

PAMPACARE LVC: APLICATIVO PARA CONTROLE DE CASOS DE LEISHMANIOSE VISCERAL CANINA

João Victor Charão Germain*
Prof. Dr. Fábio Paulo Basso**
Profa. Dra. Débora da Cruz Payão Pellegrini***

RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo móvel para o controle e notificação de casos de leishmaniose visceral canina em Uruguaiiana/RS, com foco na experiência do usuário e soluções inovadoras, tais como funcionamento offline (offline-first), integração com serviços de mapas, e informações epidemiológicas baseadas em dados inseridos no aplicativo. A leishmaniose visceral canina é uma doença infecciosa grave que afeta animais e humanos, tornando-se de grande importância para a saúde pública e animal. O aplicativo proposto visa simplificar o processo de notificação de casos, fornecer informações úteis e relevantes sobre a doença e sua prevenção, e permitir o controle e monitoramento dos casos de leishmaniose visceral canina na região de Uruguaiiana/RS.

Palavras-chaves: Aplicativo móvel; Leishmaniose Visceral Canina; Controle de doenças; Monitoramento epidemiológico.

ABSTRACT

This work presents the development of a mobile application for the control and notification of cases of canine visceral leishmaniasis in Uruguaiiana/RS, focusing on user experience and innovative solutions, such as offline-first functionality, integration with map services, and epidemiological information based on data entered in the application. Canine visceral leishmaniasis is a serious infectious disease that affects animals and humans, making it of great importance for public and animal health. The proposed application aims to simplify the process of reporting cases, provide useful and relevant information about the disease and its prevention, and allow the control and monitoring of cases of canine visceral leishmaniasis in the region of Uruguaiiana/RS.

Keywords: Mobile Application; Canine Visceral Leishmaniasis; Disease Control; Epidemiological Monitoring.

1. INTRODUÇÃO

A Leishmaniose Visceral Canina (LVC) é causada por um protozoário do gênero leishmania. Os principais reservatórios do agente são os roedores e os cães, a trans-

* Aluno do Curso de Engenharia de Software da Universidade Federal do Pampa, Alegrete, Rio Grande do Sul, Brasil
E-mail: joaogerman.aluno@unipampa.edu.br

** Orientador, Professor do Curso de Engenharia de Software da Universidade Federal do Pampa, Alegrete, Rio Grande do Sul, Brasil, E-mail: fabio basso@unipampa.edu.br

*** Co-Orientadora, Professora do Curso de Medicina Veterinária da Universidade Federal do Pampa, Uruguaiiana, Rio Grande do Sul, Brasil, E-mail: deborapellegrini@unipampa.edu.br

missão para o invertebrado ocorre pela picada de insetos dípteros pertencentes à família Psychodidae e subfamília Phlebotominae, encontrados nas regiões tropicais, especialmente nas regiões quentes e úmidas do planeta (ALBUQUERQUE; LANGONI, 2018).

No Brasil, a doença é transmitida através da picada do mosquito pertencente à família dos flebotomídeos, ao gênero Lutzomyia e à espécie Lutzomyia longipalpis. Este vetor é conhecido popularmente por mosquito-palha, birigui ou tatuquiras, e se constitui no principal vetor brasileiro. (SCHIMMING; SILVA, 2012).

A LVC é uma doença grave e potencialmente fatal para cães. Além disso, cães infectados podem transmitir a doença para humanos. Portanto é crucial que os casos de infecção sejam identificados e rastreados de forma eficiente para controlar e prevenir a propagação da doença.

Apesar da grande quantidade de sistemas de informação disponíveis atualmente, o controle e notificações dos casos ainda é feito de forma manual em Uruguiana RS. Os dados são preenchidos em documentos de papel e encaminhados para os laboratórios, e quando retornam, são inseridos manualmente em planilhas do google. Esse processo é demorado e suscetível a falhas humanas pois necessita de múltiplos preenchimentos manuais,

1.1. MOTIVAÇÃO

Diante do cenário apresentado, é importante que os casos de LVC sejam identificados e rastreados de forma eficiente. A notificação e o registro dos casos são fundamentais para o planejamento e a implementação de medidas de controle e prevenção da doença. A análise epidemiológica baseada nos dados inseridos no aplicativo pode fornecer informações valiosas sobre os fatores de risco e as tendências da doença, o que é essencial para a tomada de decisões e a implementação de medidas eficazes.

O aplicativo móvel é uma forma eficiente e inovadora de lidar com a doença. O aplicativo será desenvolvido com base em princípios de usabilidade e acessibilidade, para garantir que seja fácil de usar e acessível para os usuários-alvo, incluindo órgãos de controle sanitário e veterinários. Ele incluirá funcionalidades como notificação de casos, registro de dados, geração de relatórios para envio aos laboratórios e análise epidemiológica baseada em dados coletados pelo aplicativo. Essas funcionalidades permitirão que os usuários identifiquem e rastreiem os casos de LVC de forma eficiente, e também permitirão que os órgãos de saúde pública e veterinária compreendam a distribuição e evolução da doença em diferentes áreas.

Além disso, o aplicativo será desenvolvido com segurança de dados em mente, para garantir que os dados dos usuários sejam protegidos e armazenados de forma segura. Isso incluirá medidas de segurança como autenticação de usuário e propriedade de registro dos dados.

Em suma, o desenvolvimento de um aplicativo móvel para notificação de casos de LVC é justificado pela sua importância na saúde pública e animal, e por sua eficiência e inovação ao lidar com a doença. Ele tem o potencial de melhorar significativamente o controle e prevenção da LVC e contribuir para a saúde pública e animal em Uruguai-ana RS.

1.2. PROBLEMA

Uruguaiana é considerada área de transmissão para LV, pois tem ocorrência de casos caninos e humanos da enfermidade e presença do vetor em vários bairros do município (MASSIA et al., 2016). Em levantamentos sorológicos e nos atendimentos às demandas espontâneas realizados no município, no período de 2009 a 2019, a VAS realizou 3.339 investigações para LVC, por demanda espontânea e por inquérito sorológico, sendo que destes, 1548 foram reagentes.(SAÚDE, 2020)

Os cães são os principais reservatórios domésticos da LV. Essa situação pode ser explicada pela prevalência canina ser mais alta quando comparada à humana e, também, pelo elevado número de caninos assintomáticos, podendo representar 80% dos animais infectados. A leishmaniose visceral canina (LVC) é uma doença sistêmica que pode atingir qualquer órgão, tecido ou fluido e que se manifesta por sinais clínicos não específicos como lesões de pele, linfadenomegalia, perda de peso progressiva, onicogribose, atrofia muscular, intolerância ao exercício, perda de apetite, letargia, esplenomegalia, poliúria, polidipsia, lesões oculares, onicogribose, claudicação, vômito e diarreia (SOLANO-GALLEGO et al., 2009).

Nesse sentido, melhorias na coleta de dados caninos são certamente possíveis e aconselháveis. A vigilância da LVC, embora mantendo seu componente passivo atual e melhorando a implementação da amostragem ativa, pode ser adaptada para informar com precisão a frequência de LV. Estudos anteriores mostraram a contribuição dos dados de animais na redução da incerteza em torno das estimativas de risco de área (BOAZ et al., 2018).

Desta forma, a busca por uma melhor compreensão acerca dos principais determinantes associados à doença, à população acometida e ao meio ambiente auxiliará no controle e na prevenção desta enfermidade, servindo de modelo estratégico de controle para outras doenças (REPERANT; CORNAGLIA; OSTERHAUS, 2013).

1.3. OBJETIVOS

A proposta deste trabalho é desenvolver um aplicativo móvel para notificação de casos de LVC, voltado para órgãos de controle sanitário e clínicas veterinárias. Como função principal, o aplicativo deve ser capaz de ajudar na identificação e rastreamento eficiente da doença. Com isso, visando ajudar no controle da propagação da doença e na compreensão da distribuição e evolução em diferentes áreas, possibilitando assim melhores estratégias de controle.

O objetivo geral deste trabalho é desenvolver um aplicativo móvel para controle dos casos de LVC em Uruguaiana/RS, e assim contribuir com o controle epidemiológico da doença na cidade. Os objetivos específicos são: 1 - Permitir a notificação

eficiente dos casos de LVC na cidade de Uruguaiana/RS. 2 - Realizar uma busca por softwares semelhantes em bases de patentes para efeitos de comparação. 3 - Desenvolver funcionalidade de *data mart* (FAN; POULOVASSILIS, 2004) com informações epidemiológicas em forma de gráficos e tabelas para garantir melhor compreensão e tomada de decisão dos órgãos sanitários. 4 - Garantir a segurança dos dados inseridos no aplicativo. 5 - Desenvolver funcionalidade para controle de perfil de usuário e propriedade de registro. 6 - Permitir o funcionamento offline do aplicativo e garantir a sincronização dos dados quando conectado. 7 - Fazer um estudo exploratório sobre o tema deste projeto e avaliar o potencial impacto na saúde pública de Uruguaiana RS.

1.4. METODOLOGIA

A construção do trabalho foi dividida em três partes principais: Projeto, Desenvolvimento e Pesquisa. No primeiro momento serão colocados em prática os conhecimentos aprendidos durante o curso de Engenharia de Software no processo de elicitar os requisitos do sistema e produzir os artefatos da documentação do mesmo. Após essa etapa, será iniciado o desenvolvimento do aplicativo. E por último, será realizado um estudo exploratório nas áreas da veterinária e tecnologia voltadas a doenças relacionadas, para que se possa justificar a importância do mesmo.

No que diz respeito à distribuição de tempo, espera-se passar pelas fases de Projeto e Desenvolvimento durante o TCC1 e pelas fases de Desenvolvimento e Pesquisa no TCC2. A etapa de desenvolvimento será mais longa e por esse motivo estará presente nas duas disciplinas, sendo no TCC1 o momento de cobrir os requisitos com prioridade imprescindível para a utilização do aplicativo e no TCC2 serão realizadas implementações de funções para melhorar os dados epidemiológicos e permitir funcionamento offline.

A Figura 1 mostra em BPMN as principais atividades referentes ao estudo necessário para a conclusão do TCC 2. O primeiro processo é referente ao TCC 1 que já foi defendido anteriormente. O segundo processo ilustra as principais atividades executadas no TCC 2. Destaca-se a implementação dos requisitos complementares que ajudaram a agregar ainda mais valor ao produto final. Também destaca-se a necessidade de atividades que buscam homologar e implantar o software.

No decorrer do TCC2, decidiu-se por realizar apenas uma release, o que mudou um pouco a ordem dos processos apresentados na Figura1 pois não houve homologação 1 e 2. O restante seguiu como planejado.

1.5. ORGANIZAÇÃO

O presente artigo é estruturado em três seções, além da introdução. A primeira seção apresenta o embasamento teórico, que aborda os elementos da engenharia de software utilizados para moldar o processo de definição e implementação do aplicativo PampaCare LVC. Além disso, esta seção discorre sobre informações importantes do projeto PampaCare, como um todo, suas ações extensionistas e requisitos.

A segunda seção é dedicada à apresentação do aplicativo PampaCare LVC, no qual são discutidos os processos práticos utilizados durante o desenvolvimento do mesmo, divididos entre Definição e Implementação. Este capítulo também apresenta seções específicas que enfatizam informações importantes do sistema, como o uso do Firebase, um serviço externo, para atender requisitos essenciais da aplicação.

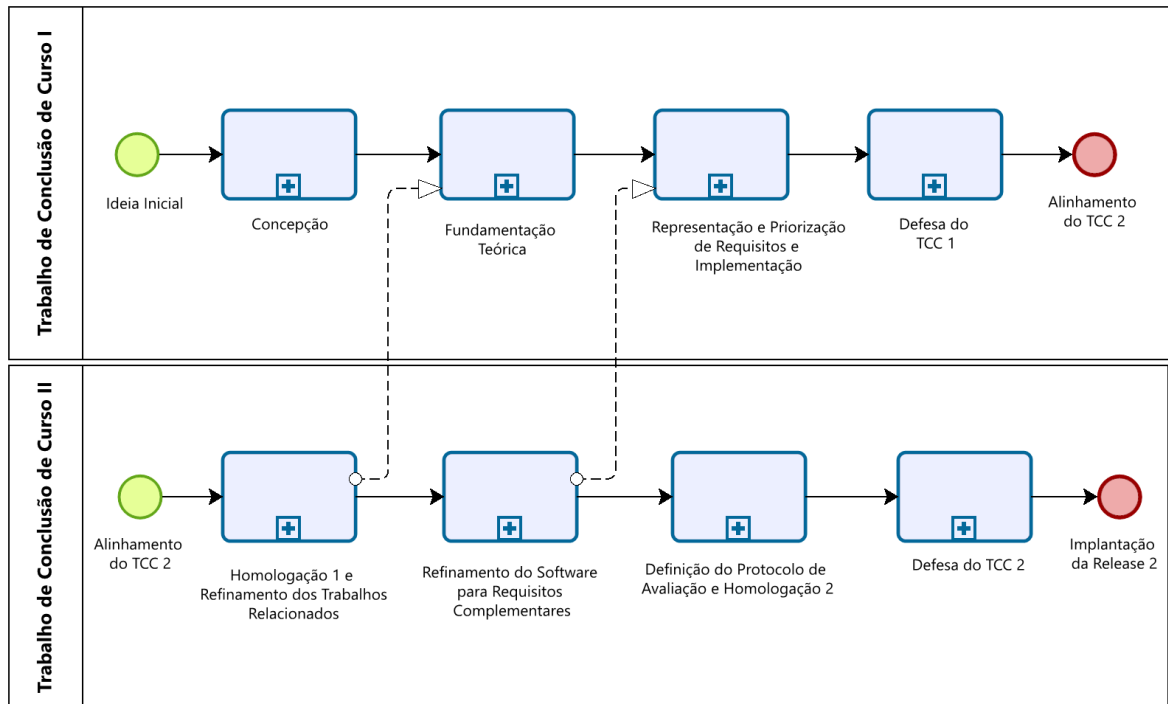


Figura 1 - Processos da Pesquisa.

Por fim, as Considerações Finais apresentam uma análise dos resultados obtidos durante o desenvolvimento do projeto, destacando suas contribuições.

Dessa forma, a estrutura do trabalho segue uma sequência lógica e organizada, apresentando uma análise teórica sólida e detalhes práticos do desenvolvimento do aplicativo, culminando em uma avaliação crítica e reflexiva das contribuições do projeto.

2. FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica é dividida entre: a) elementos da engenharia de software utilizados para a execução do trabalho de conclusão de curso (Seção 2.1); b) elementos para posicionar a metodologia com características de extensão associadas com o trabalho e a posição da presente contribuição para com o projeto maior denominado Pampacare (Seção 2.2), e c) discutir os trabalhos relacionados na Seção 2.3.

2.1. ELEMENTOS DA ENGENHARIA DE SOFTWARE

Para o acompanhamento técnico da implementação da solução para o problema, discussões sobre arquitetura de software são necessárias. Esta seção traz essa discussão como segue.

2.1.1. PROCESSO DE DESENVOLVIMENTO

O processo de desenvolvimento do aplicativo baseou-se no Scrum, apesar de conter apenas alguns elementos deste *framework*, portanto um recorte adaptado para a dimensão do problema à ser resolvido ao longo do TCC. Ele passou por diversas

etapas que foram organizadas em sprints, com o objetivo de entregar funcionalidades a cada ciclo de desenvolvimento.

Segundo (PRESSMAN, 2011), um sprint é um período de tempo fixo, geralmente entre duas e quatro semanas, durante o qual o time de desenvolvimento trabalha em uma série de funcionalidades que devem ser entregues até o final do período. Essas funcionalidades são definidas com base nos requisitos levantados e priorizadas pelo time de desenvolvimento em conjunto com o cliente.

Antes do início do desenvolvimento, foi realizada a elicitación de requisitos e casos de uso. A elicitación de requisitos é uma atividade fundamental para o sucesso do projeto, pois permite a identificação das necessidades dos usuários e as funcionalidades que devem ser implementadas no aplicativo. Segundo (PRESSMAN, 2011), a elicitación de requisitos é a atividade mais crítica em todo o processo de desenvolvimento de software.

Já a definição de casos de uso, segundo (SOMMERVILLE, 2019), é uma técnica de modelagem que descreve como um sistema se comportará em resposta a um conjunto de estímulos. Os casos de uso foram utilizados para descrever as principais funcionalidades do aplicativo, tais como a notificação de casos suspeitos, o registro de informações sobre os exames, o acompanhamento dos casos notificados e relatórios epidemiológicos.

Com base nos requisitos levantados, foram definidos os sprints e as funcionalidades a serem desenvolvidas em cada um deles. As funcionalidades foram priorizadas de acordo com a sua importância para as partes interessadas e para a solução do problema de controle e prevenção da Leishmaniose Visceral Canina, utilizando o método MoSCoW (Oracle, 2000).

Durante cada sprint, foram realizadas atividades de análise, projeto, implementação e testes. A análise consistiu em definir os requisitos de cada funcionalidade a ser desenvolvida e em identificar as possíveis soluções para implementá-las. O projeto consistiu em definir a arquitetura do sistema e a estrutura de classes e componentes necessários para implementar as funcionalidades. Para o desenvolvimento do aplicativo móvel, foram utilizadas as tecnologias Flutter e Firebase. Flutter é um framework desenvolvido pela Google para a criação de aplicativos móveis multiplataforma. Já o Firebase é uma plataforma de desenvolvimento de aplicativos móveis que oferece recursos como autenticação de usuários, armazenamento de dados em tempo real e análise de dados.

O uso de tecnologias modernas como Flutter e Firebase permitiu a criação de um aplicativo móvel robusto e de fácil manutenção. Segundo (PRESSMAN, 2011), a utilização de tecnologias modernas é fundamental para garantir a qualidade e a eficiência do processo de desenvolvimento de software.

2.1.2. TÉCNICAS DE ENGENHARIA DE REQUISITOS

Durante o processo de engenharia de requisitos, é fundamental utilizar técnicas que permitam uma compreensão mais aprofundada das necessidades e expectativas dos usuários e outras partes interessadas envolvidas no projeto. Segundo (PRESSMAN, 2011), a engenharia de requisitos é um processo sistemático para a identificação, análise, validação e documentação das necessidades e restrições do sistema

sendo desenvolvido. (SOMMERVILLE, 2011) destaca que as técnicas de engenharia de requisitos devem ser escolhidas com base nas características do projeto, bem como nas habilidades e experiência da equipe envolvida.

Entre as técnicas mais comuns utilizadas na engenharia de requisitos, podemos citar: 1) Entrevistas com as partes interessadas; 2) Grupos focais; 3) Análise de documentos e registros.

É importante lembrar que cada projeto é único, e que as técnicas de engenharia de requisitos devem ser escolhidas de acordo com as necessidades específicas do projeto em questão. Além disso, é fundamental garantir a validação dos requisitos com as partes interessadas, a fim de garantir que todas as expectativas foram atendidas e que o escopo do projeto está bem definido. Como destaca (PRESSMAN, 2011), a validação é o processo de confirmação de que um conjunto completo e correto de requisitos foi capturado.

No projeto em questão, foi utilizada a técnica de entrevistas com as partes interessadas para extração dos requisitos. Essa técnica é uma das mais comuns e eficazes na engenharia de requisitos.

Além disso, (SOMMERVILLE, 2011) destaca que as entrevistas são particularmente úteis quando se trata de obter informações detalhadas sobre as necessidades e expectativas dos usuários e outras partes interessadas, principalmente porque permitem uma conversa estruturada e interativa, que pode levar a *insights* e informações que não seriam obtidas de outra forma.

Após a escrita dos requisitos, foi realizada a validação com as partes interessadas para fechamento do escopo. Essa é uma etapa crucial na engenharia de requisitos, que deve ser realizada para garantir que todos os requisitos foram compreendidos corretamente e que não há inconsistências ou ambiguidades.

A técnica de análise de documentos também foi utilizada, uma vez que foi possível ter acesso a ficha de investigação de LVC utilizada oficialmente no estado, e desta forma, elicitamos quais eram as informações mais importantes a se conter em uma notificação de casos da doença.

2.1.3. MÉTODO DE PRIORIZAÇÃO DE REQUISITOS MoSCoW

Inventado por Dai Cleg (Oracle, 2000), o método MoSCoW é utilizado para priorização de requisitos. O método consiste em 4 diferentes classificações, sendo elas:

- **Must have:** Requisitos extremamente necessários para o funcionamento do projeto, não podem ser deixados de fora.
- **Should have:** Requisitos importantes e capazes de agregar valor ao projeto, mas não necessários para o funcionamento do fluxo principal do mesmo.
- **Could have:** Requisitos capaz de agregar valor ao projeto, mas que não possuem grande impacto final.
- **Won't have for now:** Requisitos deixados de fora do projeto atual, mas que seria interessante ter no futuro.

Após passar pelo processo de priorização, foi decidido por implementar no TCC1 os requisitos tidos como Must have. No TCC 2, defendido no presente artigo, os requisitos tidos como Should have e Could have foram implementados, além de correções remanescentes do TCC 1. Essas implementações foram capazes de agregar ainda mais valor ao projeto, deixando o aplicativo perfeitamente alinhado às necessidades descritas durante o processo de elicitação de requisitos.

Na figura 2, é possível observar os requisitos que ficaram para TCC2 na planilha de controle dos requisitos.

Cód.	Nome	Descrição	Tipo	MoSCoW	Status
4	Listagem de usuários	O aplicativo deve possuir uma tela de listagem de usuários que será visualizada apenas pelo administrador	Funcional	Should Have	Realizado
5	Dados de usuário logado	O usuário logado, independente do tipo, deve ter acesso para alterar suas informações de cadastro e excluir a sua conta.	Funcional	Should Have	Realizado
12	Cadastro de testes/exames	O aplicativo deve possuir formas de inserir testes/exames nos cadastros de notificações de casos.	Funcional	Should Have	Realizado
13	Exportar PDF	O aplicativo deve possuir funcionalidade de exportar uma notificação de caso em PDF.	Funcional	Should Have	Realizado
14	Funcionamento offline	Os cadastros de proprietário, cachorro e notificação devem funcionar de forma offline, tendo seus dados sincronizados com o banco de dados quando conectado à internet.	Funcional	Should Have	Pendente
16	Dados epidemiológicos	O aplicativo deve conter relatórios epidemiológicos baseados nos dados de notificações de casos existentes.	Funcional	Should Have	Realizado
17	Dashboard	O aplicativo deve possuir uma tela de dashboard com gráficos para demonstração dos dados epidemiológicos de forma consolidada.	Funcional	Should Have	Realizado
18	Cadastro, edição e exclusão de notificação de casos	O usuário deve ser capaz de inserir os dados de proprietário, cachorro e notificação tanto no mesmo formulário, como de forma individual.	Funcional	Could Have	Pendente

Figura 2 - Planilha de controle dos requisitos

2.1.4. CAMADA DE BACKEND DA APLICAÇÃO

O Firebase é uma plataforma de Backend-as-a-Service (BaaS) do Google que oferece um conjunto completo de soluções para os desenvolvedores que vão desde autenticação de usuários até análise de dados (GOOGLE, 2023a), os recursos incluem: 1) Autenticação (Authentication); 2) Armazenamento em nuvem (Cloud Firestore); 3) Banco de dados em tempo real (Realtime Database); 4) Análises e notificações (Analytics e Cloud Messaging); 5) Funções na Nuvem (Cloud Functions).

Para solucionar o problema apresentado, as funções utilizadas do Firebase foram a Authentication, Cloud Firestore e Cloud Functions. O uso do Firebase no desenvolvimento de aplicativos móveis traz diversas vantagens comparados a um backend comum, principalmente no que diz respeito ao tempo de desenvolvimento.

2.1.5. CAMADA DE FRONTEND DA APLICAÇÃO

Na camada de frontend, foi desenvolvida uma solução para smartphones que será disponibilizada para Android e IOS. Para que fosse possível disponibilizar em ambas as plataformas, foi utilizado o framework Flutter no processo de desenvolvimento, o que tornou mais rápido e eficiente a geração do resultado esperado.

O Flutter é um framework desenvolvido pela Google para a criação de aplicativos móveis, web e desktop. Ele permite criar aplicativos móveis para Android e iOS a partir de um único código-fonte, utilizando uma linguagem de programação chamada Dart. O Flutter oferece uma abordagem diferente da maioria dos frameworks de desenvolvimento móvel, permitindo que os desenvolvedores construam interfaces de usuário customizadas e com alto desempenho (GOOGLE, 2023b).

O Flutter utiliza *widgets* como blocos de construção para criar a interface de usuário da aplicação. Os *widgets* no Flutter são classes em Dart que são usadas para criar e gerenciar elementos visuais, como botões, campos de entrada, listas e muitos outros elementos. Os *widgets* podem ser combinados em uma hierarquia para criar a interface de usuário da aplicação. (FLUTTER, 2023)

Além disso, o Flutter também permite criar *widgets* personalizados, que podem ser ajustados para atender às necessidades específicas da aplicação. Essa flexibilidade permite que os desenvolvedores criem interfaces de usuário complexas e customizadas, sem comprometer o desempenho ou a usabilidade da aplicação.

Durante o TCC2, fora do que foi elicitado anteriormente, surgiu a necessidade de redesenhar a identidade visual do app para que estivesse nos padrões da unipampa e o pampacare no que diz respeito a cores e ícones. O Flutter foi um grande aliado nesse processo, tornando mais rápida essa atualização.

2.2. PROJETO PAMPACARE E SISTEMAS DE INFORMAÇÕES DISTRIBUÍDOS

O objetivo do projeto Pampacare é desenvolver uma plataforma de gerenciamento de dados que proporcione o monitoramento, integração e análise de informações referentes à saúde humana, animal e ambiental para fortalecer as ações de vigilância, controle, mitigação e prevenção de pandemias e endemias. Segundo os proponentes do projeto, o acesso às informações e indicadores, que são obtidos de diversas origens (diversos apps tratando de doenças específicas), possibilitará a adoção de ações capazes de atenuar e até impedir a expansão de doenças.

De modo a promover esta colaboração por meio de uma circulação eficiente da informação, o projeto Pampacare propõe o desenvolvimento de sistemas de informações descentralizados, que podem ser desenvolvidos utilizando de diversas tecnologias. Neste contexto, o problema de LVC demanda o desenvolvimento de um sistema focado, que será integrado posteriormente ao sistema de escopo maior denominado Pampacare. O Pampacare, portanto, é um software que opera sobre os conceitos de *data warehouse*, coletando dados de sistemas independentes para gerar novas análises de tendências de infestações, suas correlações, dentre outras necessidades de monitoramento.

Como resultados esperados do Pampacare, destaca-se a potencialização das colaborações interorganizacionais e agilidade na difusão de dados para tomada de decisão pelas autoridades públicas competentes.

2.2.1. AÇÕES EXTENSIONISTAS DO PROJETO

Tipicamente, disciplinas de resolução de problemas são fundamentadas em práticas de Aprendizado Baseado em Problemas (ABP)(DOURADO, 2015), que demandam dos alunos o engajamento em times para a resolução de problemas práticos propostos pelos professores. A disciplina de Resolução de Problema 5 tem como objetivo promover ações curriculares de extensão no curso de Bacharel em Engenharia de Software, exercitando conhecimentos voltados para práticas de manutenção e evolução de software em times de desenvolvimento de software. O software apresentado neste TCC tem origem como demanda desta disciplina.

Na edição de 2021 alunos de cada grupo interagiram com os agentes externos para a captura a validação de requisitos do software. Como expectativa dos professores e dos agentes externos, a aplicação de um processo de modernização resultou em um sistema do tipo *Data Mart*, que se somou à um sistema maior para *Data Warehouse* denominado Pampacare. O objetivo dos grupos foi aplicar um processo de modernização baseada em conceitos de *Data Warehouse* para ETL (Extract, Transformation and Load)(FAN; POULOVASSILIS, 2004), exercitando assim elementos da ementa da disciplina como Reengenharia de software e Processo de evolução. Também foi objetivo o desenvolvimento de um software para cadastro dos dados, eliminando assim o registro por planilhas eletrônicas utilizados pelos profissionais da saúde.

2.2.2. AGENTES EXTERNOS E SISTEMAS DE INFORMAÇÕES REQUISITADOS

Debora da Cruz Payao Pellegrini e Laura Ilarraz Massia - Débora é Professora e Pesquisadora da Unipampa do Campus de Uruguaiiana, e Laura é sua doutoranda. Requisitaram o desenvolvimento do software para LVC em 2021 e também outros dois softwares em 2022: Sistema de informação para Senso canino, e o LV incidente em humanos. O **Pampacare LVC** trata-se do sistema de informação cuja pesquisa exploratória e aplicada é discutida neste documento. Ele está focado em fomentar um sistema de informações que servirá para a execução de ações de vigilância, controle, mitigação e prevenção do LVC.

Debora da Cruz Payao Pellegrini e Rafaela de Carvalho Machado - Rafaela pesquisadora doutoranda do campus de Uruguaiiana. Em 2022 requisitaram o desenvolvimento do software para registro e relatório de dados de Aedes Aegypti (Dengue). Este software está em desenvolvimento por outro aluno de graduação, também em condução como tema de trabalho de TCC, e é denominado como **Pampacare Dengue**.

Jenifer Harter e Rhayanna de Vargas Perez - Jenifer é Professora Pesquisadora da área de saúde humana atuante no campus de Uruguaiiana, e Rhayanna é sua doutoranda. Em 2022 requisitaram o desenvolvimento do software para registro e relatório de dados de Tuberculose, o **Pampacare TB** (Pampacare Tuberculose).

2.3. TRABALHOS RELACIONADOS

O presente capítulo tem como objetivo apresentar trabalhos acadêmicos relacionados ao aplicativo proposto neste trabalho, buscando comparar suas soluções e analisar suas semelhanças e diferenças em relação ao aplicativo proposto. Embora o tema da leishmaniose seja específico, existem propostas de aplicativos semelhantes

que visam resolver o mesmo problema principal: a notificação de casos de leishmaniose visceral canina.

Um exemplo de aplicativo semelhante é o App C7 LVC (VASCONCELLOS et al., 2021) (ver a Figura 5), que tem como objetivo notificar os casos de LVC na cidade de Santa Maria/RS. Embora este seja um bom exemplo de aplicativo voltado para o controle de leishmaniose, existem poucos trabalhos acadêmicos sobre o assunto e muitos deles se concentram em soluções para o controle da leishmaniose em humanos (RUBIANO et al., 2021) e (NADRI et al., 2020).

Outro aplicativo relacionado encontrado é o "Leishmania Detection"(NAKAMURA, 2022), desenvolvido como Trabalho de Conclusão de Curso em outra universidade. Esse aplicativo tem um foco principal diferente, que é realizar o cálculo do índice de infecção utilizando visão computacional. Apesar do foco distinto, a solução proposta utiliza as mesmas tecnologias escolhidas para o Pampacare LVC: Flutter e Firebase.

Com o objetivo de encontrar softwares semelhantes, também foram realizadas buscas por registros no Instituto Nacional da Propriedade Intelectual (INPI) buscando por soluções semelhantes. O método de busca utilizado foi: Qualquer uma das palavras: 'LEISHMANIOSE LEISHMANIA LVC no TítuloPrograma'. Foram encontrados 3 softwares registrados, sendo eles:

1) "Aplicativo de Registro de Casos Suspeitos de Leishmaniose Canina e Humana - LEISH APP"(SILVA et al., 2021)

2) "Leishmania App - Cálculo d Índice de Infecção de Macrófagos por Leishmanias"(SOUSA, 2020)

3) "LVC"(TAVARES; MATTOS, 2002)

Ao analisar as soluções propostas em trabalhos acadêmicos relacionados, observa-se que a maioria delas se concentra na resolução do problema em si, sem levar em consideração a experiência do usuário. Diferentemente dos trabalhos citados, a solução proposta nesta monografia tem um foco especial na experiência do usuário, que se traduz em uma interface moderna e intuitiva. O design das telas do aplicativo foi pensado para ser intuitivo e fácil de usar, proporcionando uma experiência agradável para o usuário.

Além disso, o aplicativo proposto neste trabalho apresenta funcionalidades únicas, como o funcionamento offline, que permite que o usuário possa utilizá-lo mesmo sem conexão com a internet, e a análise de dados dos casos notificados pelo aplicativo, o que pode ajudar no planejamento e tomada de decisões para o controle da leishmaniose.

A presente contribuição também pode ser comparada com os resultados das ações de extensão conduzidas na disciplina de Resolução de Problemas 5. O mesmo problema foi explorado em 2021 por 24 alunos de graduação, divididos em cinco grupos, ao longo da disciplina de Resolução de Problemas 5, do curso de Engenharia de Software da Unipampa no Campus de Alegrete. Como resultado da execução da disciplina, interfaces de usuário foram desenvolvidas para plataforma Mobile (ver a Figura 3) e para Web (ver a Figura 4). Apesar da tentativa dos professores que os alunos desse sequência após o término da disciplina, nenhum dos softwares produzidos foram concluídos. Portanto, optou-se pela reimplementação na proposta atual.

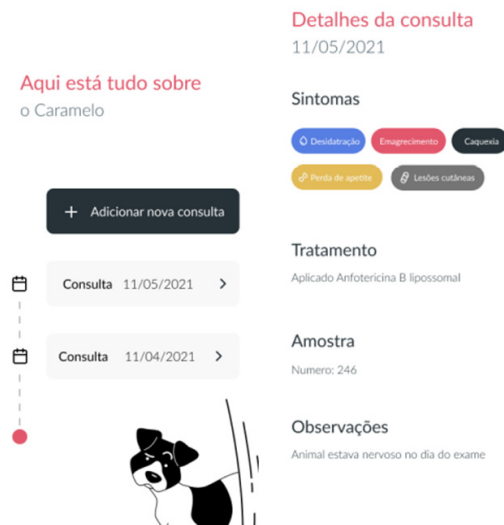


Figura 3 - Duas Telas em Dispositivo Mobile para o Diagnóstico da Doença, Desenvolvidas por um Grupo de Alunos em Flutter numa Versão Depreciada.

Figura 4 - Tela em Browser, Desenvolvida por Outro Grupo de Alunos para o Cadastro das Informações em uma Arquitetura em Microsserviços e Plataforma Web.

Os softwares produzidos em Resolução de Problemas 5 na edição de 2021 foram rotulados como sistemas distribuídos, uma vez que utilizou-se de arquitetura em microserviços para integrar o front-end e o back-end. Já a arquitetura do sistema atual é denominada como Back-end as a Service (BaaS), uma vez que utiliza uma abordagem baseada em banco de dados não relacional operado em nuvem. Outra diferença é que o modelo de dados é o modelo relacional, que nos resultados da disciplina encontravam-se desnormalizados, como é possível visualizar na estrutura da tela mostrada na Figura 4, não refletindo as reais necessidades de registro de informações associadas com o animal, seu proprietário e seus exames. Isso implicava em redundância do cadastro de informações, problema que este trabalho superou mesmo adotando um modelo não relacional. Outra diferença para com os softwares produzidos em Resolução de Problemas 5 é quanto à propriedade dos dados registrados, o que exigiu no sistema aqui proposto o gerenciamento de perfis de usuários e suas autorizações sobre a visualização de dados.

Durante o processo de escrita do presente trabalho, o aplicativo proposto foi tema de mais duas publicações de pesquisadores relacionados ao mesmo, sendo elas:



Figura 5 - Telas do aplicativo C7 LVC. (VASCONCELLOS et al., 2021)

1) "Aplicativo de vigilância e monitoramento de leishmaniose visceral canina (PAMPACARE LVC) - uma abordagem de saúde única em Uruguai (RS)", submetido por Laura Ilarraz Massia ao Conselho Editorial da revista Vigilância Sanitária em Debate – Sociedade, Ciência & Tecnologia (Visa em Debate).

2) "Pampacare LVC: Uma Demanda da Vigilância Sanitária Resultando em Objetos de Aprendizagem na Engenharia de Software", submetido por Prof. Dr. Fabio Paulo Basso ao Simpósio Brasileiro de Engenharia de Software 2023.

Os trabalhos desenvolvidos demonstram a importância do aplicativo que foi construído e seu possível impacto na comunidade, trazendo ainda mais relevância ao presente trabalho.

3. APLICATIVO PAMPACARA LVC

Esta seção abordará o processo de definição, implementação e implantação do aplicativo Pampacare LVC, subdividido em várias subseções para uma melhor compreensão e organização das informações apresentadas.

3.1. DEFINIÇÃO

Antes de iniciar a implementação da solução proposta, foi realizado um processo de definição do problema, composto por três fases principais: coleta de requisitos, especificação dos casos de uso e definição da arquitetura.

Na fase de coleta de requisitos, foram realizadas entrevistas com as partes interessadas e coletadas as fichas manuais utilizadas no processo de notificação dos casos de LVC atualmente. Nessa etapa, foram definidos os requisitos e sua priorização conforme o MoSCoW, definindo assim a sua divisão entre TCC1 e 2.

No total, foram especificados 22 requisitos, considerando tanto os funcionais quanto os não-funcionais, que foram distribuídos em sprints de desenvolvimento e divididos entre os períodos de TCC 1 e TCC 2.

Na parte de especificação dos casos de uso, os principais requisitos do sistema foram detalhados e foi criada uma descrição estendida dos casos de uso, que inclui

conceitos sobre como a tela deve funcionar e suas condições de operação, com base nos requisitos estabelecidos. No processo de definição do aplicativo, a última etapa foi a definição do conjunto de tecnologias e arquitetura a serem utilizadas para a resolução do problema em questão. A seleção desses elementos deve levar em consideração diversos fatores, como a disponibilidade do aplicativo em diferentes plataformas e a eficiência do desenvolvimento.

Com base nesses requisitos, foi escolhida uma tecnologia de desenvolvimento híbrido multiplataforma, o Flutter, que possibilita a construção do aplicativo para Android e IOS com apenas um código, reduzindo assim o tempo de desenvolvimento sem abrir mão da qualidade.

A escolha da tecnologia de persistência de dados também é crucial para o sucesso do projeto. Foi selecionado o Cloud Firestore, um serviço de banco de dados NoSQL da plataforma Firebase, que permite rápida configuração e é escalável, garantindo que o aplicativo possa lidar com grandes volumes de dados sem perda de desempenho. Além disso, o Firebase oferece outras ferramentas que podem ser usadas em conjunto com o Cloud Firestore, como o Authentication, que permite uma autenticação segura e fácil para os usuários do aplicativo.

É importante destacar que a escolha dessa stack de desenvolvimento permitiu um desenvolvimento rápido e eficiente, mesmo com apenas um desenvolvedor, com foco na experiência do usuário sem abrir mão da qualidade no controle dos dados e segurança na autenticação.

Mais detalhes sobre o Firebase e sua integração com o aplicativo serão apresentados em uma seção exclusiva, na qual serão explorados em detalhes as funcionalidades do serviço e a sua importância para o projeto.

3.2. IMPLEMENTAÇÃO

O processo de desenvolvimento do aplicativo, conforme mencionado anteriormente, foi planejado e dividido em períodos distintos para atender às disciplinas de TCC 1 e 2. Para garantir uma gestão eficiente e uma visão clara do progresso do projeto e priorização dos requisitos, foi criada uma planilha de controle dos requisitos do sistema, permitindo um acompanhamento mais detalhado das etapas de desenvolvimento. Um trecho dessa matriz pode ser visualizado na Figura 2.

Durante o período de TCC 1, foi dada prioridade ao desenvolvimento dos requisitos com maior importância e prioridade para garantir que o aplicativo pudesse ser implementado e utilizado pelas partes interessadas em um modelo de produto mínimo viável ao final desse período. Para o período de TCC 2, foram selecionados os requisitos que adicionam valor e complementam ainda mais a solução. Alguns desses requisitos são apresentados na Tabela 1.

Embora nenhum dos requisitos definidos para o período do TCC 2 seja essencial para que o aplicativo cumpra suas funções básicas, essas funcionalidades têm um valor significativo e agregam muito à solução final. De fato, elas podem ser o diferencial que torna essa solução única e inovadora em relação às alternativas disponíveis no mercado, pois o aplicativo tem como foco proporcionar a melhor experiência do usuário e tornar o processo de notificação menos burocrático e demorado, visando aumentar a aderência e, conseqüentemente, o controle da doença.

Tabela 1 - Exemplo de requisitos especificados

Requisito	Descrição
RQ17	O aplicativo deve possuir funcionalidade de exportar uma notificação de caso em PDF
RQ18	O usuário deve ser capaz de inserir os dados de proprietário, cachorro e notificação tanto no mesmo formulário, como de forma individual
RQ19	O aplicativo deve conter relatórios epidemiológicos baseados nos dados de notificações de casos existentes
RQ20	O aplicativo deve possuir uma tela de dashboard com gráficos para demonstração dos dados epidemiológicos de forma consolidada
RQ21	Os cadastros de proprietário, cachorro e notificação devem funcionar de forma offline, tendo seus dados sincronizados com o banco de dados quando conectado à internet

No que diz respeito ao andamento do projeto, ao final do TCC2, todos os requisitos Must Have estão implementados, assim como a maioria dos requisitos classificados como Should Have, desta forma, o app está pronto para ser implantado e testado pelos usuários. Houve uma mudança de cronograma no que diz respeito a implantação, antes prevista para acontecer em duas releases, agora será feita ao final do TCC2 de uma vez só.

3.2.1. AUTENTICAÇÃO E PROPRIEDADE DE REGISTRO

A implementação do requisito de autenticação de usuários foi desenvolvido utilizando o Firebase authentication, inicialmente utilizando cadastro por e-mail e senha. A utilização do serviço do Firebase permitiu uma rápida implementação desse requisito, além de permitir fácil adição de novas formas de login se necessário futuramente.

Junto ao requisito de autenticação, o aplicativo necessitava de um controle de propriedade de registro e níveis de usuário. As regras de acesso para cada nível de usuário podem ser observados na Tabela 2

Tabela 2 - Permissões de Acesso

Operação	Admin	Usuário	Visualizador
Visualizar	Sim	Apenas próprios	Sim
Cadastrar	Sim	Apenas próprios	Não
Atualizar	Sim	Apenas próprios	Não
Excluir	Sim	Apenas próprios	Não

Para a implementação desse requisito, foi utilizado o atributo authorId no cadastro de cada registro, assim como uma verificação do nível de acesso do usuário nas regras do Cloud Firestore.

3.2.2. CLOUD FIRESTORE

O Cloud Firestore é um banco de dados NoSQL fornecido pelo Firebase, que oferece armazenamento e sincronização de dados em tempo real. É altamente escalável e flexível, proporcionando acesso rápido e eficiente aos dados, juntamente com recursos avançados de consulta e segurança.

No Firestore, os dados são organizados em coleções e documentos. As coleções agrupam documentos relacionados, enquanto os documentos são unidades de armazenamento de dados no formato JSON. Cada documento tem um ID exclusivo, gerado automaticamente ou definido pelo desenvolvedor, e pode conter objetos aninhados para suportar estruturas complexas.

Na Figura 6, é possível visualizar de que forma os dados estão dispostos na base de dados.

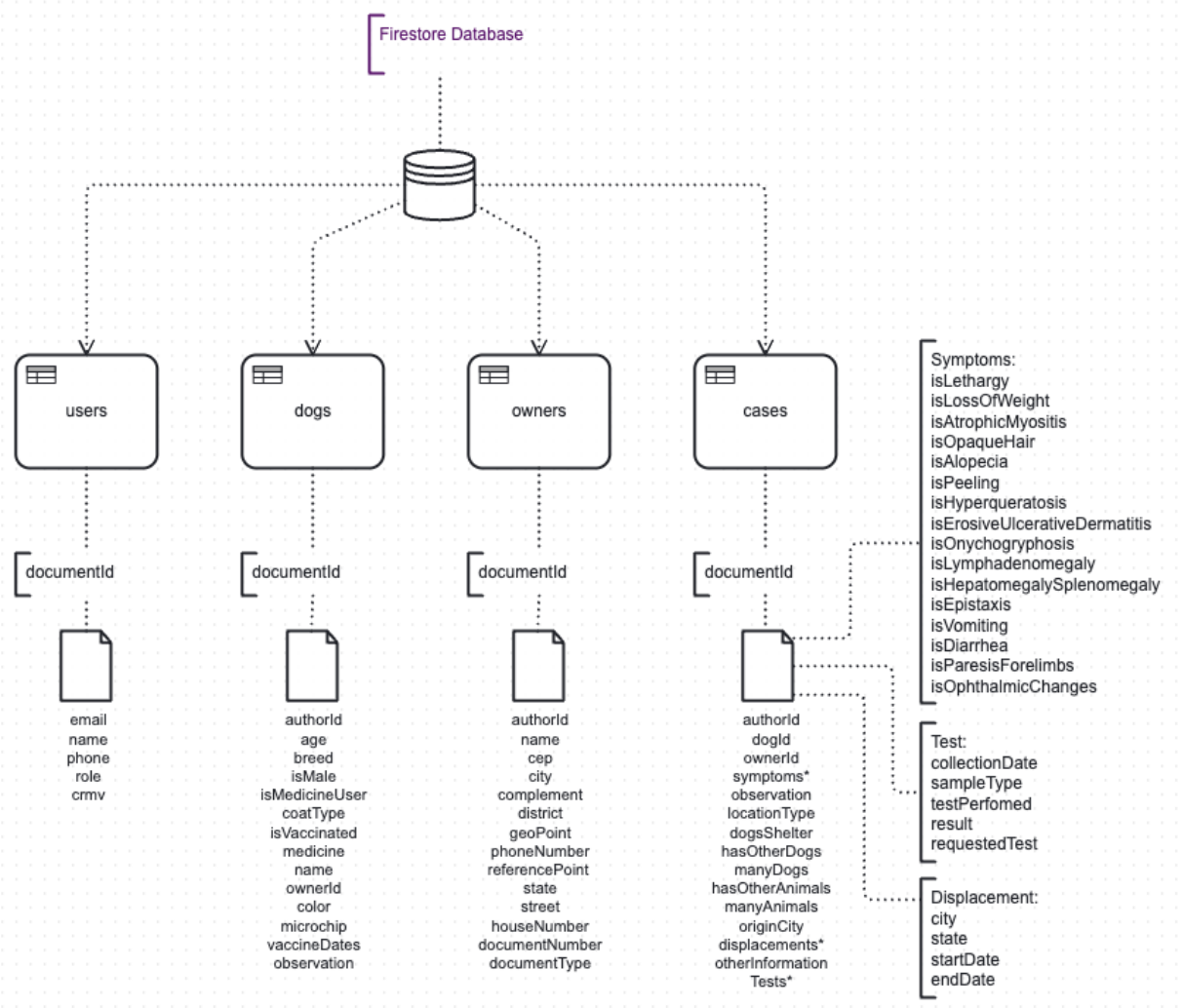


Figura 6 - Organização dos dados no Firestore

O Firestore permite a definição de regras de segurança que controlam o acesso aos dados, garantindo que apenas usuários autorizados possam ler e gravar informações. Essas regras, escritas em linguagem de programação, permitem personalizar o nível de acesso com base em critérios como autenticação, identificação do usuário e tipo de operação. Podem ser aplicadas em níveis de coleção, documento e campo, restringindo o acesso a dados específicos. Essas regras foram utilizadas em conjunto com a autenticação para garantir que o acesso não seja realizado por usuários não autorizados.

3.2.3. CLOUD FUNCTIONS

As Cloud Functions do Firebase são unidades de código executadas em resposta a eventos específicos e acionados por serviços do Firebase ou eventos do Google Cloud Platform. Elas são uma forma eficiente de implementar lógica de servidor sem precisar gerenciar uma infraestrutura de servidores.

Visando maior segurança, apenas usuários com permissão de admin podem criar novos usuários no sistema, desta forma, foi utilizado Cloud Functions para que fosse possível essa implementação no Firebase. Ao definir o email e o nível de acesso do novo usuário, a function é responsável por inseri-lo no Firebase Authentication assim como criar um registro para o mesmo na coleção de usuários. A senha padrão é "123456", que deve ser alterada pelo usuário no seu primeiro acesso.

No que se trata das informações epidemiológicas, uma function é responsável por coletar os dados nas coleções do firestore e retornar pro aplicativo apenas os dados necessários para construir os gráficos, poupando assim o aplicativo de precisar requisitar todas as notificações presentes no Firestore e passando esse trabalho ao servidos. Além disso, também torna possível mudar os parâmetros que calculam os valores dos gráficos sem precisar lançar uma nova versão do aplicativo.

Fora o que foi citado, as functions estão sendo utilizadas no projeto também para exclusão de usuários, seja ela realizada pelo admin, que pode excluir qualquer usuário, ou por qualquer outro usuário que tem o direito apenas de excluir a própria conta. Ao excluir um usuário, os registros por ele inseridos não sofrem alterações, porém, só poderão ser visualizados por um admin uma vez que o proprietário do registro não existe mais.

3.3. GEOLOCALIZAÇÃO

Uma das exigências do sistema era que o cadastro do proprietário do animal incluísse informações geográficas precisas, a fim de facilitar o rastreamento e mapeamento dos casos da doença. Atualmente, essa tarefa é realizada manualmente, com os profissionais de saúde buscando as coordenadas geográficas no Google Maps e as inserindo nas fichas de notificação. Com a utilização do aplicativo, é possível coletar a localização atual por meio do GPS do próprio dispositivo móvel e adicioná-la diretamente ao cadastro, além de permitir o cadastro manual, caso necessário.

O aplicativo também oferece uma funcionalidade adicional que permite ao usuário visualizar a localização do proprietário do animal e do caso suspeito no mapa, por meio de um botão de redirecionamento para o Google Maps. Essas informações geográficas são valiosas para a construção de relatórios epidemiológicos que podem ser utilizados para fins de controle sanitário na cidade de Uruguaiana/RS.

O uso do aplicativo simplifica o processo de coleta e registro de informações geográficas, além de fornecer informações em tempo real para a equipe de saúde. Além disso, o uso do GPS permite coletar informações precisas de localização, melhorando a eficiência do controle de doenças e permitindo a identificação de áreas de maior risco.

3.4. DEMONSTRAÇÃO DO APP

Serão exibidas algumas das telas do aplicativo que foram desenvolvidas durante o TCC2, juntamente com explicações sobre os fluxos de utilização do aplicativo.

A Figura 7 exibe o fluxo principal do aplicativo, onde o usuário é capaz de gerenciar as notificações registradas, assim como registrar novas.

Já a Figura 8 exibe o fluxo de gerenciamento de usuários, sendo a própria conta ou a conta de terceiros na perspectiva do admin.

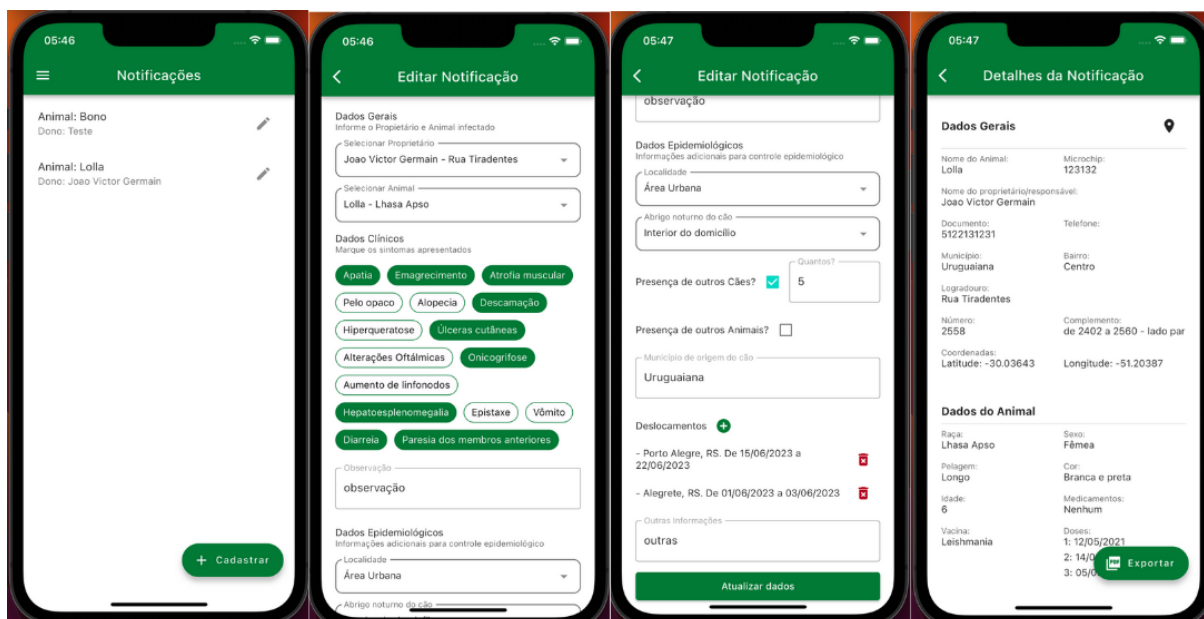


Figura 7 - Fluxo de Notificações

3.5. AVALIAÇÃO DE EXPERIÊNCIA DO USUÁRIO

Para realizar uma avaliação da experiência do usuário ao usar o aplicativo, foi utilizada a técnica User Experience and Usability Guidelines for Agile Project (UXUG-AP) (SOUSA; VALENTIM, 2021). Esta técnica, segundo (SOUSA; VALENTIM, 2021), foi construída com a intenção de dar suporte ao projeto da Usabilidade e UX durante o ciclo de vida do processo de desenvolvimento ágil. A técnica é dividida em 11 categorias, onde cada categoria apresenta suas respectivas subcategorias e cada subcategoria possui diretrizes que apoiam o projeto da Usabilidade e da UX em projetos ágeis.

Após passar pelo processo proposto na técnica, através de uma autoavaliação, chegou-se no resultado apresentado na Tabela 3. Como pode-se observar, o aplicativo cumpriu com 76% das subcategorias, ou seja, possui um desempenho satisfatório no que diz respeito a experiência de usuário e usabilidade. Para atualizações, em breve, deseja-se alcançar 92% de cobertura, de modo a restar apenas a parte de Acessibilidade como não atingido para ser feito no futuro.

4. CONSIDERAÇÕES FINAIS

A solução proposta por meio deste aplicativo visa suportar aos profissionais da saúde funcionalidades para registro de dados relativos à diagnóstico e tratamento da

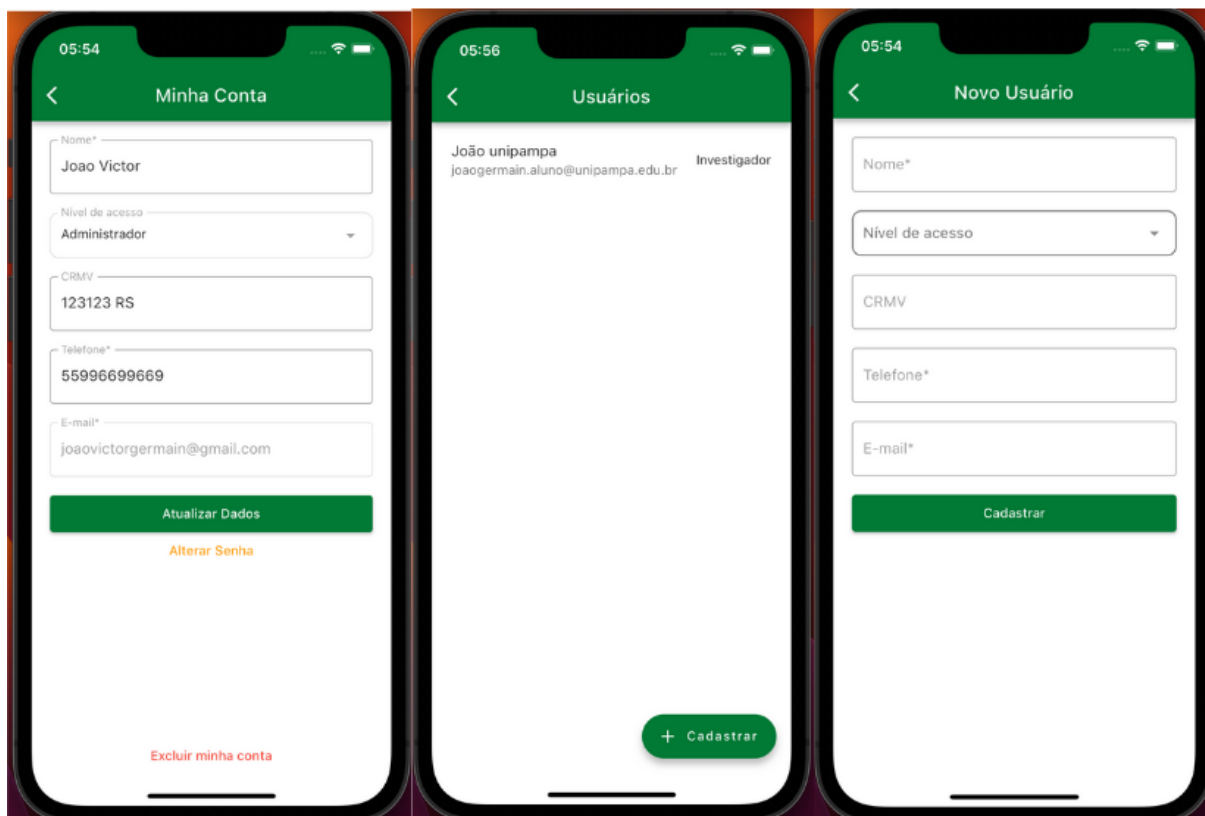


Figura 8 - Fluxo de Usuários

leishmaniose visceral canina (LVC), além de servir como base de dados para órgãos sanitários no controle mais efetivo da doença. Este projeto tem como objetivo principal contribuir significativamente para o controle da doença em Uruguaiana/RS.

A versão final do aplicativo, com todas as funcionalidades elicítadas está prevista para lançamento até o final de julho. As funcionalidades da versão final incluem o funcionamento offline e os relatórios epidemiológicos, que serão cruciais para as pesquisas de campo em locais sem conexão com a internet e para a análise dos dados consolidados posteriormente. Além disso, funcionalidades como a exportação em PDF permitirão o envio dos dados para cidades que não utilizam o aplicativo.

Por fim, o presente trabalho contribui para a realização da principal meta para com os agentes externos de Resolução de Problemas 5 de 2021: concluir o suporte ferramental proposto. Das ações extensionistas é esperado nos cursos de Ciência da Computação e de Engenharia de Software a sequência das ações após execução de disciplinas de Resolução de Problemas, alocando alunos de graduação e pós-graduação da Unipampa no processo construtivo, que visa formar mão de obra especializada. O proponente deste trabalho participou daquela edição da disciplina de 2021 e deu sequência num estudo mais aprofundado do tema. Logo, este trabalho deu o seguimento extensionista almejado, demonstrando ações sequentes que visaram dar suporte aos colaboradores/agentes externos, integrando ações da área da Saúde com a área de estudo do presente TCC.

Tabela 3 - Técnica UXUG-AP

Categorias	Subcategorias	Possui
Requisitos	Troca de Informações - Time/Cliente	Sim
	Entrevistas e Workshops	Sim
	Requisitos Chave	Sim
Entendimento das Necessidades do Usuário	Iniciantes e Especialistas	Sim
	Crianças, Jovens, Adultos e idosos	Sim
	Leigos, Acadêmicos e Profissionais	Sim
Acessibilidade	Deficientes Visuais I	Não
	Deficientes Visuais II	Não
Facilidade no Uso	Localização	Sim
Feedback Informativo	Mensagens de Alerta e Confirmação	Não
	Mensagens de Erro	Não
	Componentes de Carregamento	Sim
	Títulos e Links	Sim
Prevenção de Erros	Campos obrigatórios	Não
	Limitando campos	Não
	Apresentação Autoexplicativa	Sim
Agrupamento de Informação	Independência da Informação	Sim
	Modularização da Informação	Sim
Sequência de Ações	Organização das Ações Sequenciais	Sim
	Comportamento das Ações Sequenciais	Sim
Sentimento de Pertencimento	Conectividade Emocional	Sim
Grau de Importância	Posição da Informação	Sim
	Termos Relevantes	Sim
Privacidade	Controle de Informação	Sim
	Senhas	Sim

REFERÊNCIAS

ALBUQUERQUE, A. L. H. d.; LANGONI, H. A prática do tratamento na leishmaniose visceral canina (lvc) em clínicas veterinárias, cuidados e protocolos. **Veterinária e Zootecnia**, v. 25, n. 1, p. 132–141, jun 2018. ISSN 0102-5716.

BOAZ, R. et al. Integration of animal health and public health surveillance sources to exhaustively inform the risk of zoonosis: an application to visceral leishmaniasis data in brazil. **Spat Spatio-Temporal Epidemiol**, v. 29, p. 177–185, 2018.

DOURADO, S. C. de Souza e L. Aprendizagem baseada em problemas (abp): Um método de aprendizagem inovador para o ensino educativo. **HOLOS**, v. 5, n. 0, p. 182–200, 2015. ISSN 1807-1600.

FAN, H.; POULOVASSILIS, A. Schema evolution in data warehousing environments – a schema transformation-based approach. In: **Conceptual Modeling – ER 2004**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 639–653.

FLUTTER. **Widgets**. 2023. <<https://flutter.dev/docs/development/ui/widgets>>.

GOOGLE. **Firestore**. 2023. <<https://firebase.google.com/>>.

GOOGLE. **Flutter**. 2023. <<https://flutter.dev/>>.

MASSIA, L. et al. Leishmaniose visceral canina em três bairros de uruguaiana–rs. **Vigilância Sanitária em Debate: Sociedade, Ciência & Tecnologia**, v. 4, n. 1, p. 113–119, 2016. ISSN 2317-269X. Disponível em: <<https://visaemdebate.incqs.fiocruz.br/index.php/visaemdebate/article/view/679/298>>.

NADRI, K. et al. Designing a mobile-based self-care application for patients with cutaneous leishmaniasis: An effective step in patients' self-care and participation. **Frontiers in Health Informatics**, v. 9, n. 1, p. 29, 2020.

NAKAMURA, V. R. **Desenvolvimento de Aplicativo Móvel para o Cálculo do Índice de Infecção por Leishmaniose Usando Visão Computacional**. Dissertação (Trabalho de Conclusão de Curso) — Pontifícia Universidade Católica de Goiás, Goiânia, 2022. Graduação em Engenharia de Computação, Escola Politécnica.

Oracle. **Custom Development Method Fast Track (CDM Fast Track) Method Handbook**. Release 1.2.0. Parkway, California, USA: Oracle Corporation, 2000.

PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. 7. ed. [S.l.]: McGraw-Hill, 2011.

REPERANT, L. A.; CORNAGLIA, G.; OSTERHAUS, A. D. The importance of understanding the human–animal interface. In: **One Health: The Human-Animal-Environment Interfaces in Emerging Infectious Diseases**. London: Springer, 2013. p. 1–45.

RUBIANO, L. et al. Adaptation and performance of a mobile application for early detection of cutaneous leishmaniasis. **PLoS neglected tropical diseases**, Public Library of Science San Francisco, CA USA, v. 15, n. 2, p. e0008989, 2021.

SAÚDE, S. M. de. **Nota informativa: leishmaniose visceral**. Uruguaiana, 2020. Disponível em <https://www.uruguaiana.rs.gov.br/imgeditor/file/saude/vas/informativo/2020_Nota_Leishmaniose_Visceral_Dez_2020.pdf>.

SCHIMMING, B. C.; SILVA, J. R. C. Pinto e. Leishmaniose visceral canina - revisão de literatura. **Revista Científica Eletrônica de Medicina Veterinária**, Periódicos Semestral, X, n. 19, Jul 2012. ISSN 1679-7353.

SILVA, A. L. A. et al. **Aplicativo de Registro de Casos Suspeitos de Leishmaniose Canina e Humana - LEISH APP**. 2021. Registro no INPI. Nº do Pedido: BR 51 2021 001493 6.

SOLANO-GALLEGO, L. et al. Directions for the diagnosis, clinical staging, treatment and prevention of canine leishmaniosis. **Veterinary parasitology**, Elsevier, v. 165, p. 1–18, 2009.

SOMMERVILLE, I. **Software Engineering**. 9. ed. Harlow, England: Addison-Wesley, 2011.

SOMMERVILLE, I. **Engenharia de software**. 10. ed. [S.l.]: Pearson, 2019.

SOUSA, A. d. O.; VALENTIM, N. M. C. Designing usability and ux with uxug-ap: An observational study and an interview with experts. In: **XVII Brazilian Symposium on Information Systems (Uberlândia) (SBSI 2021)**. New York, NY, USA: Association for Computing Machinery, 2021. p. 9. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/3466933.3466959?casa_token=X457gAaxFLMAAAAAA:K_UCPcJvfh6oOos-Jrd4DLnWdJ54GKR0_-J7hic3YRYWhTicj9uPC0VFt9sFd14Ochu9Vcwb_3La>.

SOUSA, C. A. D. **Leishmania App - Cálculo do índice de Infecção de Macrófagos por Leishmanias**. 2020. Registro no INPI. N^o do Pedido: BR 51 2020 001565 4.

TAVARES, F. L. A.; MATTOS, J. C. M. **LVC**. 2002. Registro no INPI. N^o do Pedido: 04465-1.

VASCONCELLOS, J. et al. Information technology by mobile communication for the notification of canine visceral leishmaniasis. **Pesquisa Veterinária Brasileira**, Universidade Federal de Santa Maria, v. 41, n. e06771, 2021.