

LocalizaPet: Desenvolvimento de um Sistema Web para Localização de Animais Perdidos, Doações para ONGs e Adoção de Animais

Rafael Padilha de Meneses¹, Gustavo Stangherlin Cantarelli¹, Guilherme Chagas Kurtz¹

¹Curso de Ciências da Computação – Universidade Franciscana
CEP 97010-032 – Santa Maria – RS – Brasil

rafaelmeneses21@gmail.com, gus.cant@gmail.com, guitarro17@gmail.com

Abstract. *This article presents a web application aimed at helping communication between institutions and users related to pets. The system allows the common user to search for animals registered for adoption, disclosure of lost pets and can help institutions in need. Institutions will be able to advertise animals for adoption, events and request resources and voluntary help. The methodology for the implementation used was the Feature-Driven Development (FDD). The application was developed for the web platform using Spring Boot for the back-end, React for the front-end and MongoDB for the database. With this, an application was developed with proposed functions, as a means used for users concerned with animals to communicate, as well as assisting institutions.*

Resumo. *Este artigo apresenta uma aplicação web direcionada para auxiliar a comunicação entre instituições e usuários relacionados aos animais de estimação. O sistema permite ao usuário comum buscar por animais cadastrados para adoção, divulgação de pets perdidos e poderá ajudar instituições necessitadas. As instituições poderão divulgar animais para adoção, eventos e solicitar recursos e ajuda voluntária. A metodologia para a implementação utilizada foi a Feature-Driven Development (FDD). A aplicação foi desenvolvida para a plataforma web utilizando Spring Boot para o back-end, React para o front-end e para o banco de dados foi utilizado MongoDB. Com isso, a aplicação foi desenvolvida com as funções propostas, sendo utilizada como meio para os usuários preocupados com os animais se comunicarem, assim auxiliando também as instituições.*

1. Introdução

As instituições que prestam ajuda aos animais abandonados passam muito trabalho, visto que, geralmente, precisam de suporte de voluntários e doações, não conseguindo dar conta da alta quantidade de gatos e cachorros que vivem nas ruas. Não há veículos suficientes para expor esse problema e os que existem não conseguem proporcionar a ajuda necessária. Por esse motivo e outros, desenvolver uma aplicação web que forneça essas ferramentas seria um bom avanço para as instituições necessitadas de ajuda e doações, auxiliando também usuários donos de pets.

Ao analisar este problema, foram identificadas algumas aplicações móveis, sendo a maioria delas focadas em adoções de animais, geralmente com avaliações baixas e

funcionalidades precárias. A mesma pesquisa foi realizada para a plataforma web, que também resultou em poucas aplicações e com as mesmas complicações.

Com base na carência de aplicações web e móveis para esses problemas, visando buscar uma melhor experiência para os usuários que querem ajudar ou adotar animais, este projeto apresenta o desenvolvimento de uma aplicação que resolva isso, facilitando a comunicação entre as instituições e as pessoas. A aplicação oferece aos usuários a divulgação de pets para adoção e divulgação de animais desaparecidos. As instituições podem se cadastrar para obter recursos ou voluntariado e cadastrar animais para adoção.

1.1. Objetivo geral

O trabalho teve como objetivo desenvolver uma aplicação web que possa ser acessada também por dispositivos móveis, voltada para ajudar os animais de rua e instituições que fazem esse tipo de trabalho. A aplicação serve como veículo de divulgação das instituições para juntar recursos e, com estes, resolver questões como: consultas, remédios, cirurgias e alimentos.

Outro problema resolvido é a de adoções de animais, que precisam de um lar e, mesmo existindo pessoas que gostariam de adotar, não tem êxito, já que não conhecem um serviço ou instituição para fazer essa adoção. Como a ideia do projeto é ajudar animais, a aplicação tem uma sessão para usuários divulgarem pets perdidos, que muitas vezes utilizam redes sociais e talvez não alcancem as pessoas certas.

1.2. Objetivos específicos

Os objetivos específicos deste trabalho são:

- Pesquisar e estudar as instituições que ajudam animais e realizam adoção na cidade de Santa Maria/RS;
- Utilizar metodologia FDD (*Feature Driven Development*) no projeto para encontrar melhores resultados;
- Estudar e utilizar a linguagem Spring Boot e ReactJS;
- Estudar e implementar o banco de dados utilizando MongoDB para aplicações web;
- Criar uma interface amigável para que a aplicação alcance público de todas as idades, utilizando mapa e geolocalização.

2. Referencial Teórico

Esta seção apresenta a base do trabalho com finalidade de fundamentar o objetivo do mesmo a partir dos estudos relacionados.

2.1. Animais domésticos

A interação dos animais com o ser humano aparece desde os tempos primitivos, os animais auxiliavam na caça, proteção do território e no transporte de cargas. Essa interação pode ter surgido a partir da relação com lobos, que desenvolveram características dóceis. O homem depende da interação com outras espécies para sua sobrevivência, antigamente como predador, passando depois para domesticação [Caetano 2010].

Dotados de muita energia, interagem com o ser humano e possuem amor incondicional com seus tutores. Crianças se tornam mais afetivas, solidárias, sensíveis e com maior senso de responsabilidade ao conviver com animais domésticos, já algumas pessoas mais idosas tratam os animais de estimação como membros da família [Nebbe 2000].

2.1.1. Relação com o animal de estimação

O animal de estimação precisa de um tutor, e este, por sua vez, tem a responsabilidade de suprir as necessidades físicas e psicológicas do animal, bem como evitar acidentes e transmissões de doenças que causem quaisquer danos à comunidade ou ambiente inserido [Ishikura *et al.* 2017].

A interação do homem com o animal de estimação, por mais afetuosa que seja, é caracterizada por uma relação de autoritarismo, visto que o homem é quem decide pelo animal quando é a hora de brincar, sair, comer e até mesmo se vai impedir a reprodução dele por meio de castração. Quando o animal ajuda em terapias, é importante verificar se o trabalho realizado não está se tornando uma atividade de estresse, para evitar futuros problemas [Giumelli *et al.* 2016].

Existem benefícios terapêuticos na relação do homem com o animal, e são aplicados em diferentes áreas da saúde. O simples fato de brincar com o seu animal de estimação mostra um aumento nos níveis de oxitocina e dopamina (hormônios que dão sensação de prazer e bem estar). Os donos de cães tendem a ter a pressão sanguínea mais baixa e menos chances de desenvolverem doenças no coração. Eles auxiliam também como coterapeutas, ajudando no tratamento de doenças psiquiátricas, no suporte de pessoas hospitalizadas, ajudando idosos e crianças [Revista Arco 2020].

2.1.2. Abandono de animais

Segundo a Organização Pan-Americana de Saúde, as zoonoses são doenças de animais vertebrados que são naturalmente transmitidas aos seres humanos. A urbanização, invasão humana do habitat da vida selvagem, mudança climática, viagens internacionais e aumento de intimidade dos animais com os seres humanos são alguns fatores que ajudam no surgimento dessas doenças zoonóticas [OPAS 2021].

Ao início da pandemia do COVID-19, as instituições que cuidam dos animais se surpreenderam com o aumento de adoções, pelo motivo de encontrar uma companhia no período do isolamento. Os indivíduos adotaram seus companheiros de forma irresponsável, esquecendo que é necessário um planejamento para prever os possíveis gastos com o animal e o espaço que ele precisa para correr e brincar e não se deve esquecer que é preciso ter um local para suas necessidades. O estresse causado pelo isolamento social da pandemia trouxe resultados negativos, no período de 15 de março a 30 de junho de 2020, foram registrados pelas entidades 482% mais denúncias sobre maus-tratos em comparação ao ano de 2019 [Veiga 2020].

A desinformação e medo das pessoas também contribuiu para o aumento de abandono dos animais de estimação, durante a pandemia. Muitas notícias falsas sobre a transmissão do vírus de animais para pessoas foram passadas por meio de redes sociais. O distanciamento social usado para evitar novos contágios ocasionou também negativamente para os animais que vivem nas ruas e são dependentes dos recursos e

cuidados humanos, ficando, então, desprovidos de água, alimento e outros recursos essenciais [WAP 2019].

2.2. Terceiro setor

O terceiro setor é aquele que não é público e nem privado, no sentido convencional desses termos, porém tem relação com ambos. Ou seja, esse setor é o conjunto de organismos e organizações privadas sem fins lucrativos, com objetivos sociais e públicos, principalmente atuando com o voluntariado, embora não sendo integrante do governo. O primeiro setor que é constituído pelo Estado, e o segundo setor é o mercado, já o terceiro é diferente do primeiro, por que suas entidades são de natureza privada e do segundo por não visar o lucro [Paes 2020].

As ONGs (Organizações não Governamentais) protetoras dos animais sofrem consequências do enorme problema que é o abandono. Essas instituições sem fins lucrativos são mantidas com ajuda de voluntários e doações, elas auxiliam animais em situação de risco, abandonados ou vítimas de maus-tratos, que precisam de ração, medicamentos e abrigo [Matos 2012].

2.3. Geolocalização

Geolocalização é a localização geográfica de um usuário ou dispositivo de computação por meio de variados mecanismos de coleta de dados. Geralmente, é usado o endereço de roteamento de rede ou dispositivos GPS internos para determinar o local. As fontes de informações de localização incluem Sistema de Posicionamento Global (GPS) e localização por sinais de rede, como endereço IP (*Internet Protocol*), RFID (*Radio Frequency Identification*), WiFi (*Wireless Fidelity*) e endereços Bluetooth MAC (*Macintosh Operating System*) e GSM (*Global System for Mobile Communication*) [Google 2021].

Geolocalização é a prática de associar um recurso digital a um local físico. As informações são fornecidas a partir das coordenadas de latitude e longitude do indivíduo, para poder ser localizado em qualquer lugar do planeta. Com a localização, os indivíduos têm novas funcionalidades para pesquisas e organizações de conteúdos, tirando maior proveito da crescente interconectividade dos sistemas de computadores [Educause 2008].

2.4. Tecnologias

Esta seção apresenta as principais tecnologias utilizadas no desenvolvimento da aplicação web do presente trabalho. As demais tecnologias serão apresentadas na Seção 4.5.

2.4.1. Spring Boot

O Spring framework foi criado por Rod Johnson e oferece diversos módulos para serem utilizados dependendo do projeto. Inicialmente desenvolvido para a criação de aplicações web escritas em Java, sendo um ecossistema para facilitar o desenvolvimento dessas aplicações. Já o Spring Boot é uma ferramenta que surgiu a partir do framework Spring, com o intuito de facilitar ainda mais. Ele já traz para a aplicação uma configuração com a maioria dos componentes necessários, permitindo uma produção mais rápida e o mínimo esforço [Andrade 2021].

2.4.2. ReactJS

O ReactJS é uma biblioteca de construção de interface, ele é responsável pela interação com o usuário, coletando os dados, enviando ao servidor e recebendo os dados para a construção da página. Ao receber os dados em JSON do servidor, o React constrói as SPAs (Single Page Applications), que servem para evitar carregamentos desnecessários em partes da página que são repetidas, mudando só os conteúdos necessários [ReactJS 2021a].

O ReactJS facilita na criação de UIs (*User Interface*), durante o desenvolvimento ele atualiza e renderiza de forma eficiente os componentes necessários ao alterar os dados. A lógica do componente é escrita em JavaScript, facilitando na comunicação dos diversos tipos de dados ao longo da aplicação [ReactJS 2021b].

2.4.3. Google Maps API

O Google Maps tem várias API's, porém a que foi utilizada no desenvolvimento do projeto é a Javascript API, que é usada para incorporar um mapa do Google em sua página web, possibilitando algumas manipulações e adicionando alguns conteúdos. A API se encontra no Google Cloud, uma plataforma que consiste em um conjunto de recursos físicos e recursos virtuais nos data centers do Google por todo o mundo. O serviço disponibiliza vários recursos gratuitos, mas com limites, ao acabar o período gratuito, todos os recursos são interrompidos e é solicitado ao usuário a realizar um upgrade para utilizar os recursos novamente [Google 2022].

2.4.4. MongoDB

O MongoDB é um banco de dados orientado a documentos livre, foi escrito na linguagem C++, sendo de código aberto e multiplataforma. Sua classificação é NoSQL, ou seja, é um banco de dados não relacional, utilizando um método de armazenamento de dados que é modelado diferente das relações tabulares usadas em bancos relacionais. Ele utiliza documentos JSON (JavaScript Object Notation) e esquemas de banco de dados, permitindo estratégias mais flexíveis [MongoDB 2021].

2.4.5. Feature-Driven Development

O Feature-driven development (FDD) é uma metodologia ágil de desenvolvimento de software que permite desenvolver sistemas de forma rápida e tem vantagens por ser fácil de introduzir novas funcionalidades. Por usar processos simples, ele tem uma rápida exposição do projeto a novos elementos, tornando o trabalho mais dinâmico por iniciar e acabar um processo [Barbosa *et al.* 2008].

Os processos do FDD são descritos a partir do planejamento e a construção, na parte de planejamento, tendo as etapas de: desenvolver um modelo abrangente, construir uma lista de funcionalidades e planejar por funcionalidades. Na parte de construção, tem-se: detalhar a funcionalidade e construir a funcionalidade. Esse método tem vantagens, visto que é organizado em pequenas funcionalidades, resultando em melhores descrições, revisões e é mais fácil de encontrar erros [Moretti Junior 2021].

3. Trabalhos Correlatos

Nesta seção, serão apresentadas algumas características ou módulos, que serviram como base para a realização e organização deste projeto.

3.1. Conta Comigo: aplicação mobile para gerenciamento de doações, eventos e voluntariado do terceiro setor

O trabalho realizado por Bataglin e Cantarelli (2019) teve como objetivo o desenvolvimento de uma aplicação móvel para plataforma Android, voltada para o voluntariado e instituições do terceiro setor. A aplicação é usada para pessoas necessitadas de algum tipo de recurso encontrar pessoas dispostas a ajudar, ou para usuários que gostariam de prestar serviço voluntário. Para o desenvolvimento do sistema, foi utilizada a plataforma Android, com a linguagem Flutter, no *back-end* foi usado Firebase para armazenar os dados.

A partir dos estudos realizados, apresentou-se o desenvolvimento da aplicação móvel com o intuito de auxiliar a doação, voluntariado e divulgação de eventos. Para a produção do sistema, foi utilizado as boas práticas da metodologia FDD (*Feature Driven Development*), apresentando os diagramas pertinentes à arquitetura da aplicação, apresentando as principais finalidades.

3.2. Aplicação Android direcionada para ONGs e projetos objetivando viabilizar a adoção de animais domésticos

A aplicação Android de Souza e Cantarelli (2018) é direcionada para organizações de proteção animal do terceiro setor, que auxilia na gestão de adoções de animais domésticos e possibilita a publicação de campanhas para arrecadar fundos. Com isso, o objetivo inicial é divulgar as instituições que recolhem animais de rua, encontrando pessoas interessadas na adoção destes e ou pessoas que gostariam de ajudar com recursos.

O desenvolvimento da aplicação foi feito com a plataforma Android e o *back-end* utilizando Firebase, junto com as ferramentas Firebase Authentication que fornece serviços para autenticar usuários. Desse modo, a aplicação conseguiu suportar as organizações e projetos de proteção animal, contribuindo com divulgações de animais recolhidos das ruas tornando o ciclo de doação mais dinâmico e auxiliou na proposta de juntar recursos para essas instituições.

3.3. AniMaps: um sistema de informação geográfica para a adoção de animais

AniMaps desenvolvido por Batista *et al.* (2018) foi pensado para ser capaz de auxiliar na adoção de animais e usou ferramentas de geoprocessamento que permite ao usuário buscar por região. O projeto conseguiu aliar a tecnologia a uma questão social, diminuindo o abandono de animais na cidade e motivando a prática da adoção.

O sistema foi desenvolvido na plataforma NetBeans, que é capaz de integrar diversos recursos, possibilitando o uso das tecnologias HTML5, CSS3, JavaScript e PHP, que foram utilizadas na produção da aplicação para a plataforma web. Na parte de armazenamento dos dados, foi usado o Sistema de Gerência de Banco de dados MySQL, a parte principal da aplicação que é a geolocalização, utilizou-se a API do Google Maps.

3.4. Considerações sobre os trabalhos correlatos

Os trabalhos correlatos buscam desenvolver uma ferramenta para auxiliar na comunicação entre os necessitados e os que querem ajudar. Entretanto, com o tempo, novos problemas aparecem, necessitando de novas soluções, resultando em uma necessidade de ferramentas mais atuais e que resolvam mais problemas.

Bataglin e Cantarelli (2019), em seu trabalho Conta Comigo, mostram os problemas do terceiro setor e pensam como solução uma aplicação móvel como meio de comunicação para a população. Para organizações de proteção animal, a aplicação móvel proposta por Souza e Cantarelli (2018) centraliza-se no âmbito da adoção, possibilitando a divulgação de eventos para arrecadar fundos. No AniMaps desenvolvido por Batista *et al.* (2018), utilizou-se geolocalização para buscar regiões que realizam a adoção de animais, possibilitando a divulgação de instituições.

Logo, o objetivo do presente trabalho é unir as funcionalidades dos trabalhos citados, assim criando um nicho de pessoas que se interessam em ajudar os animais e instituições, que, por sua vez, ajudam animais, necessitam de voluntariado e doações.

4. Metodologia

O presente trabalho utiliza os métodos ágeis que são utilizados para desenvolvimento de sistemas de maneira rápida, utilizando a implementação de novas funcionalidades sempre que necessário [Barbosa *et al.* 2008]. O método a ser utilizado é o FDD, que é composto pelo desenvolvimento do modelo abrangente, lista de funcionalidades, planejamento por funcionalidade, detalhamento por funcionalidade e construção por funcionalidade.

4.1. Desenvolvimento do modelo abrangente

Ao desenvolver o modelo abrangente, as definições do escopo do projeto são detalhadas a partir das especificações do domínio de negócio [Silva *et al.* 2009]. Utiliza-se dos requisitos e funcionalidades pedidas pelo cliente, a fim de estruturar o sistema [Barbosa *et al.* 2008]. Dessa maneira, foi proposto o Diagrama de Domínio apresentado na Figura 1. Nesse diagrama, é desenvolvida a visão geral da aplicação, demonstrando sua estrutura, entidades e dependências.

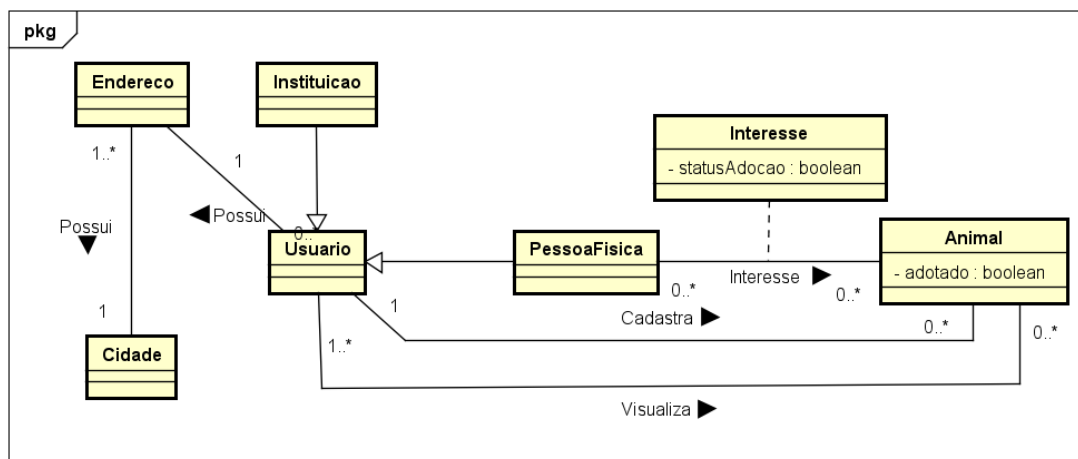


Figura 1. Diagrama de Domínio do sistema

4.2. Lista de funcionalidades

Esta etapa é realizada para o levantamento das necessidades do cliente, organizando as funcionalidades necessárias [Silva *et al.* 2009]. Essas funcionalidades são agrupadas em ordem hierárquica de acordo com a prioridade do cliente [Barbosa *et al.* 2008].

Na Tabela 1, são detalhados os requisitos funcionais do sistema, junto com sua descrição e complexidade.

Tabela 1. Requisitos Funcionais do sistema

Requisitos Funcionais		
Funcionalidade	Descrição	Complexidade
RF01: Cadastrar usuário	Deve ter a função de cadastrar o usuário.	Baixo
RF02: Gerenciar dados de usuário	Deve poder alterar os dados do usuário.	Médio
RF03: Cadastrar animal	Deve ter a função de cadastrar os animais.	Médio
RF04: Gerenciar dados do animal	Deve alterar os dados dos animais.	Médio
RF05: Cadastrar Instituição	Deve ter a função de cadastro de instituições.	Baixo
RF06: Gerenciar dados da instituição	Deve permitir a alteração dos dados das instituições.	Médio
RF07: Detalhar animal	Deve detalhar as informações do animal.	Baixo
RF08: Detalhar instituição	Deve detalhar as informações da instituição.	Baixo
RF09: Listar seus animais	Deve listar todos os animais cadastrados do usuário ou instituição.	Baixo
RF10: Integrar o mapa do Google Maps	Deve realizar a integração com o Mapa para a navegação do usuário.	Alto
RF11: Autenticação com JWT	Deve realizar autenticação com JWT para identificar usuário.	Alto
RF12: Listar pontos no mapa	Deve listar os pontos de interesse para os usuários.	Alto
RF13: Filtrar pontos no mapa	Deve ter a opção de filtrar os pontos do mapa.	Alto
RF14: Cadastrar pontos no mapa	Deve cadastrar os pontos no mapa, a partir de cadastros dos animais ou atualizando dados da instituição.	Alto

A seguir, na Tabela 2, são apresentados os detalhes dos requisitos não funcionais do sistema que servem para manter a proposta de desenvolvimento da aplicação, as principais tecnologias e caminhos que o trabalho deve tomar.

Tabela 2. Requisitos não funcionais do sistema

Requisitos Não Funcionais	
Funcionalidade	Descrição
RNF01:	A aplicação deverá ser desenvolvida em Spring Boot e React, armazenando os dados no MongoDB.
RNF02:	A interface da aplicação será desenvolvida em cima de um mapa, para melhor interação do usuário.

4.3. Planejamento por funcionalidade

Ao planejar as funcionalidades, é determinada a ordem de implementação, e o conjunto inicial de datas para a realização de cada funcionalidade e ordenado os conjuntos mais importantes por sua complexidade [Barbosa *et al.* 2008]. Na Tabela 3, foi descrita a estimativa de tempo baseada na complexidade de cada requisito, as implementações das funcionalidades totalizaram 198 horas.

Tabela 3. Estimativa de tempo para implementação das funcionalidades

Código	Funcionalidade	Tempo (horas)
RF01	Cadastrar usuário	6
RF02:	Gerenciar dados de usuário	12
RF03:	Cadastrar animal	12
RF04:	Gerenciar dados do animal	12
RF05:	Cadastrar instituição	6
RF06:	Gerenciar dados de instituição	12
RF07:	Detalhar animal	6
RF08:	Detalhar instituição	6
RF09:	Listar seus animais	6
RF10:	Integrar o mapa do Google Maps	24
RF11:	Autenticação com JWT	24
RF12:	Listar pontos no mapa	24
RF13:	Filtrar pontos no mapa	24
RF14:	Cadastrar pontos no mapa	24
Total		198

As demais tabelas implementadas, onde são ilustrados os descritivos para o desenvolvimento da aplicação, são encontradas no Apêndice A.

Logo em seguida, é construído o Diagrama de caso de uso. Na Figura 2, é desenvolvida a interação dos atores com os casos de usos, mostrando suas relações e como devem ser as funcionalidades do sistema.

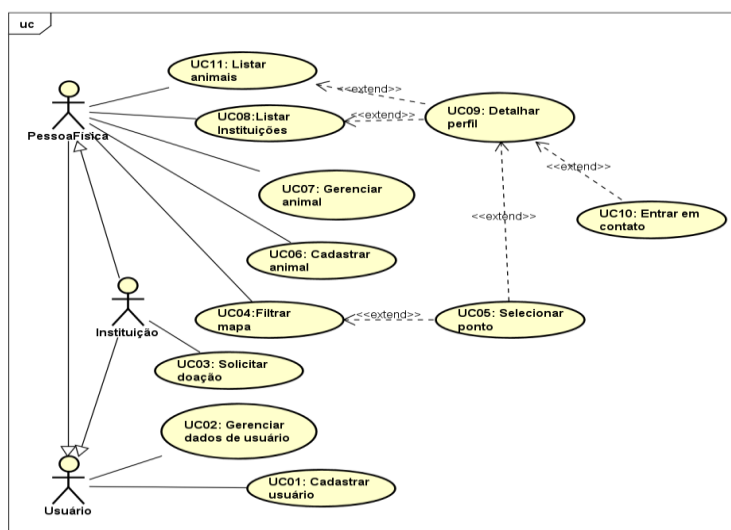


Figura 2. Diagrama de Caso de Uso

4.4. Detalhamento por funcionalidade

Esta é a parte onde é desenvolvido o esboço do modelo de interface do usuário e os diagramas de sequência e de classe são gerados [Silva *et al.* 2009]. Abaixo, a Figura 3 detalha o Diagrama de atividade principal do sistema, onde inicia o usuário para realizar login ou se cadastrar. Após essa etapa, tem-se a solicitação da localização do usuário, para ser exibido o mapa com as demais funcionalidades, que são o gerenciamento de dados, cadastro de animal, filtros do sistema e ou finalizar a aplicação.

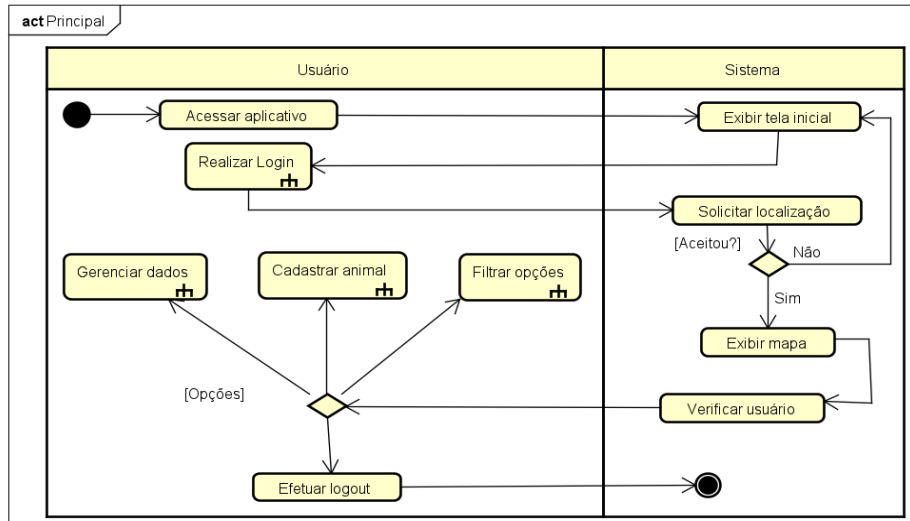


Figura 3. Diagrama de Atividade - Principal

Os demais diagramas de atividades implementados para o desenvolvimento da aplicação são encontrados no Apêndice B.

Conforme a Figura 4, o Diagrama de classes é desenvolvido para demonstrar os objetos, as funcionalidades, atributos e relacionamentos do sistema.

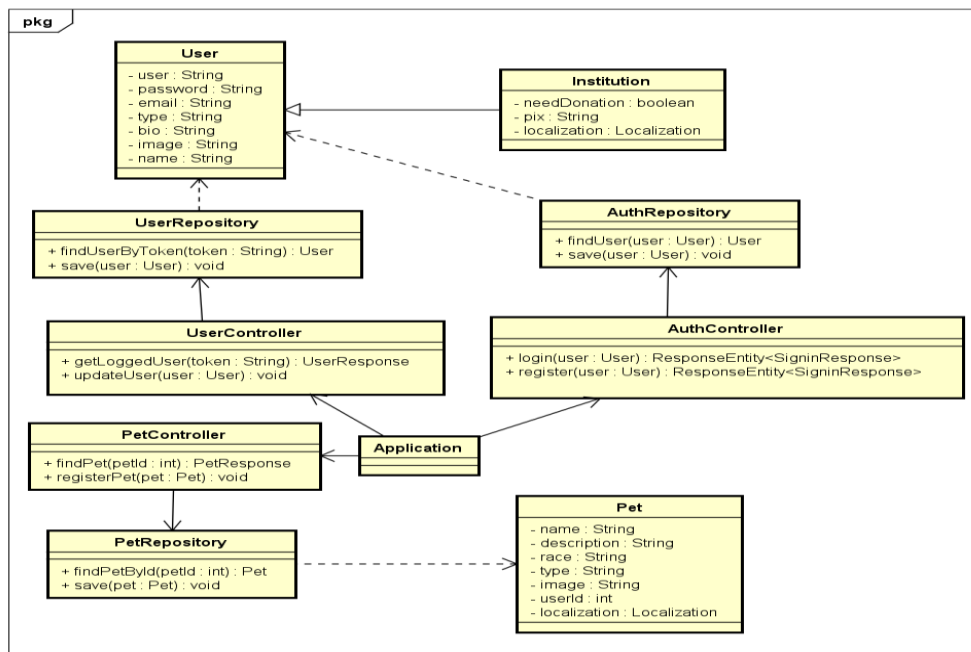


Figura 4. Diagrama de Classes

Ao representar a estrutura dos objetos relacionados do sistema, a Figura 5 mostra o Diagrama de entidade relacionamento (DER).

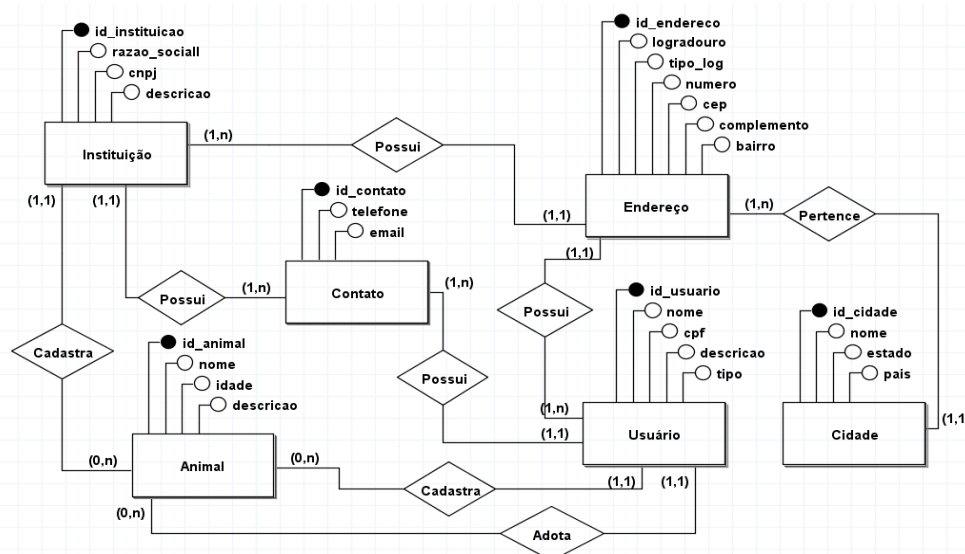


Figura 5. Diagrama Entidade Relacionamento

4.5. Construção por funcionalidade

O processo de construção das funcionalidades é o momento onde o código é gerado, produzindo, a cada interação, uma função para cada funcionalidade, agregando valor ao cliente [Silva *et al.* 2009]. Nesta seção, são detalhadas as funcionalidades no momento de seu desenvolvimento.

Iniciando pelo *back-end*, foi necessário utilizar uma autenticação para identificar o usuário que está realizando as ações dentro da aplicação. Essa autenticação foi implementada utilizando JSON Web Token (JWT), um padrão de criação de dados com assinatura e criptografia.

Abaixo, na parte esquerda da Figura 6, está a estrutura do projeto que se inicia pela *source* até as *packages* que organizam a API. Na *controller*, tem-se as classes onde são chamados os *endpoints* da aplicação; no *mapper*, são transformados os dados de requisições do usuário para os modelos; em *model*, ficam as classes que servem de espelho para as entidades do bando de dados; no *payload*, é mapeado os dados que o usuário recebe e envia; em *repository*, tem-se as classes que fazem a comunicação com o banco de dados; em *security*, ficam as configurações de segurança da API junto com as configurações do JWT; e no *package* de *service*, está toda a manipulação dos dados e suas lógicas, desde pegar os dados do usuário a moldá-los para serem salvos. A configuração do servidor, os dados para acessar o banco de dados e para configurar o JWT estão localizados no *application.yaml*.

Partindo para a aplicação front-end, na parte direita da Figura 6, é apresentada a estrutura do projeto. Em *public*, está o arquivo html com a div root utilizada pelo React, que é onde ele altera os componentes do projeto; em *assets*, tem-se as imagens e estilizações padrões; já em *constants*, ficam as variáveis imutáveis do projeto; no *contexto*, tem o arquivo com a lógica para salvar o token do usuário em uma variável utilizada em todo projeto; nos *hooks*, são estruturadas as requisições para a API; em *routes*, são especificadas as páginas em suas respectivas rotas; a pasta de ui tem os

components que o React reutiliza em cada página; e em *screens*, ficam as páginas. A base do projeto se inicia no arquivo `index.js`, que chama o arquivo `App.jsx` onde desencadeia as chamadas dos componentes.

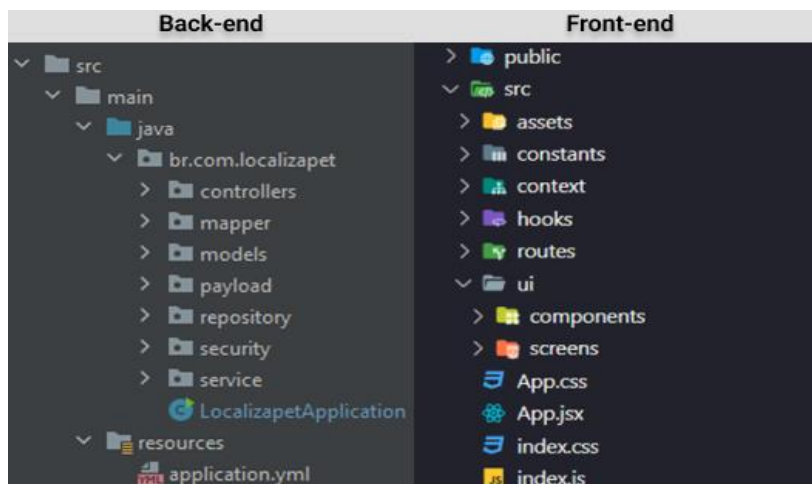


Figura 6. Estrutura do projeto

Na Figura 7, apresenta-se o código que é responsável pelo *endpoint* de login, ele recebe por parâmetro os dados que o usuário insere na página pelo `RequestBody` e que é encaminhado para o método que é responsável pela lógica.

```
@PostMapping("/signin")
public ResponseEntity<SignInResponse> login(@Valid @RequestBody LoginRequest loginRequest) {
    return authService.login(loginRequest);
}
```

Figura 7. Endpoint de login

A lógica responsável pelo login está abaixo na Figura 8. Esse método recebe, então, as informações disponibilizadas pelo o usuário, que são o usuário e sua senha. Com esses dados, é feita a verificação se o usuário é registrado. Logo em seguida, é gerado o token JWT e verificados os cargos do usuário. Busca-se o usuário no banco de dados pelo `id` e, caso ele não exista, é retornado um erro; caso exista, é retornado os dados do usuário junto ao JWT para que a aplicação *front-end* utilize.

```
public ResponseEntity<SignInResponse> login(LoginRequest loginRequest){
    Authentication authentication = authenticationManager.authenticate(
        new UsernamePasswordAuthenticationToken(loginRequest.getUsername(), loginRequest.getPassword()));

    SecurityContextHolder.getContext().setAuthentication(authentication);
    String jwt = jwtUtils.generateJwtToken(authentication);

    UserDetailsImpl userDetails = (UserDetailsImpl) authentication.getPrincipal();
    List<String> roles = userDetails.getAuthorities().stream()
        .map(GrantedAuthority::getAuthority)
        .collect(Collectors.toList());

    User user = userRepository
        .findById(userDetails.getId())
        .orElseThrow(() -> new RuntimeException(HttpStatus.BAD_REQUEST,
            "Não foi possível encontrar um usuário com esse id.));

    SignInResponse signinResponse = SignInMapper.toResponse(user);

    return ResponseEntity.ok(new SignInResponse(jwt,
        signinResponse.getId(),
        signinResponse.getUsername(),
        signinResponse.getEmail(),
        roles));
}
```

Figura 8. Método da service de login

5. Resultados e Trabalhos Futuros

Ao desenvolver a aplicação, foram encontrados alguns obstáculos, principalmente na parte de segurança. Ao iniciar o *back-end* do sistema, foi decidido criar o projeto totalmente do zero com as últimas versões das dependências do Spring Boot; sendo assim, existiam poucos exemplos de aplicações com as configurações de segurança mais novas. Os mesmos problemas se repetiram ao configurar a autenticação com o JWT.

Já no *front-end*, o maior desafio foi como manter o token do usuário vivo por meio de cookies e utilizando um *context* do React. O que levou bastante tempo também foi ao integrar a API do Google Maps com o LocalizaPet, por mais que sua documentação seja extensa e completa, haviam mais exemplos com JavaScript do que exemplos com aplicações React, então foi necessário entender a melhor forma para separá-los em componentes.

A Figura 9 ilustra a página principal da aplicação, local em que o usuário tem a possibilidade de visualizar pelo mapa onde estão os animais desaparecidos, animais para adoção e também instituições. Existe uma barra lateral no lado esquerdo que disponibiliza algumas seções, uma delas é a de filtro, possibilitando filtrar os resultados do mapa.

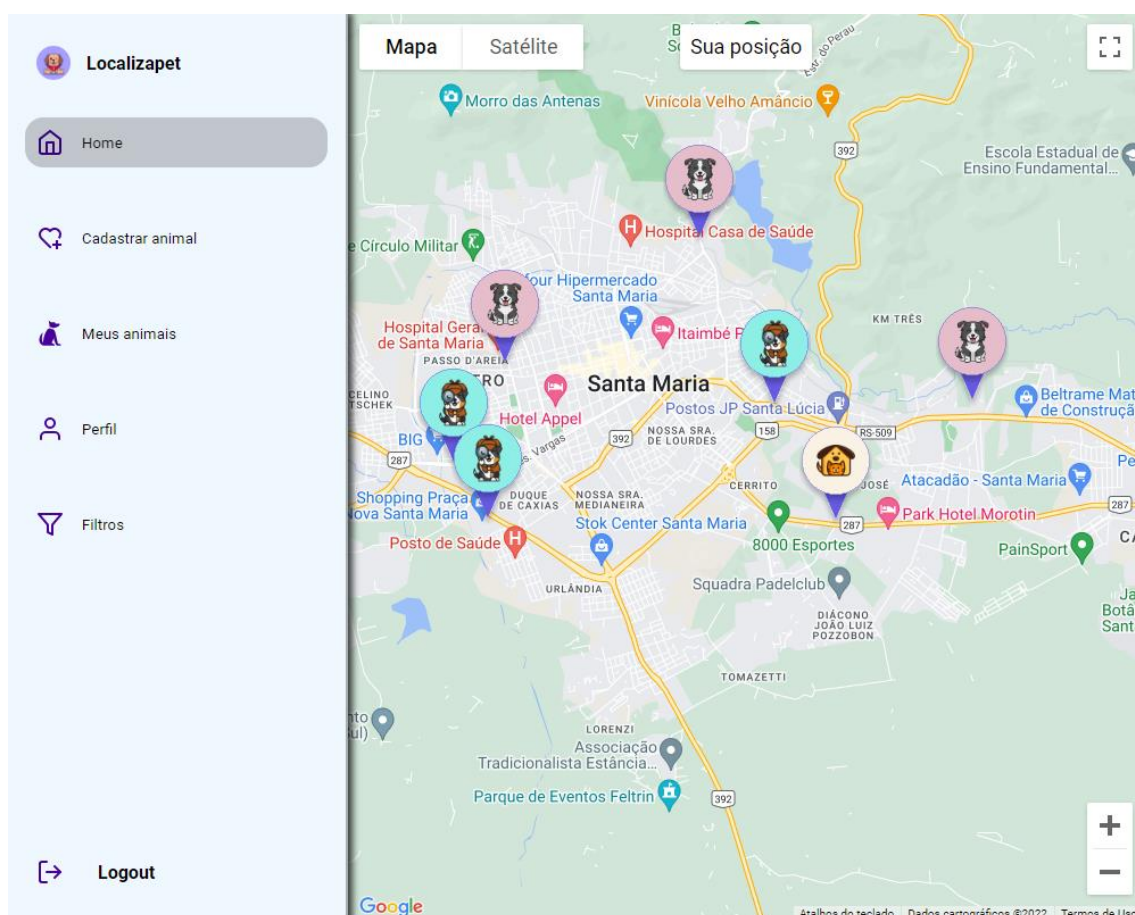


Figura 9. Página principal

A Figura 10 ilustra a página de cadastro do animal. Ao interagir com a opção, o usuário será encaminhado para a página onde irá informar os dados do seu pet e selecionar se ele está desaparecido ou se é um animal para adoção. Após realizar o cadastro, o animal irá aparecer na página principal da aplicação, no mapa com seu respectivo ícone.

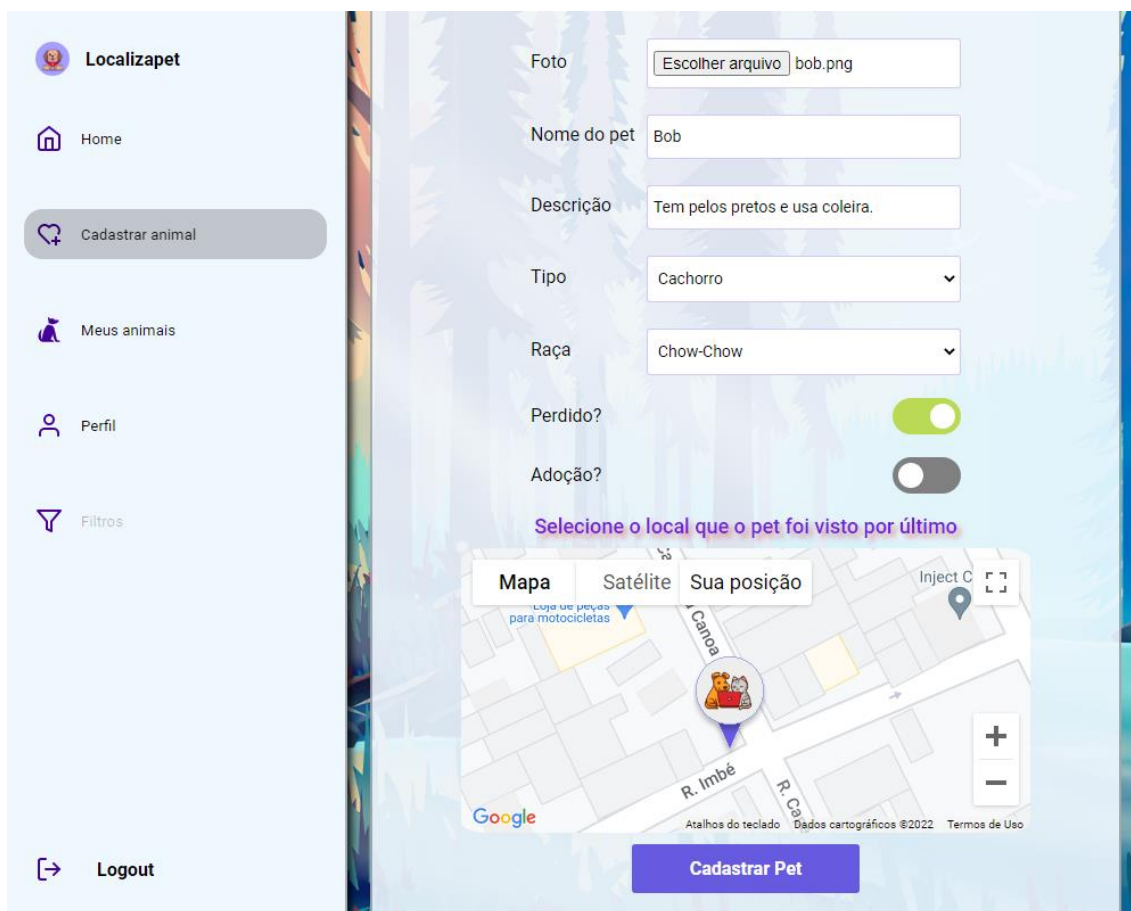


Figura 10. Página de cadastro do pet

Para completar o desenvolvimento, são descritas, no Apêndice C, as outras páginas da aplicação que contemplam as demais opções do sistema.

Para a aplicação evoluir, adaptá-la completamente para uma versão móvel seria o primeiro passo, tendo em vista alcançar um público muito maior. Outra funcionalidade ideal para a aplicação é a implementação de um chat; no momento o sistema serve para divulgar o contato do usuário, porém uma forma de comunicação dentro da própria aplicação resolve esse problema.

6. Conclusão

Este artigo propôs o desenvolvimento e implementação de uma aplicação web para a comunicação de instituições e usuários, divulgando necessidade de doações, voluntariado, adoção de animais e auxiliando os donos com animais perdidos.

Com base nas pesquisas realizadas, foi encontrada uma carência de uma aplicação para facilitar a comunicação do setor relacionado a ajudar os animais em Santa Maria/RS e em vários outros lugares da região. As aplicações disponíveis utilizam tecnologias mais antigas e sem funcionalidades com ferramentas mais novas, como WhatsApp. Os trabalhos correlatos auxiliaram no desenvolvimento da aplicação, possibilitando identificar funções importantes em uma aplicação dessa área, sendo uma boa base para a implementação.

Para a implementação da aplicação, foi utilizada a metodologia ágil FDD pelo fator de sua flexibilidade e desenvolvimento das funcionalidades. Para mostrar a construção do trabalho, foram produzidos os diagramas dessa metodologia, representando as necessidades da aplicação e seus processos. O desenvolvimento do sistema seguiu os critérios descritos na metodologia, utilizou as tecnologias propostas e serviços de terceiros para alcançar os objetivos. A aplicação abrangeu as necessidades descritas durante o artigo, as funcionalidades propostas foram implementadas e as páginas foram desenvolvidas para facilitar o uso com objetividade e simplicidade.

Por fim, ao implementar a aplicação, espera-se auxiliar os usuários que buscam por animais de estimação, mas, principalmente, espera-se ajudar os animais e as instituições que os ajudam.

Referências

- Andrade, A. P. (2021) “O que é Spring?”, <https://www.treinaweb.com.br/blog/o-que-e-o-spring>, Outubro.
- Barbosa, A. *et al.* (2008) “Metodologia ágil: Feature-Driven Development”, <https://docplayer.com.br/1330073-Metodologia-agil-feature-driven-development.html>, Abril.
- Bataglin, M. L. R. & Cantarelli, G. S. (2019) “Conta Comigo: aplicação móvel para gerenciamento de doações, eventos e voluntariado do terceiro setor”, Trabalho Final de Graduação, Universidade Franciscana, Santa Maria, RS.
- Batista, T. S., Batista, V. S. & Faria, V. I. (2018) “AniMaps: um sistema de informação geográfica para a adoção de animais”, Trabalho de Conclusão de Curso, Centro Federal de Educação Tecnológica de Minas Gerais, Divinópolis, MG, <https://www.digddv.cefetmg.br/wp-content/uploads/sites/52/2018/02/AniMaps.pdf>, Abril.
- Caetano, E. C. S. (2010) “As contribuições de TAA – Terapia Assistida por Animais à Psicologia”, Trabalho de Conclusão de Curso, Universidade do Extremo Sul Catarinense, Criciúma, SC, <https://silo.tips/download/as-contribuioes-da-taa-terapia-assistida-por-animais-a-psicologia>, Abril.
- Educause. (2008) “7 Things you should know about... Geolocation”, <https://www.educause.edu/ir/library/pdf/ELI7040.pdf>, Abril.
- Giumelli, R. D. & Santos, M. C. P. (2016). Convivência com animais de estimação: um estudo fenomenológico. In *Revista da Abordagem Gestáltica*, v. 22, n. 1, p. 49-58, http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1809-68672016000100007, Maio.
- Google. (2021) “Geolocation: displaying user or device position on maps”, <https://developers.google.com/maps/documentation/javascript/geolocation>, Abril.
- Google. (2022) “Programa gratuito do Google Cloud”, <https://cloud.google.com/free/docs/free-cloud-features>, Outubro.

- Ishikura, J. I. *et al.* (2017). Mini-hospital veterinário: guarda responsável, bem-estar animal, zoonoses e proteção à fauna exótica. In *Revista Brasileira de Extensão Universitária*, v. 8, n. 1, p. 23-30, <https://periodicos.uffrs.edu.br/index.php/RBEU/article/view/3123>, Maio.
- Matos, L. G. (2012) “Quando a ‘ajuda é animalitária’: um estudo antropológico sobre sensibilidades e moralidades envolvidas no cuidado e proteção de animais abandonados a partir de Porto Alegre-RS”, Dissertação, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, <https://lume.ufrgs.br/bitstream/handle/10183/56016/000857303.pdf>, Abril.
- MongoDB. (2021) “The database for modern applications”, <https://www.mongodb.com/>, Abril.
- Moretti Junior, J. C. (2021) “Desenvolvimento dirigido por Funcionalidades (FDD), Desenvolvimento de Software Enxuto (LSD) e Processo Unificado Ágil (AUP)”, <https://docplayer.com.br/39369379-Desenvolvimento-dirigido-por-funcionalidades-fdd-desenvolvimento-de-software-enxuto-bsd-e-processo-unificado-agil-aup.html>, Abril.
- Nebbe, L. (2000) “Nature therapy”. In *Handbook on animal assisted therapy: theoretical foundations and guidelines for practice*, Edited by A. H. Fine, Academic Press, San Diego, p. 385-414.
- Organização Pan-Americana de Saúde. (2021) “Zoonosis”, <https://www.paho.org/es/temas/zoonosis>, Abril.
- Paes, J. E. S. (2020) “Conceito de Terceiro Setor”, <https://www.escolaaberta3setor.org.br/post/conceito-de-terceiro-setor>, Abril.
- ReactJS. (2021a) “Introduzindo JSX”, <https://pt-br.reactjs.org/docs/introducing-jsx.html>, Abril.
- ReactJS. (2021b) “Uma biblioteca JavaScript para criar interfaces de usuário”, <https://pt-br.reactjs.org/>, Abril.
- Revista Arco. (2020) “Motivação errada e falta de planejamento na adoção de animais podem levar ao abandono”, <https://www.ufsm.br/midias/arco/adocao-animais/>, Abril.
- Silva, F. G., Hoentsch S. C. P. & Silva, L. (2009). Uma análise das metodologias ágeis FDD e scrum sob a perspectiva do modelo de qualidade MPS.BR. In *Scientia Plena*, v. 5, n. 2, p. 1-13, <https://www.scientiaplenu.org.br/sp/article/view/678>, Junho.
- Souza, M. A. & Cantarelli, G. S. (2018) “Aplicativo mobile direcionado para ONGs e projetos objetivando viabilizar a adoção de animais domésticos”, Trabalho Final de Graduação, Universidade Franciscana, Santa Maria, RS.
- Veiga, E. (2020) “A ‘epidemia de abandono’ dos animais de estimação na crise do coronavírus”, <https://www.bbc.com/portuguese/brasil-53594179>, Abril.
- World Animal Protection. (2019) “Como atuar com as populações de cães e gatos domiciliadas e em situação de rua durante a emergência sanitária da COVID-19”, <https://www.worldanimalprotection.org.br/como-atuar-com-populacoes-de-caes-e-gatos-domiciliadas-e-em-situacao-de-rua-durante-emergencia>, Abril.

Apêndice A. Planejamento por funcionalidade – Descritivos

Conforme a Tabela 4, o caso de cadastrar animal é explicado, junto com as pré-condições, pós-condições, fluxo principal, alternativo e de exceção. A relação com os atores, as extensões e inclusões com outros casos de usos também são descritas.

Tabela 4. Descritivo de UC06: Cadastrar animal

Identificação	UC06
Caso de Uso	Cadastrar animal
Descrição	Caso de uso responsável pelo cadastro de um animal no sistema.
Ator Principal	PessoaFísica
Ator(es) Secundário(s)	N/A
Pré-condições	
PC1	Necessidade de divulgar animal
PC2	Necessário o login do usuário.
Pós-condições	
PO1	Usuário registra o animal no sistema.
PO2	O sistema disponibiliza as informações do animal a partir do mapa.
PO3	Aplicação redireciona o usuário para a página de detalhes.
Fluxo Principal	
FP1	O usuário acessa a opção de cadastrar animal.
FP2	O usuário informa todos os dados do animal.
FP3	O usuário informa a localização do animal.
FP4	O usuário seleciona se é animal para adotar ou perdido.
FP5	O usuário finaliza o cadastro.
FP7	Fim de caso de uso.
Fluxo Alternativo	
FA1	N/A
Fluxo Exceções	
FE1	Só é permitido cadastrar animais como pontos no mapa.
FE1.1	O sistema retorna ao [FP2].
FE2	É obrigatório ter nome no cadastro do animal.
FE2.1	O sistema retorna ao [FP2].
Regras de negócio	N/A
Casos de uso Estendidos	N/A
Casos de uso Incluídos	N/A
Observações	N/A

A Tabela 5 descreve o caso de uso de detalhar perfil, mostrando também todos os detalhes sobre como é a funcionalidade no sistema ao ser utilizado pelos atores.

Tabela 5. Descritivo de UC09: Detalhar perfil

Identificação	UC09
Caso de Uso	Detalhar perfil
Descrição	Caso de uso responsável pelo detalhamento das informações do perfil.
Ator Principal	Pessoa Física
Ator(es) Secundário(s)	N/A
Pré-condições	
PC1	Necessidade de acessar as informações do perfil.
PC2	Necessário o login do usuário.
Pós-condições	
PO1	Filtragem dos pontos no mapa.
PO2	Interação com o ponto desejado.
PO3	O sistema redireciona para a página.
Fluxo Principal	
FP1	O usuário filtra pelos pontos desejados no mapa.
FP2	O usuário acessa o ponto de interesse.
FP3	O sistema redireciona para o perfil desejado com todas as informações.
FP4	O usuário entra em contato.
FP5	Fim de caso de uso.
Fluxo Alternativo	
FA1	N/A
Fluxo Exceções	
FE1	Não há instituições cadastradas na região.
FE1.1	O sistema retorna ao [FP1].
FE2	Não há animais perdidos cadastrados na região.
FE2.1	O sistema retorna ao [FP1].
FE3	Não há animais para adotar cadastrados na região.
FE3.1	O sistema retorna ao [FP1].
Regras de negócio	N/A
Casos de uso Estendidos	UC10: Entrar em contato. UC05: Selecionar ponto.
Casos de uso Incluídos	N/A
Observações	N/A

Apêndice B. Detalhamento por funcionalidade - Diagramas de atividades

A seguir, a Figura 11 demonstra as necessidades para realizar o login no sistema do LocalizaPet. No momento da realização, se o usuário não tiver cadastro, a aplicação irá fornecer os passos para a execução desse registro, possibilitando a escolha entre usuário comum ou instituição. Ao fornecer a opção, os dados são solicitados e as devidas verificações são executadas, posteriormente solicitando a confirmação do e-mail. Caso o usuário já tenha um cadastro, só serão necessárias as verificações no momento do login.

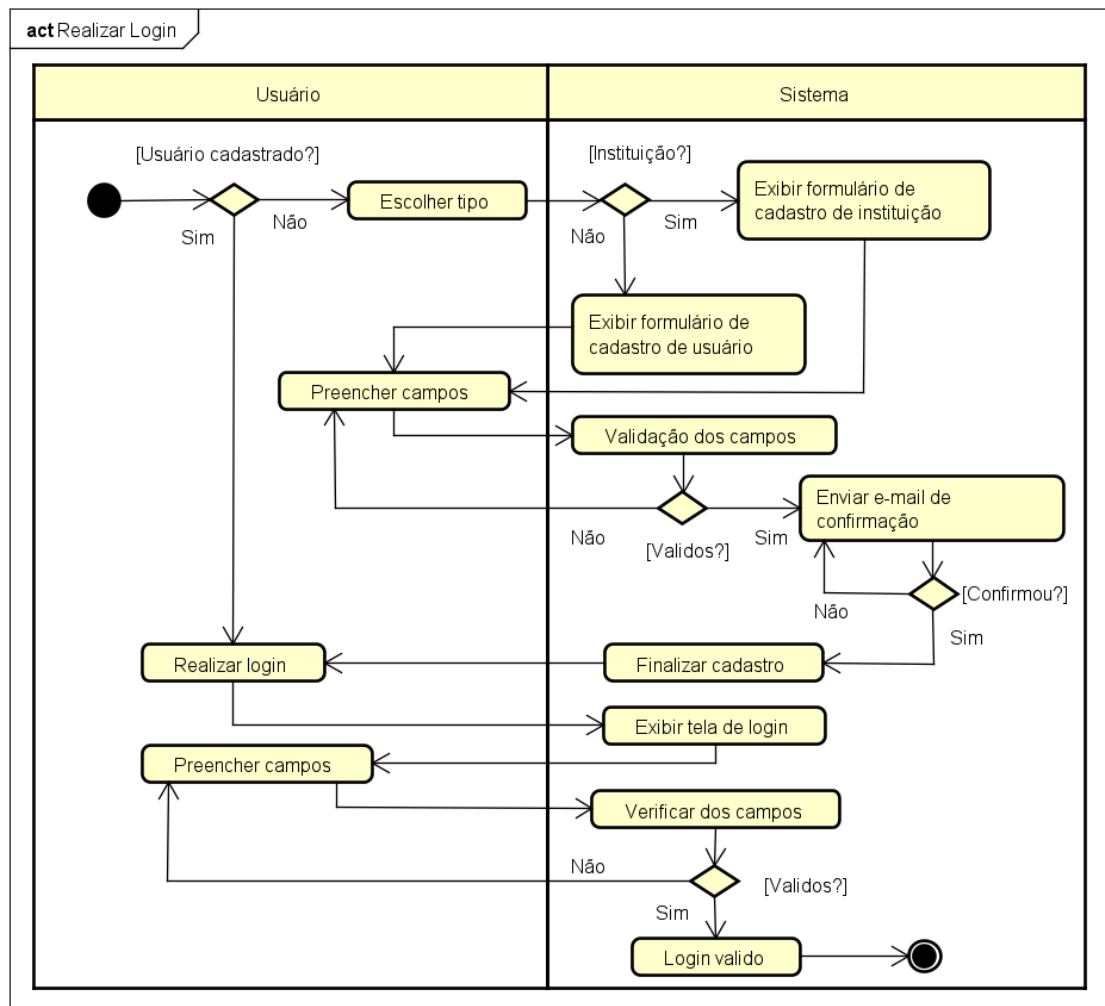


Figura 11. Diagrama de atividade - Realizar login

A Figura 12 exibe o cadastro do animal, que é realizado pelos usuários. Parecido como os outros formulários, são solicitados os dados, são verificados e ficará registrado no sistema.

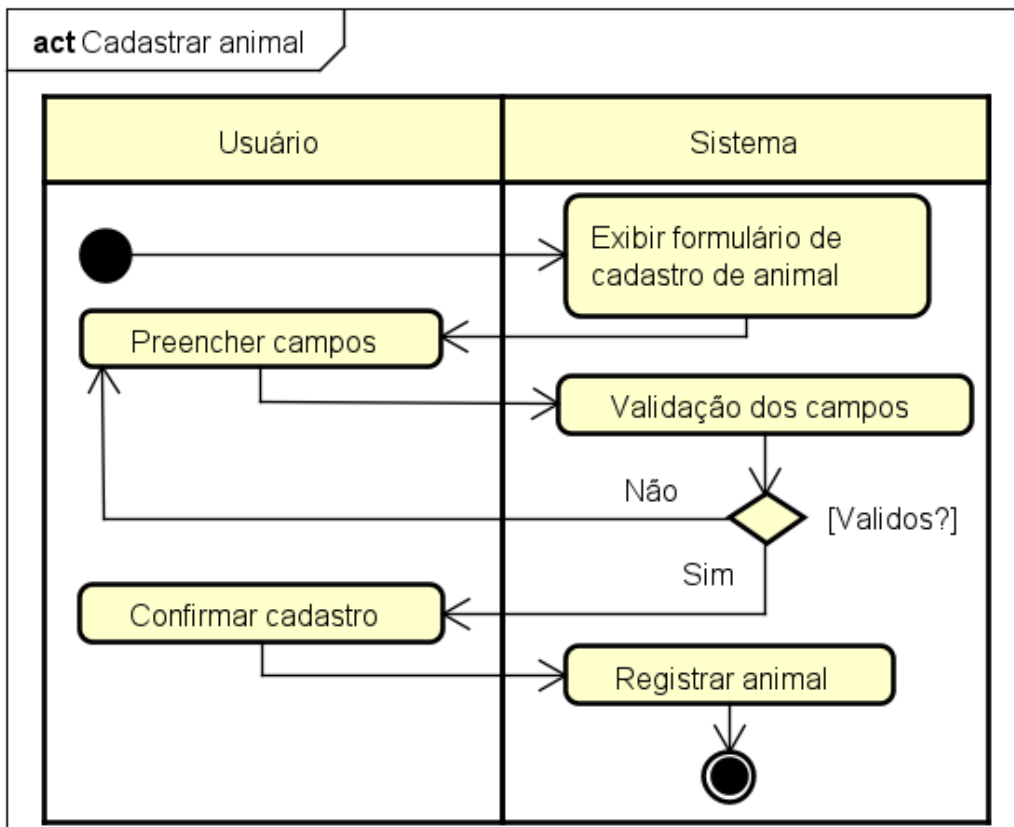


Figura 12. Diagrama de atividade - Cadastrar animal

A Figura 13 ilustra a função de filtrar opções do mapa, sendo encontrado no menu do sistema, para que o usuário organize os pontos de interesse e consiga interagir da melhor forma. Ao encontrar o que deseja, o usuário pode interagir com o ponto e prosseguir com o fluxo desejado.

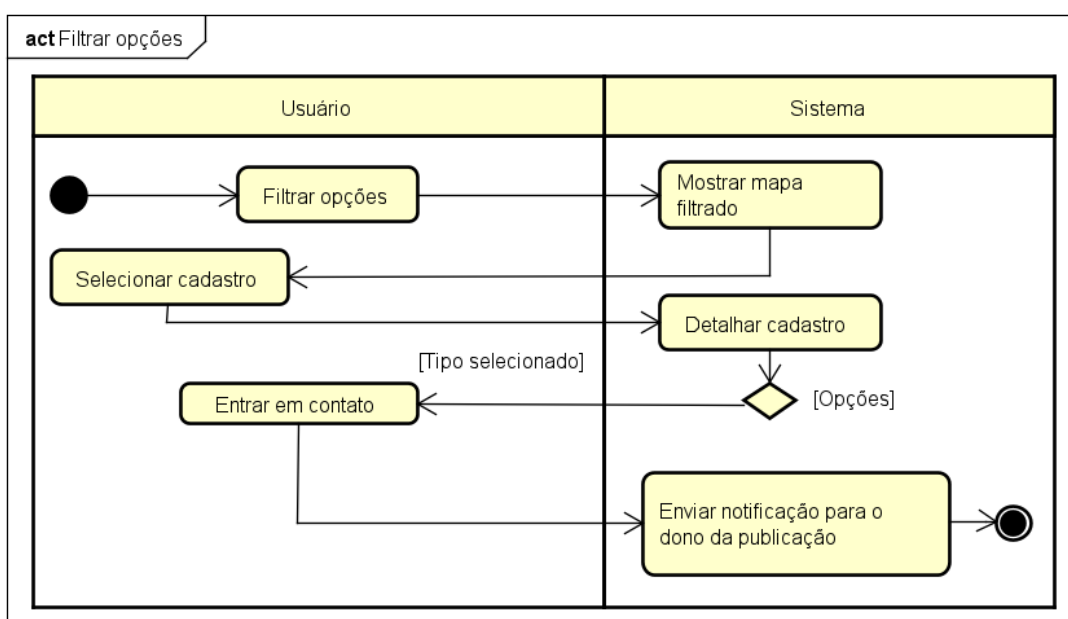


Figura 13. Diagrama de atividade - Filtrar opções

A Figura 14 ilustra a função de gerenciar dados, onde o usuário pode apagar o cadastro e alterar suas informações preenchendo os campos necessários.

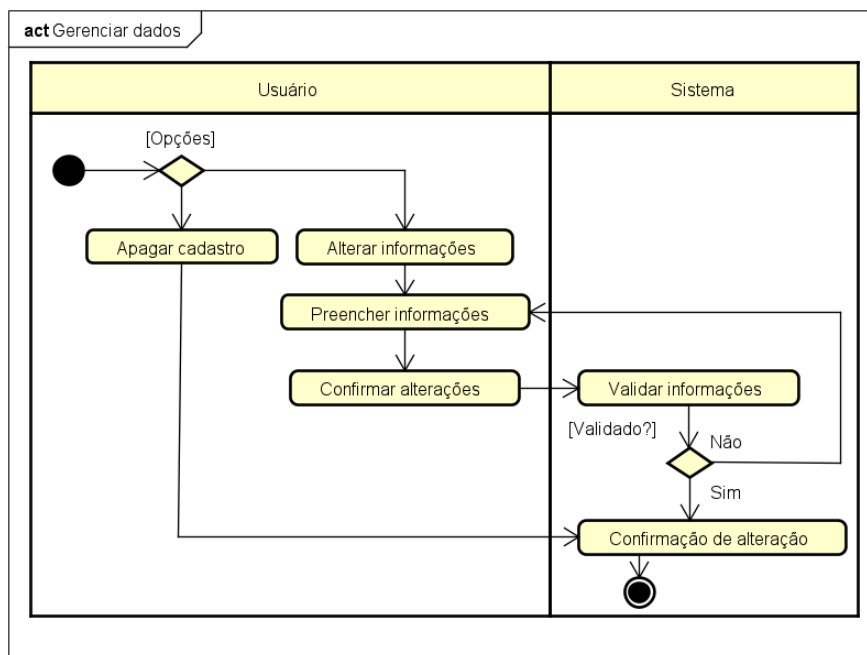


Figura 14. Diagrama de atividade – Gerenciar dados

Apêndice C. Resultados e Trabalhos Futuros

A seguir, na Figura 15, é apresentada a primeira página do sistema, que exibe o local onde o usuário irá acessar a aplicação web. Se não tiver uma conta para acessar o sistema, existe o link logo abaixo do botão de login para encaminhá-lo para a página de registro.

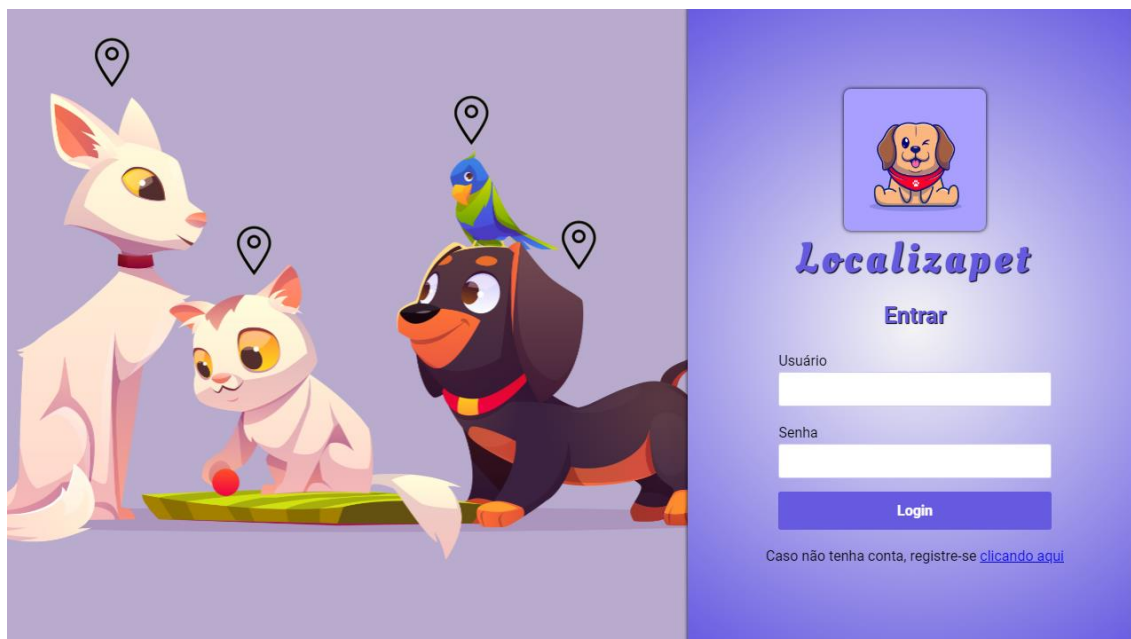


Figura 15. Página de login

A Figura 16 ilustra a página de registro do usuário com as informações necessárias da conta, e se for uma instituição, tem-se um botão para atribuir o cargo.

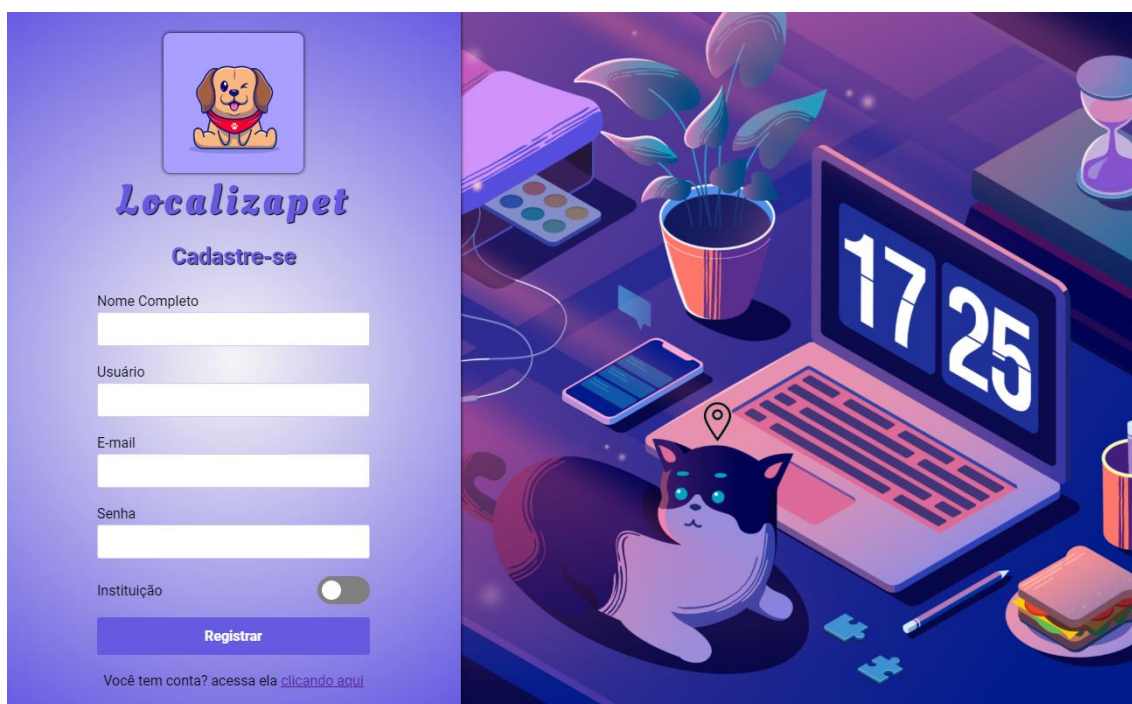


Figura 16. Página de cadastro

Na Figura 17, é apresentada a página do perfil do usuário, possibilitando alterar suas informações caso necessite. O perfil de instituição tem os campos de pix, necessidade de doação e um local no mapa, essas informações são importantes para as instituições que necessitam de ajuda.

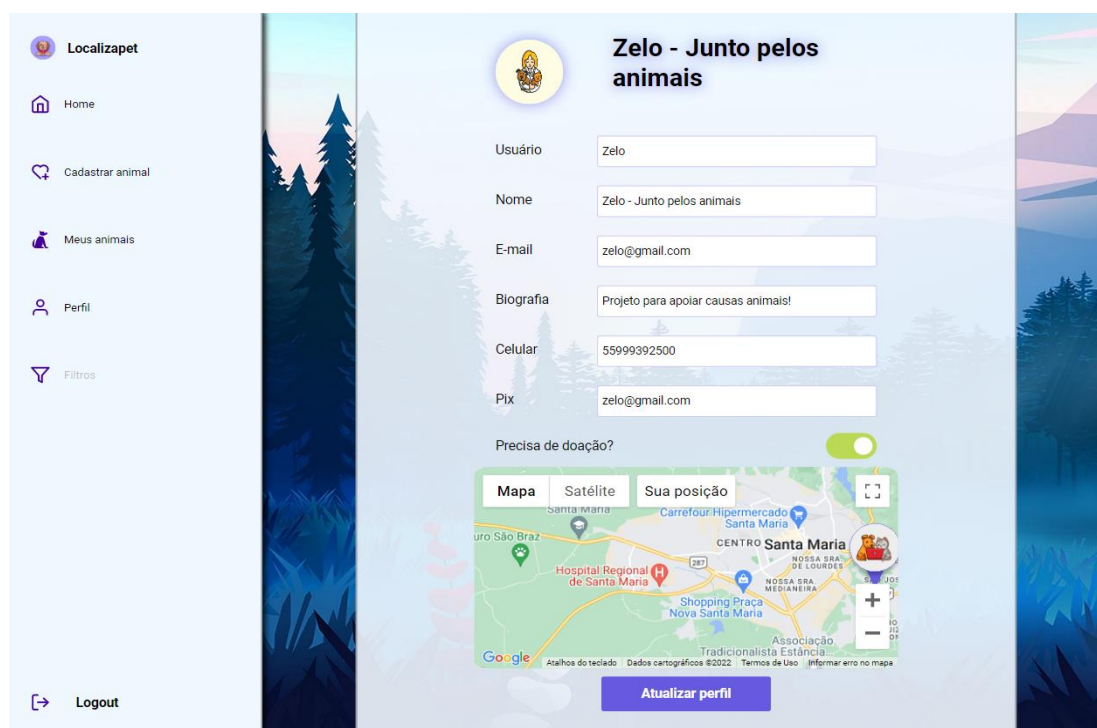


Figura 17. Página de perfil

Já na Figura 18 estão os detalhes do perfil do pet, informando o contato do responsável para que a ajuda seja oferecida. Também existe a opção de editar o pet, caso o usuário seja o dono deste.

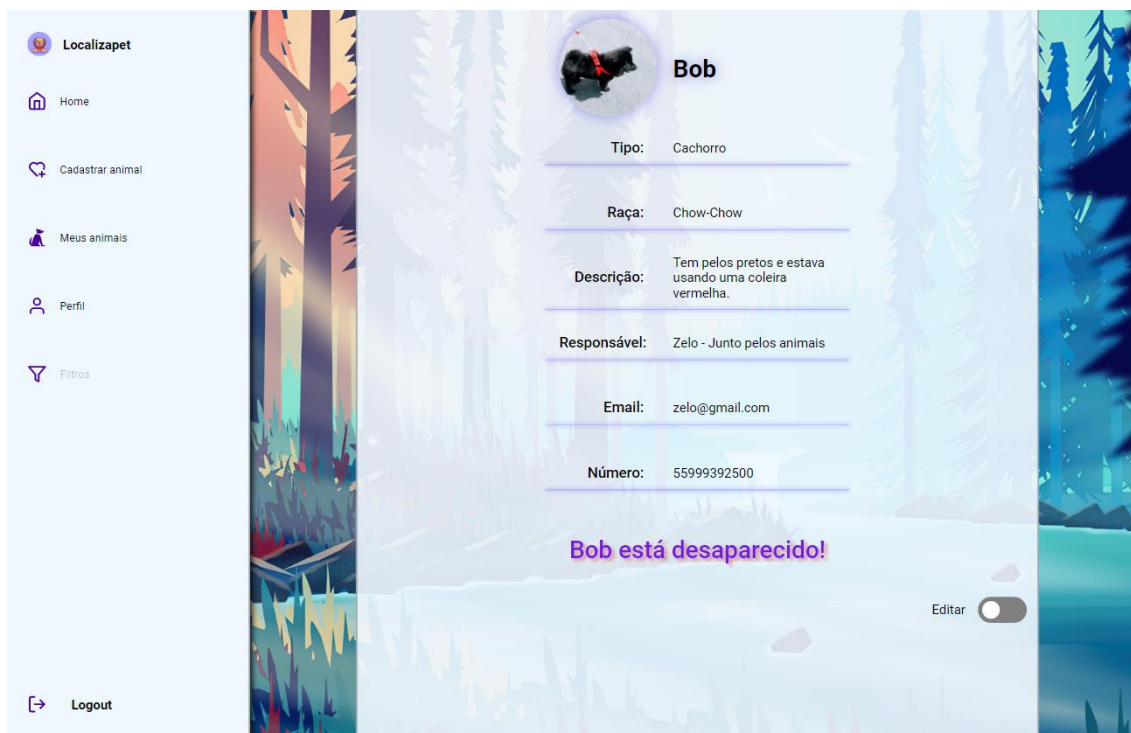


Figura 18. Página de detalhes

Em seguida, na Figura 19, apresenta-se a página de edição do pet, possibilitando a alteração de todos os dados e, se necessário, alteração do ponto no mapa.

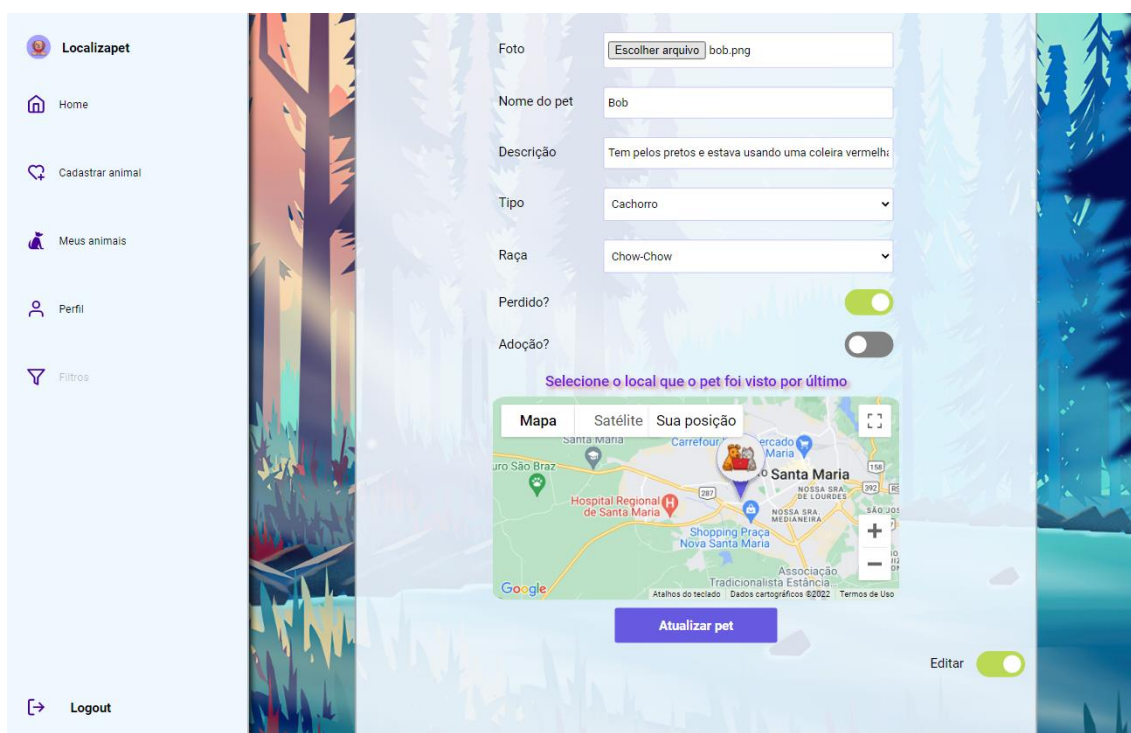


Figura 19. Página de edição do animal

A Figura 20 mostra duas opções da barra lateral, os filtros do mapa, que assim que o usuário iniciar a alteração os pontos são atualizados, e a opção de buscar os seus pets, que também atualiza no momento em que o usuário inicia a digitação da busca.

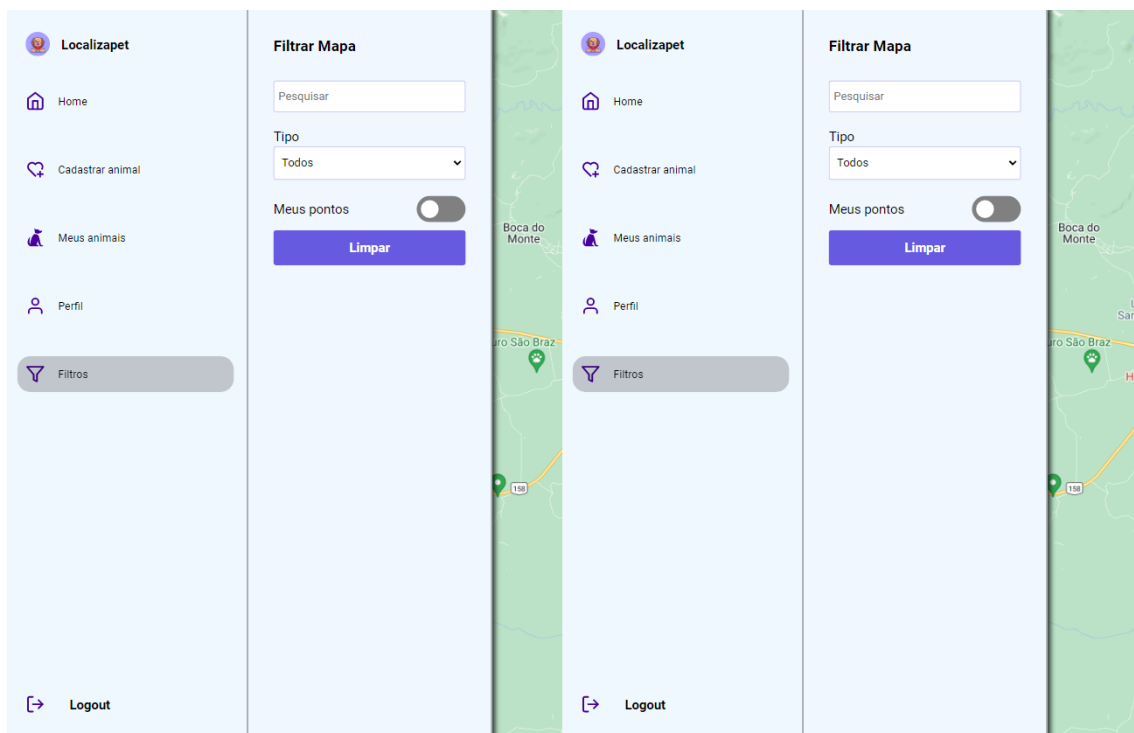


Figura 20. Opções de pesquisa de pets e filtro do mapa