e responsivos. Isso se deve ao fato de que o framework utiliza componentes nativos para renderização da interface do usuário, resultando em uma experiência mais fluída para o usuário final. Além disso, o React Native oferece a capacidade de realizar otimizações de código específicas para cada plataforma, garantindo um desempenho consistente em diferentes dispositivos móveis (Sarmiento, 2023).

3. METODOLOGIA DA PESQUISA

Neste estudo, tem como proposta um sistema mobile que irá auxiliar os profissionais epidemiológicos de Araçatuba, a monitorar casos de leishmaniose animal, em colaboração com o centro de controle de zoonoses e, desta forma, desenhar e classificar áreas de risco.

O problema a ser investigado requer uma abordagem qualitativa, pois aponta para o desenvolvimento de um protótipo de um software que atenda aos requisitos. Conforme observado por guerra, a pesquisa com Guerra (2013), a pesquisa qualitativa é uma tentativa de explicação ou compreensão da experiência conforme entendida pelos informantes e, portanto, não é limitada a um número ou estatística finita. Para alcançar os objetivos propostos, o estudo será exploratório, pois fornecerá insights mais profundos e claros do problema. Como observado por Gil citado em Oliveira (apud Oliveira, 2011), a pesquisa exploratória tem como objetivo desenvolver, esclarecer e modificar conceitos e idéias. Os procedimentos do estudo iniciaram-se com a pesquisa bibliográfica em artigos e apostilas, que é uma fonte de coleta de dados secundária sobre um tema, assunto ou problema, de acordo com Lakatos & Marconi e Cervo & Bervian (apud Oliveira, 2011).

3.1 LEVANTAMENTO DE REQUISITOS

3.1.1 Requisitos Funcionais:

Para garantir uma entrega alinhada com as necessidades dos usuários e requerimentos dos stakeholders a partir de reuniões, foi de extrema importância a realização de um amplo levantamento de requisitos. A fim de obter todos os dados necessários e realizar as tarefas esperadas e atribuídas de forma funcional, realizamos questionários e pesquisas com os usuários. Essa etapa foi crucial, pois nos permitiu compreender as necessidades e preferências dos mesmos. Posteriormente, projetamos e desenvolvemos essas funcionalidades e as fornecemos em uma plataforma mobile amigável. O sistema desenvolvido visa formar a satisfação do usuário garantindo que forneçamos as tarefas certas e adequadas ao uso.

Tabela 01. RF 1 – Exibir Interface do Usuário

Identificação	Nome
RF 01	Exibir Interface do Usuário
Descrição	
<p>O sistema deverá exibir:</p> <ol style="list-style-type: none"> 1. Interface Inicial (HomeScreen): <ol style="list-style-type: none"> 1.1 Exibir o logotipo do aplicativo centralizado na interface. 1.2 Exibir um botão "ENTRAR" que, quando clicado, navega para a interface de Relatório de Casos. 2. Interface de Relatório de Casos de Leishmaniose (RelatorioDeCasos): <ol style="list-style-type: none"> 2.1 Campos de Dados do Grupo: <ol style="list-style-type: none"> 2.1.1 GT: Um campo seletor (Picker) para escolher entre opções predefinidas. 2.1.2 GVE: Um campo seletor (Picker) com opções predefinidas. 2.1.3 ATL: Um campo seletor (Picker) para escolher entre opções predefinidas. 2.1.4 Área: Um campo seletor (Picker) para selecionar uma área específica. 2.1.5 Quadra: Um campo de entrada numérico para inserir o número da quadra (Obrigatório). 2.2 Campos de Dados do Morador: 	

- 2.2.1 CodEnd: Um campo de entrada numérico para o código do endereço (Obrigatório).
- 2.2.2 Logradouro: Um campo de entrada de texto para o nome da rua (Obrigatório).
- 2.2.3 N° do Imóvel: Um campo de entrada numérico para o número do imóvel (Obrigatório).
- 2.2.4 Situação do Imóvel: Um campo seletor (Picker) para escolher a situação do imóvel.
- 2.2.5 Nome: Um campo de entrada de texto para o nome do proprietário (Obrigatório).
- 2.2.6 Telefone: Um campo de entrada numérico para o telefone do proprietário (Obrigatório).

2.3 Campos de Dados dos Animais:

- 2.3.1 Nome: Um campo de entrada de texto para o nome do animal.
- 2.3.2 Registro: Um campo de entrada numérico para o registro do animal.
- 2.3.3 Idade: Um campo de entrada numérico para a idade do animal.
- 2.3.4 Sexo: Um campo seletor (Picker) para escolher o sexo do animal.
- 2.3.5 Raça: Um campo seletor (Picker) para escolher a raça do animal.
- 2.3.6 Cor: Um campo seletor (Picker) para escolher a cor do animal.
- 2.3.7 Peso: Um campo de entrada numérico para o peso do animal.
- 2.3.8 Adição de Múltiplos Animais: Permitir ao usuário especificar o número de animais e preencher os dados para cada um.
- 2.3.9 Campos de Dados de Coleta:

2.3.9.1 Autorização: Um campo seletor (Picker) para escolher se a coleta é autorizada.

2.3.9.2 Situação da Coleta: Um campo seletor (Picker) para escolher a situação da coleta.

2.3.9.3 Tamanho da Coleira: Um campo seletor (Picker) para escolher o tamanho da coleira.

2.3.9.4 Estado da Coleira: Um campo seletor (Picker) para escolher se a coleira é nova.

2.4 Funcionalidade de Geração de Relatório:

2.4.1 Botão "Gerar Relatório": Um botão que, ao ser clicado, gera um relatório PDF com todos os dados inseridos.

2.4.2 Mensagens de erro exibidas caso campos obrigatórios não estejam preenchidos:

2.4.2.1 "Erro. Quadra não preenchida."

2.4.2.2 "Erro. Código do Endereço não preenchido."

2.4.2.3 "Erro. Logradouro não preenchido."

2.4.2.4 "Erro. Número do Imóvel não preenchido."

2.4.2.5 "Erro. Nome do Proprietário não preenchido."

2.4.2.6 "Erro. Telefone do Proprietário não preenchido."

Fonte: Elaborado pelos autores

Tabela 02. RF 2 – Gerar Relatório

Identificação	Nome
RF 02	Gerar Relatório
Descrição	
<p>O sistema permitirá que o arquivo PDF gerado seja impresso ou compartilhado através de aplicativos nativos do dispositivo, como e-mail ou aplicativos de mensagens, dessa forma, gerando um documento PDF detalhado que inclua:</p> <ol style="list-style-type: none"> 1. Dados do grupo. 2. Dados do morador. 3. Dados dos animais. 4. Dados de coleta. 	

Fonte: Elaborado pelos autores

Tabela 03. RF 3 – Navegar entre Interfaces

Identificação	Nome
RF 03	Navegar entre Interfaces
Descrição	
<p>O sistema utilizará a biblioteca <i>react-navigation</i> para gerenciar a transição entre a interface inicial e a interface de Relatório de Casos de Leishmaniose, dessa forma, configurando a navegação de forma que seja intuitiva e responsiva para o usuário.</p>	

Fonte: Elaborado pelos autores

Tabela 04. RF 4 – Validar Dados

Identificação	Nome
RF 04	Validar Dados
Descrição	
<p>O sistema irá checar campos obrigatórios, dessa maneira, o mesmo realizará a função de:</p> <ol style="list-style-type: none"> 1. Implementar validações para garantir que todos os campos obrigatórios sejam preenchidos antes da geração do relatório; 2. Exibir mensagens de erro claras e específicas para orientar o usuário sobre quais campos precisam ser preenchidos. 	

Fonte: Elaborado pelos autores

3.1.2 Requisitos Não Funcionais:

Tabela 05. RNF 01 – Usabilidade Satisfatória

Identificação	Nome
RNF 01	Usabilidade Satisfatória
Descrição	
<p>O sistema deve ser projetado de modo que sua usabilidade seja simples e compreensível para um amplo espectro de usuários.</p>	

Fonte: Elaborado pelos autores

Tabela 06. RNF 02 – Manutenção Eficiente

Identificação	Nome
RNF 02	Manutenção Eficiente
Descrição	
O sistema deve ser programado de forma que seja adequadamente acessível para manutenção e atualização contínua.	

Fonte: Elaborado pelos autores

Tabela 07. RNF 03 – Disponibilidade

Identificação	Nome
RNF 03	Disponibilidade
Descrição	
O sistema deve estar permanentemente acessível e pronto para atender aos usuários a qualquer momento.	

Fonte: Elaborado pelos autores

Tabela 08. RNF 04 – Compatibilidade

Identificação	Nome
RNF 04	Compatibilidade
Descrição	
O sistema deve ser capaz de funcionar de maneira adequada em uma variedade de dispositivos.	

Fonte: Elaborado pelos autores

3.2 ANÁLISE E PROJETO DO SISTEMA

3.2.1 Lista de Casos de Uso

Tabela 9. Lista de Casos de Uso

Nº	CASO DE USO	DESCRIÇÃO	ENTRADA	SAÍDA
01	Acessar a Interface de Relatório de Casos de Leishmaniose.	Usuário abre o aplicativo e clica no botão "ENTRAR" na interface inicial.	Clicar no botão "ENTRAR".	Navegação para "RelatorioDeCasos".
02	Inserir Dados do Grupo.	Usuário preenche os campos de dados do grupo.	GT, GVE, ATL, Área, Quadra.	Dados armazenados.
03	Inserir Dados do Morador.	Usuário preenche os campos de dados do morador.	Código do Endereço, Logradouro, Nº do Imóvel, Situação do Imóvel, Nome, Telefone.	Dados armazenados.
04	Inserir Dados dos Animais.	Usuário preenche os campos de dados dos animais.	Nome, Registro, Idade, Sexo, Raça, Cor, Peso.	Dados armazenados.
05	Adicionar Múltiplos Animais.	Usuário especifica a quantidade de animais e preenche os dados para cada animal.	Número de animais, dados individuais de cada animal.	Dados armazenados.

06	Inserir Dados de Coleta.	Usuário preenche os campos de dados de coleta.	Autorização, Situação da Coleta, Tamanho da Coleira, Estado da Coleira.	Dados armazenados.
07	Gerar Relatório PDF.	Usuário clica no botão "Gerar Relatório" e gera o arquivo em formato PDF.	Dados de grupo, morador, animais e coleta preenchidos.	Msg1, Msg2, Msg3, Msg4, Msg5.
08	Compartilhar Relatório PDF.	Usuário compartilha o relatório em arquivo PDF gerado.	Arquivo PDF gerado.	Arquivo PDF compartilhado.
09	Selecionar Impressora.	Usuário seleciona uma impressora disponível.	Impressora disponível.	Impressora selecionada.
10	Imprimir Relatório PDF.	Usuário imprime o relatório PDF.	Relatório PDF gerado, impressora selecionada.	Relatório PDF impresso.

Fonte: Elaborado pelos autores

Tabela 10. Saídas de Caso de Uso

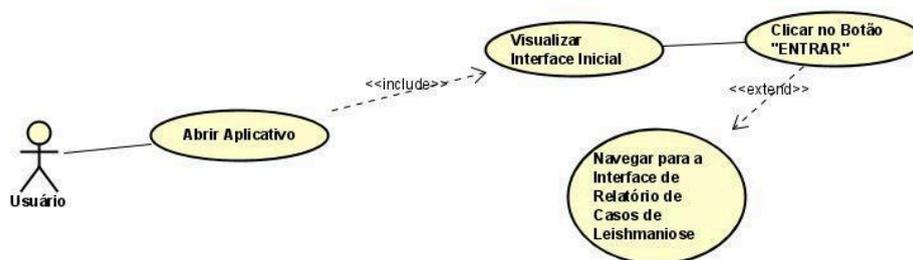
Msg1: "Erro. Quadra não preenchida."
Msg2: "Erro. Código do Endereço não preenchido."
Msg3: "Erro. Logradouro não preenchido."

Msg4: "Erro. Número do Imóvel não preenchido."
Msg5: "Erro. Nome do Proprietário não preenchido."
Msg6: "Erro. Telefone do Proprietário não preenchido."

Fonte: Elaborado pelos autores

3.2.2 Diagramas de Casos de Uso

Figura 1. Diagrama de Caso de Uso Interface Inicial.



Fonte: Elaborado pelos autores

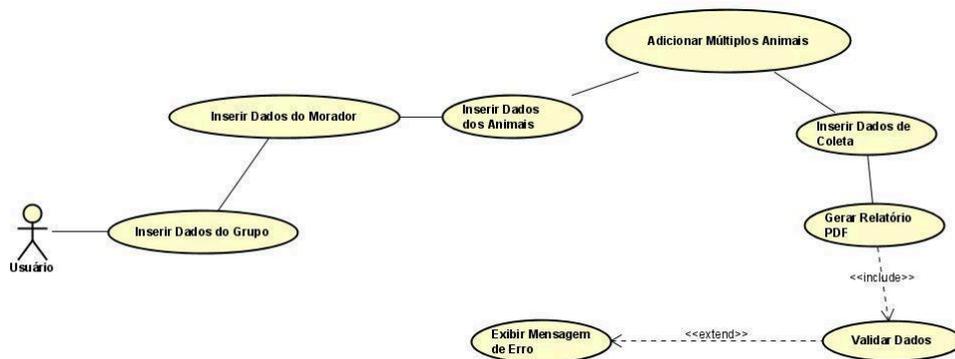
Cenário Principal:

1. Usuário deseja abrir o aplicativo.
 - 1.1 Usuário visualiza a interface inicial com o logo e o botão "ENTRAR".
 - 1.2 Usuário clica no botão "ENTRAR".
 - 1.3 Sistema navega para a interface de Relatório de Casos de Leishmaniose.

Cenário Alternativo:

2. Usuário abre o aplicativo.
 - 2.1 Usuário não interage com o botão "ENTRAR".

Figura 2. Diagrama de Caso de Uso Interface Relatório de Casos de Leishmaniose.



Fonte: Elaborado pelos autores

Cenário Principal:

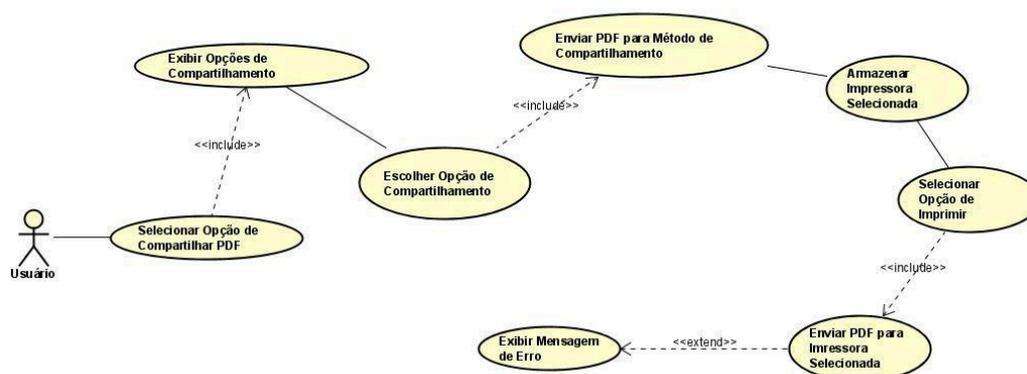
1. Usuário preenche os campos de dados do grupo: GT, GVE, ATL, Área, Quadra.
 - 1.1 Usuário preenche os campos de dados do morador: Código do Endereço, Logradouro, N° do Imóvel, Situação do Imóvel, Nome, Telefone.
 - 1.2 Usuário preenche os campos de dados dos animais: Nome, Registro, Idade, Sexo, Raça, Cor, Peso.
 - 1.3 Usuário especifica o número de animais e preenche os dados para cada um.
 - 1.4 Usuário preenche os campos de dados de coleta: Autorização, Situação da Coleta, Tamanho da Coleira, Estado da Coleira.
 - 1.5 Usuário clica no botão "Gerar Relatório".
 - 1.6 Sistema valida os dados e gera o PDF.

Cenário Alternativo:

2. Usuário clica no botão "Gerar Relatório" sem preencher todos os campos obrigatórios.
 - 2.1 Sistema exibe mensagens de erro indicando quais campos não foram preenchidos:
 - 2.1.1 Erro. Quadra não preenchida.

- 2.1.2 Erro. Código do Endereço não preenchido.
- 2.1.3 Erro. Logradouro não preenchido.
- 2.1.4 Erro. Número do Imóvel não preenchido.
- 2.1.5 Erro. Nome do Proprietário não preenchido.
- 2.1.6 Erro. Telefone do Proprietário não preenchido.

Figura 3. Diagrama de Caso de Uso Função de Compartilhamento e Impressão.



Fonte: Elaborado pelos autores

Cenário Principal:

1. Usuário seleciona a opção de compartilhar o PDF gerado.
 - 1.1 Sistema exibe as opções de compartilhamento disponíveis no dispositivo.
 - 1.2 Usuário escolhe uma opção de compartilhamento (e.g., e-mail, app de mensagens).
 - 1.3 Sistema envia o PDF para o método de compartilhamento escolhido pelo usuário.
 - 1.4 Sistema envia arquivo PDF para o método de compartilhamento escolhido pelo usuário.

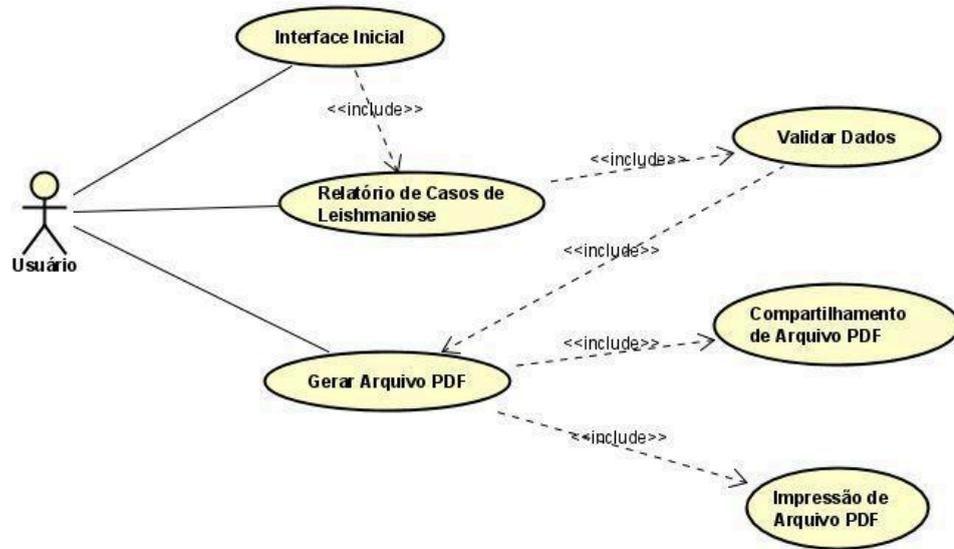
1.5 Sistema envia o PDF para o método de compartilhamento escolhido pelo usuário.

Cenário Alternativo:

2. Usuário seleciona a opção de compartilhar o PDF gerado.
 - 2.1 Sistema exibe as opções de compartilhamento disponíveis no dispositivo.
 - 2.2 Usuário seleciona uma impressora disponível.
 - 2.3 Sistema armazena a impressora selecionada.
 - 2.4 Usuário seleciona a opção de imprimir o relatório PDF.
 - 2.5 Sistema envia o PDF para a impressora selecionada.

3.2.3 Diagramas de Contexto

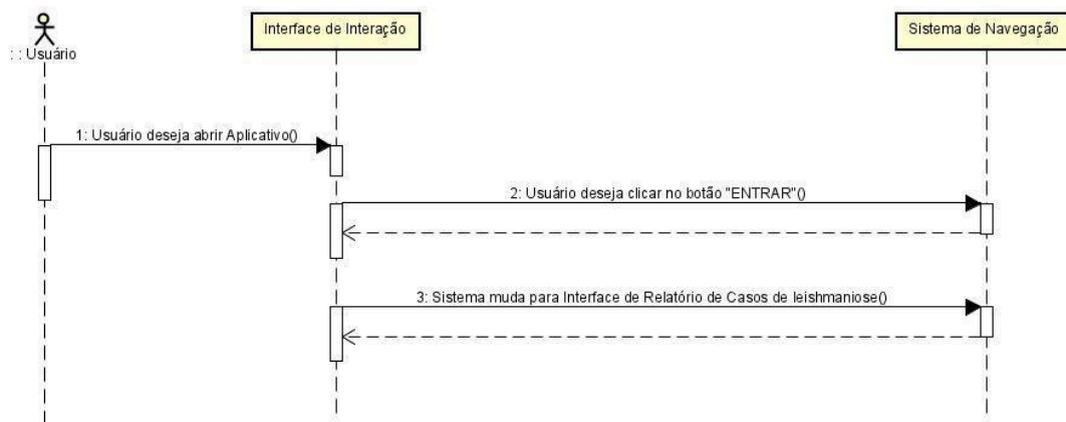
Figura 4. Diagrama de Contexto



Fonte: Elaborado pelos autores

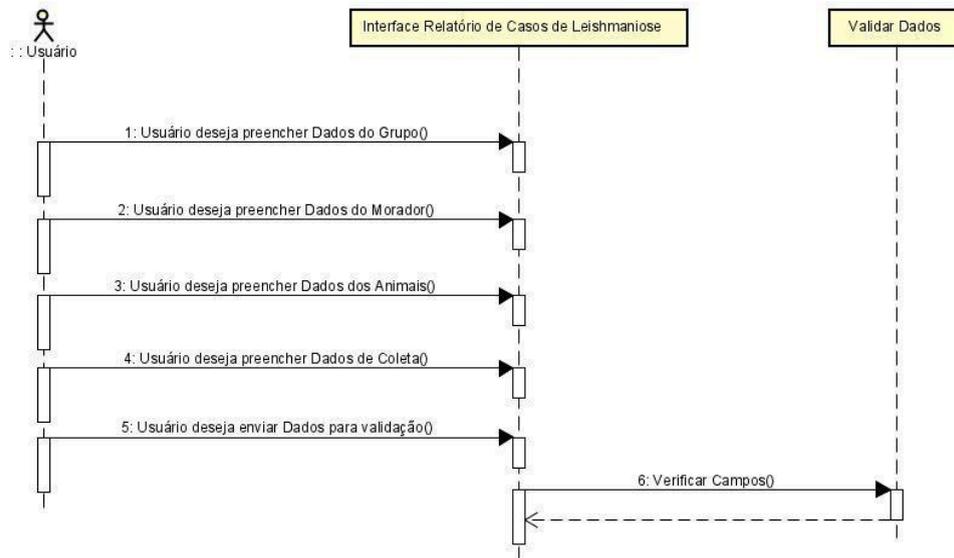
3.2.4 Diagrama de Seqüência

Figura 5. Diagrama de Seqüência do Caso de Uso Interface Inicial



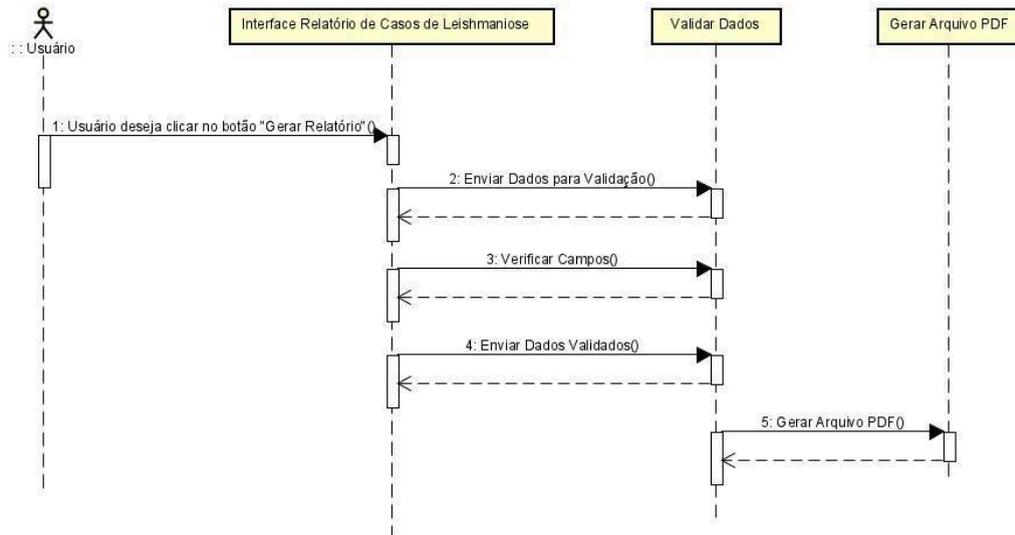
Fonte: Elaborado pelos autores

Figura 6. Diagrama de Seqüência do Caso de Uso Interface Relatório de Casos de Leishmaniose



Fonte: Elaborado pelos autores

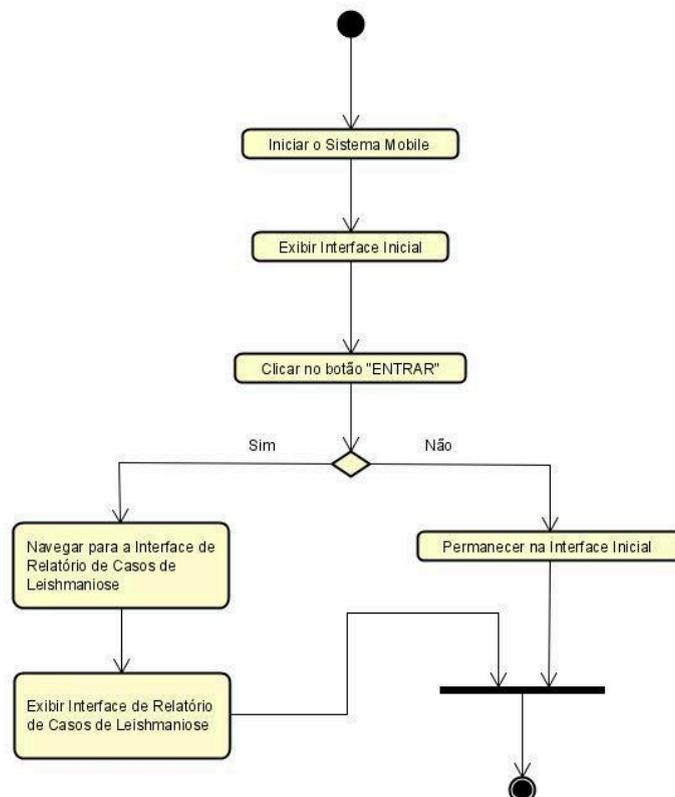
Figura 7. Diagrama de Seqüência do Caso de Uso Função de Compartilhamento e Impressão



Fonte: Elaborado pelos autores

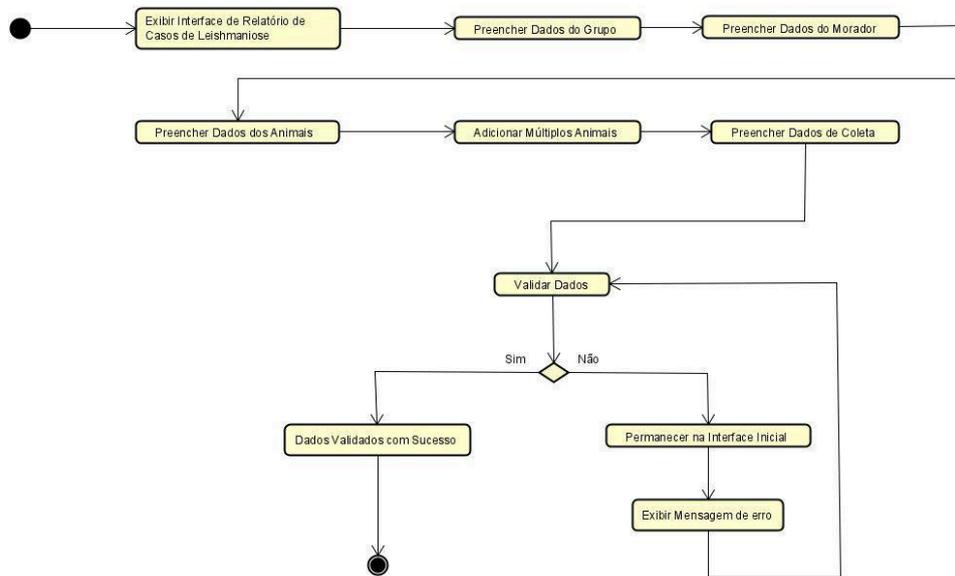
3.2.5 Diagrama de Atividades

Figura 8. Diagrama de Atividades do Caso de Uso Interface Inicial.



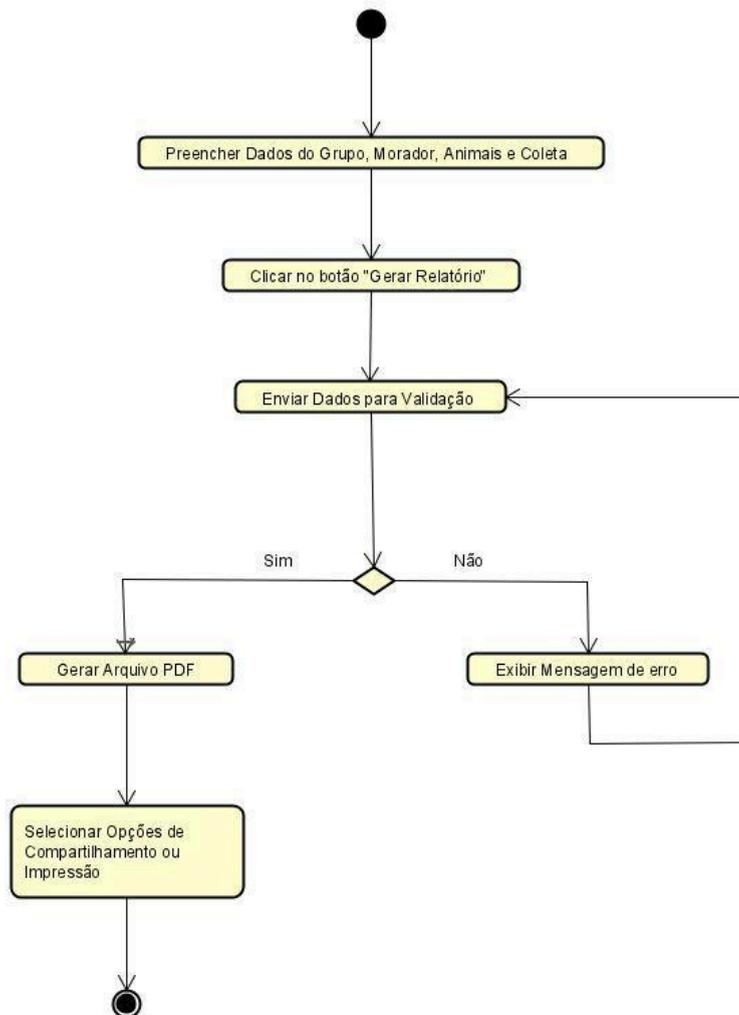
Fonte: Elaborado pelos autores

Figura 9. Diagrama de Atividades do Caso de Uso Interface Relatório de Casos de Leishmaniose.



Fonte: Elaborado pelos autores

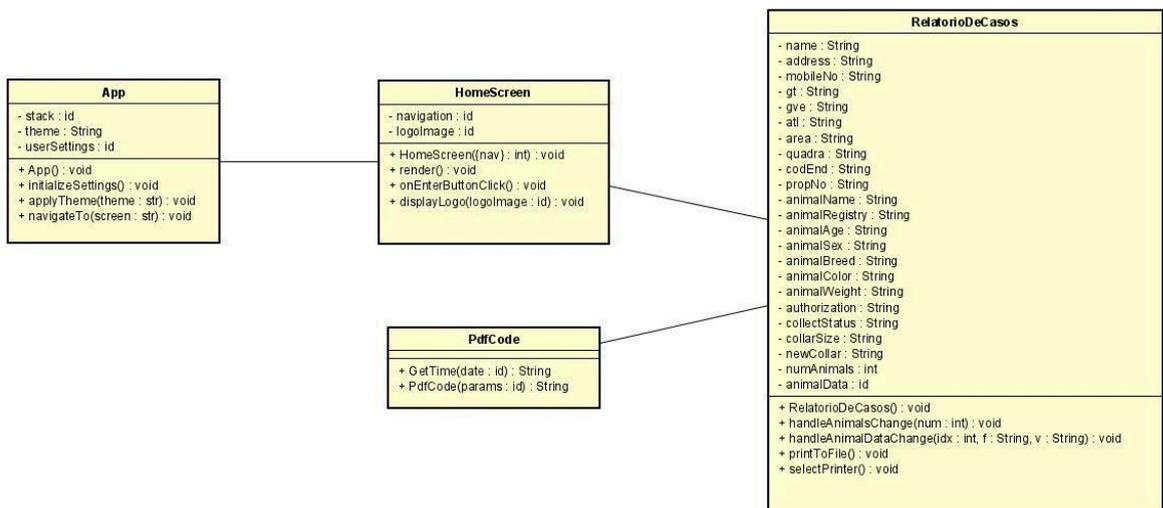
Figura 10. Diagrama de Atividades do Caso de Uso Função de Compartilhamento e Impressão.



Fonte: Elaborado pelos autores

3.2.6. Diagrama de Classes

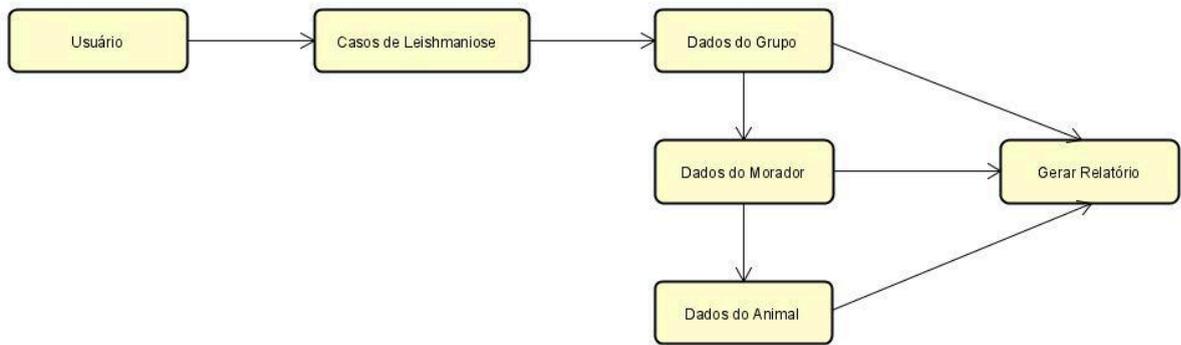
Figura 11. Diagrama de classes



Fonte: Elaborado pelos autores

3.2.7. DED

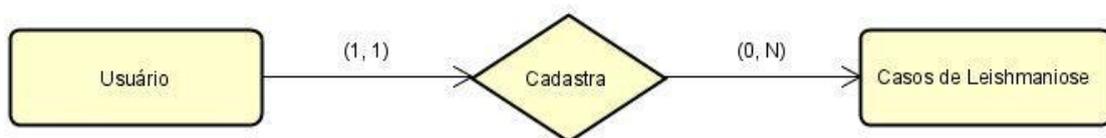
Figura 12. Diagrama de Estrutura de Dados (DED)



Fonte: Elaborado pelos autores

3.2.8. DER

Figura 13. Diagrama Entidade Relacionamento (DER)



Fonte: Elaborado pelos autores

3.2.9 Dicionário de Dados do Aplicativo

Figura 14. Dicionário de Dados – HomeScreen.js

Nome	Tipo de Dado	Descrição	Valor Padrão
navigation	Object	Objeto de navegação para mover entre diferentes telas	N/A

Fonte: Elaborado pelos autores

Figura 15. Dicionário de Dados – RelatorioDeCasos.js

Nome	Tipo de Dado	Descrição	Valor Padrão
name	String	Nome do Proprietário	""
address	String	Logradouro do imóvel	""
mobileNo	String	Telefone do proprietário	""
invoice	String	Número do documento	dateFormat(new Date(), "ddmmyyhhMss")
selectedPrinter	Object	Impressora selecionada	undefined
propSituation	String	Situação do imóvel	"2-Trabalhado"
propNo		Número do imóvel	""
name2	String	Nome secundário	""

age	String	Idade do proprietário	""
codEnd	String	Código do endereço	""
animalSex	String	Sexo do animal	"Macho"
breed	String	Raça do animal	""
register	String	Registro do animal	""
monthYear	String	Meses/Anos do animal	"Anos"
animalColor	String	Cor do animal	"5-Malhado"
animalWeight	String	Peso do animal	""
authorization	String	Autorização de coleta	"Sim"
collectStatus	String	Situação da coleta	"1-Coletado"
collarSize	String	Tamanho da coleira	"P"
newCollar	String	Descrever se a coleira é nova ou não	"Sim"
gtStatus	String	Situação GT	"09"
gveStatus	String	Situação GVE	"11"
atlStatus	String	Situação ATL	""
areaStatus	String	Situação da área	"01"
cityBlock	String	Quadra da cidade	""
numAnimals	Number	Número de animais	0
animalData	Array	Informações dos animais	[]

Fonte: Elaborado pelos autores

Figura 16. Dicionário de Dados – PdfCode.js

Nome	Tipo de Dado	Descrição	Valor Padrão
name	String	Nome do Proprietário	N/A
address	String	Logradouro do imóvel	N/A
mobileNo	String	Telefone do proprietário	N/A
invoice	String	Número do documento	N/A
selectedPrinter	Object	Impressora selecionada	N/A
propSituation	String	Situação do imóvel	N/A
propNo		Número do imóvel	N/A
name2	String	Nome secundário	N/A
age	String	Idade do proprietário	N/A
codEnd	String	Código do endereço	N/A
animalSex	String	Sexo do animal	N/A
breed	String	Raça do animal	N/A
register	String	Registro do animal	N/A
monthYear	String	Meses/Anos do animal	N/A
animalColor	String	Cor do animal	N/A
animalWeight	String	Peso do animal	N/A
authorization	String	Autorização de coleta	N/A
collectStatus	String	Situação da coleta	N/A
collarSize	String	Tamanho da coleira	N/A

newCollar	String	Descrever se a coleira é nova ou não	N/A
gtStatus	String	Situação GT	N/A
gveStatus	String	Situação GVE	N/A
atlStatus	String	Situação ATL	N/A
areaStatus	String	Situação da área	N/A
cityBlock	String	Quadra da cidade	N/A
numAnimals	Number	Número de animais	N/A
animalData	Array	Informações dos animais	N/A

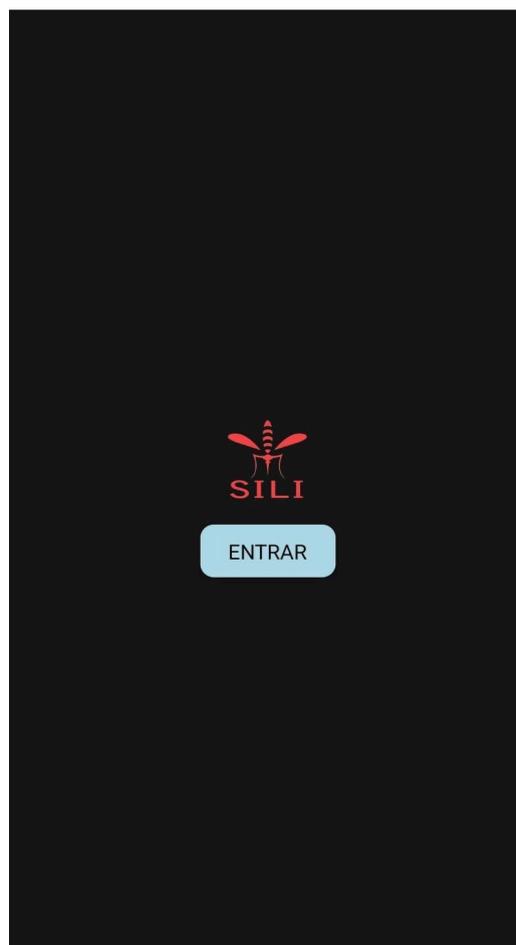
Fonte: Elaborado pelos autores

3.3 DESENVOLVIMENTO DO PROJETO

3.3.1 Protótipo

Figura 17. Tela de Login

Home



Fonte: Elaborado pelos autores

Figura 18. Relatório de Cadastro

← RelatórioDeCasos

Dados do Grupo

Documento Nº :
061124124943

GT :
09

GVE :
11

ATL :
TV

Área :
01

Quadra :
Número do Imóvel

Dados do Morador

CodEnd :
Código do Endereço

← RelatórioDeCasos

Dados do Morador

CodEnd :

Código do Endereço

Logradouro :

Nome da Rua do Imóvel

Nº :

Número do Imóvel

Situação :

2-Trabalhado

Proprietário :

Nome do Proprietário

Telefone :

(XX) XXXXX-XXXX

Dados do Animal

Quantidade de Animais :

0

← RelatorioDeCasos

Dados do Animal

Quantidade de Animais :

1

Nome do Animal :

Nome do Animal

Registro do Animal :

Registro do Animal

Idade do Animal :

Idade do Animal

Meses/Anos :

Meses

Sexo do Animal :

Macho

Raça do Animal :

Vira-Lata

Cor do Animal :

Branca

Peso do Animal :

← RelatórioDeCasos

Meses ▼

Sexo do Animal :

Macho ▼

Raça do Animal :

Vira-Lata ▼

Cor do Animal :

Branca ▼

Peso do Animal :

Peso do Animal (Kg)

Coleta Autorizada:

Sim ▼

Coleta Autorizada:

1-Coletado ▼

Coleta Autorizada:

P ▼

GERAR RELATÓRIO

Fonte: Elaborado pelos autores

Figura 19. Tela do PDF com os dados inseridos

PDF Save as PDF
▼

Copies: 01
Paper size: Letter

▼

PDF

RELATÓRIO DE COLETA

Centro de Controle de Zoonoses Araçatuba-SP

Dados do Grupo

CodMun (Código Municipal): 3502804
Município: Araçatuba-SP
GT: 09
GVE: 11

ATL: TV
GVE: 11
Área: 01
Quadra: 374

Dados do Morador

CodEnd (Código de Endereço): 123
Logradouro: logTest
Nº do Imóvel: 12345
Situação do Imóvel: 2-Trabalhado

Proprietário: propTest
Telefone: +55 18123456789

Dados dos Animais

Animal 1

Nome: dogTest1
Registro: 12345678910
Idade: 8 Anos
Sexo: Macho
Raça: Lhasa Apso
Cor: Creme
Peso: 5 kg
Coleta Autorizada: Sim
Situação da Coleta: 1-Coletado
Tamanho da Coleira: P
Coleira Nova: Sim

Animal 2

Nome: dogTest2
Registro: 1123456789
Idade: 9 Anos
Sexo: Fêmea
Raça: Vira-Lata
Cor: Caramelo

Documento Nº : 101124061608
Data da Coleta: 10-11-2024
Horário da Coleta: 6:17 PM

**Centro de Controle de Zoonoses,
Araçatuba-SP :-**
Endereço: Rua Dr. Luiz de Almeida, 145 -
Bairro Paraíso,
Araçatuba - SP, 16050-203
Telefone: (18) 3636-1180

1/2
✓

Peso: 4 kg
Coleta Autorizada: Sim
Situação da Coleta: 1-Coletado
Tamanho da Coleira: P
Coleira Nova: Sim

Animal 3

Nome: dogTest3
Registro: 1122345678910
Idade: 10 Anos
Sexo: Macho
Raça: Husky
Cor: Branca
Peso: 15 kg
Coleta Autorizada: Sim
Situação da Coleta: 1-Coletado
Tamanho da Coleira: M
Coleira Nova: Sim

Fonte: Elaborado pelos autores

3.3.2 Código Mobile

Figura 20. Tela de Código Home

```
1 import { StyleSheet, Text, View, Image, TouchableOpacity } from "react-native";
2 import React from "react";
3
4 export default function HomeScreen({ navigation }) {
5   return (
6     <View style={styles.container}>
7       <Image
8         style={{
9           width: 100,
10          height: 100,
11          }}
12        source={require("../assets/logo.png")}
13      />
14      <TouchableOpacity
15        onPress={() => navigation.navigate("RelatorioDeCasos")}
16        style={styles.button}
17      >
18        <Text>ENTRAR</Text>
19      </TouchableOpacity>
20    </View>
21  );
22 }
23
24 const styles = StyleSheet.create({
25   container: {
26     flex: 1,
27     backgroundColor: "#171717",
28     justifyContent: "center",
29     alignItems: "center",
30   },
31   button: {
32     height: 40,
33     width: 100,
34     backgroundColor: "lightblue",
35     alignItems: "center",
36     justifyContent: "center",
37     borderRadius: 10,
38     elevation: 10,
39   },
40 });
41
```

Fonte: Elaborado pelos autores

Figura 21. Tela de Dados do Grupo

```

1 <View style={styles.InputContainer}>
2   <Text style={styles.TextTitle}>Dados do Grupo</Text>
3 </View>
4
5   /* Nº do Documento */
6   <View style={styles.InputContainer}>
7     <Text style={styles.InputTitle}>Documento Nº : </Text>
8     <TextInput
9       style={styles.TextInput}
10      onChangeText={({text} => setInvoice(text)}
11      value={Invoice}
12      placeholder="Nº do Documento"
13    />
14 </View>
15
16   /* Situação GT */
17   <View style={styles.InputContainer}>
18     <Text style={styles.InputTitle}>GT : </Text>
19     <View style={styles.PickerContainer}>
20       <Picker
21         selectValue={GTstatus}
22         style={styles.Picker}
23         itemStyle={styles.Item}
24         onValueChange={({itemValue, itemIndex} => Set_GTstatus(itemValue)}
25       >
26         <Picker.Item label="01" value="01" />
27         <Picker.Item label="02" value="02" />
28         <Picker.Item label="03" value="03" />
29         <Picker.Item label="04" value="04" />
30         <Picker.Item label="05" value="05" />
31         <Picker.Item label="06" value="06" />
32         <Picker.Item label="07" value="07" />
33         <Picker.Item label="08" value="08" />
34         <Picker.Item label="09" value="09" />
35       </Picker>
36 </View>
37 </View>
38
39   /* Situação GE */
40   <View style={styles.InputContainer}>
41     <Text style={styles.InputTitle}>GE : </Text>
42     <View style={styles.PickerContainer}>
43       <Picker
44         selectValue={GEstatus}
45         style={styles.Picker}
46         onValueChange={({itemValue, itemIndex} => Set_GEstatus(itemValue)}
47       >
48         <Picker.Item label="01" value="01" />
49         <Picker.Item label="02" value="02" />
50         <Picker.Item label="03" value="03" />
51         <Picker.Item label="04" value="04" />
52         <Picker.Item label="05" value="05" />
53         <Picker.Item label="06" value="06" />
54         <Picker.Item label="07" value="07" />
55         <Picker.Item label="08" value="08" />
56         <Picker.Item label="09" value="09" />
57         <Picker.Item label="10" value="10" />
58         <Picker.Item label="11" value="11" />
59       </Picker>
60 </View>
61 </View>
62
63   /* Situação ATL */
64   <View style={styles.InputContainer}>
65     <Text style={styles.InputTitle}>ATL : </Text>
66     <View style={styles.PickerContainer}>
67       <Picker
68         selectValue={ATLstatus}
69         style={styles.Picker}
70         onValueChange={({itemValue, itemIndex} => Set_ATLstatus(itemValue)}
71       >
72         <Picker.Item label="TV" value="TV" />
73         <Picker.Item label="Test1" value="Test1" />
74         <Picker.Item label="Test2" value="Test2" />
75         <Picker.Item label="Test3" value="Test3" />
76       </Picker>
77 </View>
78 </View>
79
80   /* Área */
81   <View style={styles.InputContainer}>
82     <Text style={styles.InputTitle}>Área : </Text>
83     <View style={styles.PickerContainer}>
84       <Picker
85         selectValue={AreaStatus}
86         style={styles.Picker}
87         onValueChange={({itemValue, itemIndex} =>
88           Set_AreaStatus(itemValue)
89         )
90       >
91         <Picker.Item label="01" value="01" />
92         <Picker.Item label="02" value="02" />
93         <Picker.Item label="03" value="03" />
94         <Picker.Item label="04" value="04" />
95         <Picker.Item label="05" value="05" />
96         <Picker.Item label="06" value="06" />
97         <Picker.Item label="07" value="07" />
98         <Picker.Item label="08" value="08" />
99       </Picker>
100 </View>
101 </View>
102
103   /* Quadra */
104   <View style={styles.InputContainer}>
105     <Text style={styles.InputTitle}>Quadra : </Text>
106     <TextInput
107       style={styles.TextInput}
108       keyboardType="number-pad"
109       onChangeText={({text} => Set_CityBlock(text)}
110       value={CityBlock}
111       placeholder="Número do Imóvel"
112       placeholderTextColor="8a9a9a"
113     />
114 </View>

```

Fonte: Elaborado pelos autores

Figura 22. Tela de Código Dados do Morador

```

1  /* Código do Endereço */
2  <View style={styles.InputContainer}>
3    <Text style={styles.InputTitle}>CodEnd : </Text>
4    <TextInput
5      style={styles.textInput}
6      keyboardType="number-pad"
7      onChangeText={({text}) => Set_CodEnd(text)}
8      value={CodEnd}
9      placeholder="Código do Endereço"
10     placeholderTextColor="#a9a9a9"
11   />
12 </View>
13
14 /* Logradouro */
15 <View style={styles.InputContainer}>
16 <Text style={styles.InputTitle}>Logradouro : </Text>
17 <TextInput
18   style={styles.textInput}
19   onChangeText={({text}) => Set_Address(text)}
20   value={Address}
21   placeholder="Nome da Rua do Imóvel"
22   placeholderTextColor="#a9a9a9"
23 />
24 </View>
25
26 /* Número do Imóvel */
27 <View style={styles.InputContainer}>
28 <Text style={styles.InputTitle}>Nº : </Text>
29 <TextInput
30   style={styles.textInput}
31   keyboardType="number-pad"
32   onChangeText={({text}) => Set_PropNo(text)}
33   value={PropNo}
34   placeholder="Número do Imóvel"
35   placeholderTextColor="#a9a9a9"
36 />
37 </View>
38
39 /* Situação do Imóvel */
40 <View style={styles.InputContainer}>
41 <Text style={styles.InputTitle}>Situação : </Text>
42 <View style={styles.PickerContainer}>
43 <Picker
44   selectedValue={PropSituation}
45   style={styles.Picker}
46   onChange={({itemValue, itemIndex}) =>
47     Set_PropSituation(itemValue)
48   }
49 >
50 <Picker.Item label="Test1" value="Test1" />
51 <Picker.Item label="2-Trabalhado" value="2-Trabalhado" />
52 <Picker.Item label="Test3" value="Test3" />
53 <Picker.Item label="Test4" value="Test4" />
54 <Picker.Item label="Test5" value="Test5" />
55 <Picker.Item label="Test6" value="Test6" />
56 <Picker.Item label="Test7" value="Test7" />
57 </Picker>
58 </View>
59 </View>
60
61 /* Nome do Proprietário */
62 <View style={styles.InputContainer}>
63 <Text style={styles.InputTitle}>Proprietário : </Text>
64 <TextInput
65   style={styles.textInput}
66   onChangeText={({text}) => set_Name(text)}
67   value={name}
68   placeholder="Nome do Proprietário"
69   placeholderTextColor="#a9a9a9"
70 />
71 </View>
72
73 /* Telefone do Proprietário */
74 <View style={styles.InputContainer}>
75 <Text style={styles.InputTitle}>Telefone : </Text>
76 <TextInput
77   style={styles.textInput}
78   keyboardType="phone-pad"
79   onChangeText={({text}) => Set_Mobile_No(text)}
80   value={Mobile_No}
81   placeholder="(XX) XXXXX-XXXX"
82   placeholderTextColor="#a9a9a9"
83 />
84 </View>
85

```

Fonte: Elaborado pelos autores

Figura 23. Tela de Código Dados Gerais do Animal

```

1  {/} Quantidade de Animais */
2  <View style={styles.InputContainer}>
3    <Text style={styles.InputTitle}>Quantidade de Animais : </Text>
4    <TextInput
5      style={styles.textInput}
6      keyboardType="number-pad"
7      onChangeText={({text}) => handleNumAnimalsChange(parseInt(text))}
8      value={numAnimals.toString()}
9    />
10 </View>
11
12 {/} Nome do Animal */
13 [animalData.map((animal, index) => {
14   <View style={styles.InputContainer} key={index}>
15     <Text style={styles.InputTitle}>Nome do Animal : </Text>
16     <TextInput
17       style={styles.textInput}
18       value={animal.name}
19       placeholder="Nome do Animal"
20       placeholderTextColor="#888888"
21       onChangeText={({text}) =>
22         handleAnimalDataChange(index, "name", text)
23       }
24     />
25   }
26 />
27
28 {/} Registro de Animal */
29 <Text style={styles.InputTitle}>Registro do Animal : </Text>
30 <TextInput
31   style={styles.textInput}
32   keyboardType="number-pad"
33   value={animal.registry}
34   placeholder="Registro do Animal"
35   placeholderTextColor="#888888"
36   onChangeText={({text}) =>
37     handleAnimalDataChange(index, "registry", text)
38   }
39 />
40
41 {/} Idade do Animal */
42 <Text style={styles.InputTitle}>Idade do Animal : </Text>
43 <TextInput
44   style={styles.textInput}
45   keyboardType="numeric"
46   value={animal.age}
47   placeholder="Idade do Animal"
48   placeholderTextColor="#888888"
49   onChangeText={({text}) =>
50     handleAnimalDataChange(index, "age", text)
51   }
52 />
53
54 {/} Idade do Animal (Meses/Anos) */
55 <Text style={styles.InputTitle}>Meses/Anos : </Text>
56 <View style={styles.PickerContainer}>
57   <Picker
58     style={styles.Picker}
59     selectValue={animal.ageInUnit}
60     onValueChange={({itemValue}) =>
61       handleAnimalDataChange(index, "ageInUnit", itemValue)
62     }
63     <Picker.Item label="Meses" value="Meses" />
64     <Picker.Item label="Anos" value="Anos" />
65   </Picker>
66 </View>
67
68 {/} Sexo do Animal */
69 <Text style={styles.InputTitle}>Sexo do Animal : </Text>
70 <View style={styles.PickerContainer}>
71   <Picker
72     style={styles.Picker}
73     selectValue={animal.sex}
74     onValueChange={({itemValue}) =>
75       handleAnimalDataChange(index, "sex", itemValue)
76     }
77     <Picker.Item label="Macho" value="Macho" />
78     <Picker.Item label="Fêmea" value="Fêmea" />
79   </Picker>
80 </View>
81
82 {/} Raça do Animal */
83 <Text style={styles.InputTitle}>Raça do Animal : </Text>
84 <View style={styles.PickerContainer}>
85   <Picker
86     style={styles.Picker}
87     selectValue={animal.breed}
88     onValueChange={({itemValue}) =>
89       handleAnimalDataChange(index, "breed", itemValue)
90     }
91     <Picker.Item label="Vira-Lata" value="Vira-Lata" />
92     <Picker.Item label="Bulldog" value="Bulldog" />
93     <Picker.Item label="Lhasa Apso" value="Lhasa Apso" />
94   </Picker>
95 </View>
96
97 {/} Cor do Animal */
98 <Text style={styles.InputTitle}>Cor do Animal : </Text>
99 <View style={styles.PickerContainer}>
100   <Picker
101     style={styles.Picker}
102     selectValue={animal.color}
103     onValueChange={({itemValue}) =>
104       handleAnimalDataChange(index, "color", itemValue)
105     }
106     <Picker.Item label="Branca" value="Branca" />
107     <Picker.Item label="Bege" value="Bege" />
108     <Picker.Item label="Creme" value="Creme" />
109     <Picker.Item label="Caramelo" value="Caramelo" />
110   </Picker>
111 </View>
112
113 {/} Peso do Animal */
114 <Text style={styles.InputTitle}>Peso do Animal : </Text>
115 <TextInput
116   style={styles.textInput}
117   keyboardType="numeric"
118   value={animal.weight}
119   placeholder="Peso do Animal (Kg)"
120   placeholderTextColor="#888888"
121   onChangeText={({text}) =>
122     handleAnimalDataChange(index, "weight", text)
123   }
124 />
125
126

```

Fonte: Elaborado pelos autores

Figura 24. Tela de Código Dados de Coleta do Animal

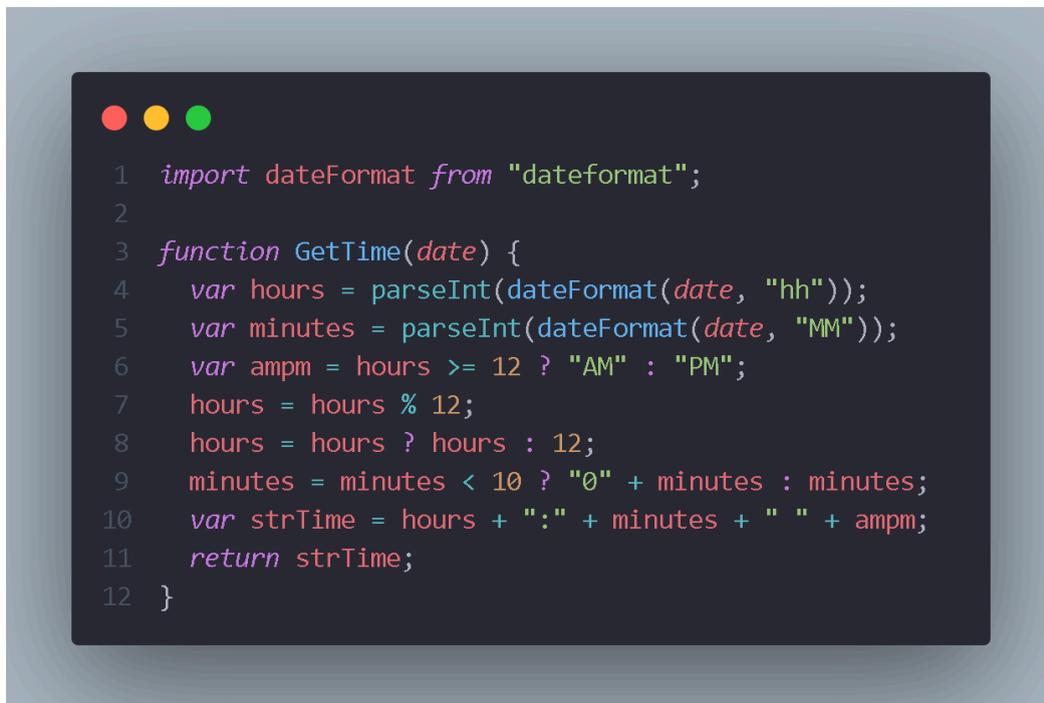
```

1  /* Autorização de Coleta */
2  <Text style={styles.InputTitle}>Coleta Autorizada: </Text>
3  <View style={styles.PickerContainer}>
4    <Picker
5      style={styles.Picker}
6      selectedValue={animal.collectionAuthorization}
7      onValueChange={(itemValue) =>
8        handleAnimalDataChange(
9          index,
10         "collectionAuthorization",
11         itemValue
12       )
13     }
14   >
15     <Picker.Item label="Sim" value="Sim" />
16     <Picker.Item label="Não" value="Não" />
17   </Picker>
18 </View>
19
20 /* Situação de Coleta */
21 <Text style={styles.InputTitle}>Coleta Autorizada: </Text>
22 <View style={styles.PickerContainer}>
23   <Picker
24     style={styles.Picker}
25     selectedValue={animal.collectionSituation}
26     onValueChange={(itemValue) =>
27       handleAnimalDataChange(
28         index,
29         "collectionSituation",
30         itemValue
31       )
32     }
33   >
34     <Picker.Item label="1-Coletado" value="1-Coletado" />
35     <Picker.Item label="2-Não Coletado" value="2-Não Coletado" />
36   </Picker>
37 </View>
38
39 /* Tamanho da Coleira */
40 <Text style={styles.InputTitle}>Coleta Autorizada: </Text>
41 <View style={styles.PickerContainer}>
42   <Picker
43     style={styles.Picker}
44     selectedValue={animal.collarSize}
45     onValueChange={(itemValue) =>
46       handleAnimalDataChange(index, "collarSize", itemValue)
47     }
48   >
49     <Picker.Item label="P" value="P" />
50     <Picker.Item label="M" value="M" />
51     <Picker.Item label="G" value="G" />
52   </Picker>
53 </View>
54
55 <View style={styles.line} />
56 </View>

```

Fonte: Elaborado pelos autores

Figura 25. Tela de Código Definição de Data e Hora Automáticas no Arquivo PDF



```
1 import dateFormat from "dateFormat";
2
3 function GetTime(date) {
4     var hours = parseInt(dateFormat(date, "hh"));
5     var minutes = parseInt(dateFormat(date, "MM"));
6     var ampm = hours >= 12 ? "AM" : "PM";
7     hours = hours % 12;
8     hours = hours ? hours : 12;
9     minutes = minutes < 10 ? "0" + minutes : minutes;
10    var strTime = hours + ":" + minutes + " " + ampm;
11    return strTime;
12 }
```

Fonte: Elaborado pelos autores

Figura 26. Tela de Código Gerar Arquivo PDF

```

1   <div style="
2   width: 100%;
3   height: auto;
4   padding: 15px;
5   display: flex;
6   flex-direction: row;
7   justify-content: space-evenly;
8   ">
9
10  <div style="
11  width: 50%;
12  align-items: flex-start;
13  ">
14  <h3 class="Invoice-user">
15  <b>Dados do Grupo</b>
16  </h3>
17  <p class="Invoice-user">
18  CodMun (Código Municipal): 3502804 <br/>
19  Município: Araçatuba-SP <br/>
20  GT: ${GTstatus} <br/>
21  GVE: ${GVEstatus}
22  </p>
23  <p class="Invoice-user">
24  ATL: ${ATLstatus} <br/>
25  GVE: ${GVEstatus} <br/>
26  Área: ${AreaStatus} <br/>
27  Quadra: ${CityBlock}
28  </p>
29  <h3 class="Invoice-user">
30  <b>Dados do Morador</b>
31  </h3>
32  <p class="Invoice-user">
33  CodEnd (Código de Endereço): ${CodEnd} <br/>
34  Logradouro: ${Address} <br/>
35  Nº do Imóvel: ${PropNo} <br/>
36  Situação do Imóvel: ${PropSituation}
37  </p>
38  <p class="Invoice-user">
39  Proprietário: ${name} <br/>
40  Telefone: +55 ${Mobile_No}
41  </p>
42  <h3 class="Invoice-user"><b>Dados dos Animais</b></h3>
43  ${animalData}
44  .map(
45    (animal, index) => {
46      <div style="margin-bottom: 10px;">
47        <h4>Animal ${index + 1}</h4>
48        <p>
49          Nome: ${animal.name} <br/>
50          Registro: ${animal.registry} <br/>
51          Idade: ${animal.age} ${animal.ageUnit} <br/>
52          Sexo: ${animal.sex} <br/>
53          Raça: ${animal.breed} <br/>
54          Cor: ${animal.color} <br/>
55          Peso: ${animal.weight} kg <br/>
56          Coleta Autorizada: ${animal.collectionAuthorization} <br/>
57          Situação da Coleta: ${animal.collectionsituation} <br/>
58          Tamanho da Coleira: ${animal.collarSize} <br/>
59          Coleira Nova: ${animal.newCollar}
60        </p>
61      </div>
62    }
63  ).join("")
64  <h3 class="Invoice-user">
65  <b>Dados de Coleta</b>
66  </h3>
67  <p class="Invoice-user">
68  Autorizada: ${Authorization} <br/>
69  Situação: ${collectStatus} <br/>
70  Tamanho da Coleira: ${collarSize} <br/>
71  Coleira Nova: ${NewCollar}
72  </p>
73  </div>
74  <div style="align-items: flex-end;">
75  <p>Documento Nº : ${Invoice}<br/>
76  Data da Coleta: ${dateFormat(Date.now(), "dd-mm-yyyy")}<br/>
77  Horário da Coleta: ${getTime(new Date())}</p>
78  <br/>
79  <p>Centro de Controle de Zoonoses, Araçatuba-SP :- <br/>
80  Endereço: Rua Dr. Luiz de Almeida, 145 - Paraíso, Araçatuba - SP, 16050-203<br/>
81  Telefone: (18) 3636-1180
82  </p>
83  </div>
84  </div>
85  </div>
86  <hr/>
87  <hr/>
88
89  <p>Dados de Contato - <br/>
90  Endereço: Rua Dr. Luiz de Almeida, 145 - Paraíso, Araçatuba - SP<br/>
91  Telefone: (18) 3636-1180
92  </p>
93
94  </div>
95
96  </div>
97  </body>
98  </html>
99  ";
100
101  export { PdfCode };

```

Fonte: Elaborado pelos autores

4.CONCLUSÃO

Diante dos fatos mencionados, esta plataforma mobile, auxilia os agentes epidemiológicos, fornecendo ferramentas para substituir a ficha de cadastro em papel. A plataforma mobile desenvolvida acelera o processo de comunicação e fiscalização da situação epidemiológica do município de Araçatuba. O projeto representa uma melhoria significativa no controle e monitoramento do processo de relato de dados do centro de controle de zoonoses.

A tecnologia fornece um meio de simplificar a documentação e a transmissão dos dados discursivos e registrar as atividades dos agentes sanitários com mais precisão e eficiência. Os relatórios em pdf também o ajudam a manter o controle dos dados inseridos neste sistema.

O desenvolvimento do sistema foi realizado utilizando tecnologias eficientes, como os frameworks React Native. As suas bibliotecas flexíveis para a manipulação e conversão de diferentes tipos de dados.

Para concluir, o estudo tem o potencial de se tornar uma base sólida para muitos outros estudos idênticos que lidam com problemas semelhantes de coleta de informações sobre doenças ou condições sanitárias precárias. Também será crucial manter e atualizar o sistema. Com o tempo, muitas sugestões específicas ou ideias de atualização serão disponibilizadas. Deste modo, pode-se aumentar o acesso das pessoas comuns à tecnologia da informação na área da saúde e facilitar o uso de tecnologias digitais pelos agentes da área. Isso pode ser útil para melhorar novas ferramentas, melhorando a saúde pública e constante da população.

Finalmente, torna-se crucial destacar que este estudo pode servir como um alicerce sólido para o desenvolvimento de projetos análogos que auxiliem em outros casos de coleta de informações sobre doenças distintas ou condições sanitárias precárias, não apenas no município de Araçatuba, mas também em outras regiões. Isso justifica a necessidade da manutenção e atualização contínua deste sistema.

Não apenas, considerando as posteriores possíveis necessidades de aprimoramento do sistema, é possível expandir o acesso às tecnologias de informação na área da saúde e melhorar o processo de atividades digitais dos agentes, com o apoio de entidades governamentais, possibilitando a ampliação de ferramentas desenvolvidas,

aperfeiçoando os serviços prestados, e, por consequência, elevando a saúde geral da população.

5. TRABALHOS FUTUROS

Com o objetivo de pensar no futuro, visando explorar as oportunidades de pesquisa que surgiram ao longo deste estudo, é importante destacar possíveis áreas de desenvolvimento do projeto. Nosso objetivo não é apenas consolidar as contribuições existentes, mas também continuar construindo o conhecimento de nossa área.

A utilização de serviços de armazenamento em nuvem trará vários benefícios, uma vez que a quantidade de dados continuará aumentando. Esses sistemas incluem medidas de segurança altamente avançadas, são escalonáveis e adaptáveis e, portanto, se ajustam a qualquer aumento nas necessidades.

Um sistema de notificação em tempo real, que permitirá aos órgãos enviar alertas rapidamente, usando diferentes canais, garantindo que nenhuma informação crucial seja deixada de lado. O desenvolvimento de ferramentas de apoio instrucional, como popups instrutivos, vídeos, documentos e uma seção de perguntas frequentes, irá melhorar a experiência do usuário.

Um bot baseado em IA pode analisar os padrões e os riscos dos dados coletados e fornecer medidas preventivas adequadas. Uma atualização de base pública em tempo real para qualquer visitante interessado pode ser aprimorada com as contribuições públicas, promovendo um maior envolvimento da comunidade.

Todas essas melhorias futuras podem contribuir significativamente para a experiência e a eficiência do sistema, bem como para sua eficácia no monitoramento e combate de doenças e outras ameaças conhecidas. Portanto, é essencial considerar essas sugestões para garantir que o sistema continue a oferecer informações precisas e atuais e a satisfazer as necessidades dos desenvolvedores e usuários.

6. REFERÊNCIAS

A GOIS. **Ionic Framework: Construa aplicativos para todas as plataformas mobile.**

2017. Disponível em:

<https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=y2EqDwAAQBAJ&oi=fnd&pg=PT2&dq=+ionic+framework+livro&ots=ULd7f9G1EE&sig=ZC7oatWjPq8voK6vExki46Wgth0#v=onepage&q&f=false>. Acesso em: 09 nov. 2024.

AGUIAR, P. F.; RODRIGUES, R. K. **Leishmaniose Visceral no Brasil: Artigo de Revisão.** 2017. Disponível em:

<<https://www.periodicos.unimontes.br/index.php/unicientifica/article/download/2119/2200>>. Acesso em: 20 ou. 2024.

ALVES, G. et al. **Investigação Preliminar da Fauna de Flebotomíneos e Seu Potencial Vetorial em Ambientes Urbanos de Volta Redonda.** In: *Congresso Brasileiro de Ciências e Saberes Multidisciplinares*. 2023. Disponível em:

<<https://conferenciasunifoa.emnuvens.com.br/tc/article/view/1051/930>>. Acesso em: 09 nov. 2024

ANJOS, L. F. R. **Evolução do Javascript em aplicações multiplataforma.** 2017.

Disponível em:

<https://repositorio.ifsc.edu.br/bitstream/handle/123456789/1019/20190311_TCC_LuizFelipeDosAnjos.pdf>. Acesso em: 01 nov. 2024.

BUSS, H. C. **Linguagem de programação Ipe: programação puramente funcional para APIs REST.** 2023. Disponível em:

<<https://repositorio.ufsc.br/bitstream/handle/123456789/248488/TCC.pdf?sequence=1&isAllowed=y>>. Acesso em: 10 nov. 2024

CUNHA, I. A. D. et al. **Missão Patinha: Site de Divulgação de Ong's de Matão-SP.**

2023. Disponível em: <<https://ric.cps.sp.gov.br/handle/123456789/22530>>. Acesso em: 02 nov. 2024.

ENDLER, M. **Uma biblioteca para testes determinísticos em sistemas distribuídos com comunicação assíncrona**. 2024. Disponível em:

<<https://www.maxwell.vrac.puc-rio.br/62872/62872.PDF>> . Acesso em: 02 nov. 2024.

FARIA, J. R. S. **Extensão para auxiliar na acessibilidade para aplicativos feitos em Flutter utilizando VSCode**. 2023. Disponível em: ufpe.br. Acesso em: 02 nov. 2024

FLANAGAN, David. **JavaScript: o guia definitivo**. Bookman Editora, 2004.

Disponível em:

<https://books.google.com.br/books?hl=pt-BR&lr=&id=zWNyDgAAQBAJ&oi=fnd&pg=PR1&dq=javascript&ots=IACHw-PdhM&sig=jOV2mN_Ye1jjzodljixe02ZOFHQ>.

Acesso em: 02 nov. 2024.

FOGANHOLI, J. N. **Importância da Leishmaniose na Saúde Pública**. 2011.

Disponível em:

<http://faef.revista.inf.br/imagens_arquivos/arquivos_destaque/fA4b0h8gC5IQUuu_2013-6-27-15-48-34.pdf?msclkid=e76ada94c35511ec8a83e5e84561f8fb>. Acesso em: 10 nov. 2024.

GUERRA, J. C. **Manual de Pesquisa Qualitativa**. 2014. Disponível em:

<<https://docente.ifsc.edu.br/luciane.oliveira/MaterialDidatico/P%C3%B3s%20Gest%C3%A3o%20Escolar/Legisla%C3%A7%C3%A3o%20e%20Pol%C3%ADticas%20P%C3%BAblicas/Manual%20de%20Pesquisa%20Qualitativa.pdf>>. Acesso em: 01 nov. 2024.

MINISTÉRIO DA SAÚDE. SECRETARIA DE VIGILÂNCIA EM SAÚDE.

DEPARTAMENTO DE VIGILÂNCIA EPIDEMIOLÓGICA. **Manual de Vigilância e Controle da Leishmaniose Visceral**. 2006. Disponível em:

<https://bvsmis.saude.gov.br/bvs/publicacoes/manual_vigilancia_controle_leishmaniose_viscerai.pdf>. Acesso em: 10 nov. 2024.

MINISTÉRIO DA SAÚDE. **Manual de Vigilância: Prevenção e Controle de Zoonoses**. 2016. Disponível em:

<<https://www.gov.br/saude/pt-br/centrais-deconteudo/publicacoes/publicacoes-svs/zoonose/manual-zoonoses-normas-2v-7julho16site.pdf/view?msckid=07a711f0c35511ec85cdf2ec6a6fc16b>>. Acesso em: 10 nov. 2024.

MINISTÉRIO DA SAÚDE. **Leishmaniose Visceral Recomendações Clínicas para Redução da Letalidade**. 2011. Disponível em:

<https://bvsm.sau.gov.br/bvs/publicacoes/leishmaniose_visceral_reducao_letalidade.pdf?msckid=db53a998c35611ec9717022402fc0669>. Acesso em: 02 nov. 2024.

OLIVEIRA, M. F. **Metodologia Científica: um manual para a realização de pesquisas em administração**. 2011. Disponível em:

<https://files.cercomp.ufg.br/weby/up/567/o/Manual_de_metodologia_cientifica_-_Prof_Maxwell.pdf>. Acesso em: 02 nov. 2024.

PEREIRA, M. H. R. **AngularJS: Uma abordagem prática e objetiva**. 2014.

Disponível em:

<https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=MUelBQAAQBAJ&oi=fnd&pg=PA13&dq=angular+livro&ots=qIfE7WCWD-&sig=wOfMSk8raF66SzWFsXUXkyCDHdY#v=onepage&q&f=false>. Acesso em: 01 nov. 2024.

ROCHA, G. C. C. **Uma Análise comparativa entre frameworks de persistência de dados em javascript**. 2022. Disponível em:

<https://repositorio.ufc.br/bitstream/riufc/70825/1/2022_tcc_gccdarocha.pdf>. Acesso em: 02 nov. 2024.

SEVERINO, R. O. **Importância das Organizações Não Governamentais (ONGs) de animais e seus impactos no Meio Ambiente e Urbano**. 2023. Disponível em:

<<https://repositorio.unesp.br/server/api/core/bitstreams/1e365cdd-cfa6-4d1d-8c8e-49dda7bfd95f/content>>. Acesso em: 09 nov. 2024

SILVA, Maurício Samy. **JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. Novatec Editora, 2020. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=6DfnDwAAQBAJ&oi=fnd&pg=PT24&dq=javascript&ots=ZmGTrov_27&sig=DfXTIr14SNNZA7iQTYLdB_uRbQU>. Acesso em: 10 nov 2024

SIQUEIRA, K. G. et al. **Orion: Solução digital para preenchimento da ficha de validação do Trabalho de Conclusão de Curso**. 2023. Disponível em: <<https://ric.cps.sp.gov.br/handle/123456789/22795>>. Acesso em: 02 nov. 2024

SOARES, A. S.; DADALTO, F. G. S. **Transformando ideias em realidade com Uizard.io**. 2024. Disponível em: <<https://ric.cps.sp.gov.br/handle/123456789/23493>>. Acesso em: 10 nov. 2024

FARIA, J. R. S. **Extensão para auxiliar na acessibilidade para aplicativos feitos em Flutter utilizando VSCode**. 2023. Disponível em: ufpe.br. Acesso em: 02 nov. 2024