Faculdades Integradas de Taquara - Faccat
Av. Oscar Martins Rangel, 4.500
Taquara, RS, CEP 95600-000

Curso de Sistemas para Internet

FINDMYPET
APP para relatar animais perdidos

Jonathan Machado[1]
Luiz Rodrigo Jardim Da Silva[2]

RESUMO

Este artigo tem por objetivo apresentar uma solução de desenvolvimento de uma aplicação móvel para auxiliar a busca de animais de estimação por seus responsáveis, que tem como intuito centralizar em um único sistema as publicações afins que são feitas em redes sociais de forma desestruturada. O sistema foi desenvolvido pensando em atrair usuários que hoje usam as redes sociais, tendo por finalidade agrupar todos os dados dos relatos em uma mesma aplicação. O sistema também oferece ao usuário a opção de cadastrar todos os seus animais com antecedência, facilitando ao usuário informar que seu animal está desaparecido, e então a partir deste momento, o sistema pede ao usuário uma descrição e permissão para acessar a localização do dispositivo, após esta etapa, a postagem é gerada automaticamente. A primeira vez que o usuário acessar o aplicativo, o sistema solicitará os dados do usuário (nome, e-mail e senha). O aplicativo irá contar com as seguintes telas: *Home, Search, Report, Settings, Profile, Message e My Pets*. O sistema é baseado em uma arquitetura Rest, ou seja, o aplicativo desenvolvido em React Native se comunicará com uma API REST desenvolvida em Node.js e MongoDB, que irá responder as requisições feitas pelo app.

Palavras-chave: Pets. Mobile. React Native.

FINDMYPET
APP TO REPORT LOST ANIMALS

ABSTRACT

This paper aims to present a solution for the development of a mobile application to help those responsible for searching for yours pets, which aims to centralize in a single

[1]Acadêmico do curso de Sistemas para Internet das Facultades Integradas de Taquara - Faccat - Taquara - RS. E-mail: jonathan.m@sou.faccat.br

[2]Professor das Faculdades Integradas de Taquara - Faccat - RS. Orientador do trabalho. E-mail: luizsilva@faccat.br

system related publications that are made on social networks in an unstructured way. The system was developed with the aim of attracting users who currently use social networks, and must group all data from the reports in the same application. The system also offers the user the option to register all their animals in advance, making it easy for the user to report that their pet is missing, and then from that moment on, the system asks the user for a description and permission to access the device location, after this step, the post is automatically generated. The first time the user accesses the application, the system will ask for the user's data (name, email and password). The application will have the following screens: Home, Search, Report, Settings, Profile, Message and My Pets. The system is based on a Rest architecture, that is, the application developed in React Native will communicate with a REST API developed in Node.js and MongoDB, which will respond to requests made by the app.

Keywords: *Pets. Mobile. React Native*

1 INTRODUÇÃO

Segundo apresentado pelo Instituto Pet Brasil (IPB), no ano de 2020, se comparado a 2019, a população *pet* mundial cresceu 1,7%. Com a pandemia e o eventual isolamento, a busca por companhia de um animal de estimação foi o que causou esse aumento significativo (IPB, 2021). Em 2018 a estimativa da população *pet* no Brasil chega a 139,3 milhões de animais, esse cálculo foi obtido através dos dados levantados pelo IBGE e atualizados pelo IPB. Segundo dados da última consulta disponibilizada pelo IBGE em 2013, a população de animais de estimação chegava a 132,4 milhões (IPB, 2019).

Se comparados a outros países em relação a animais de estimação, o Brasil é o terceiro país referente a população de animais. Outro ponto importante a ser ressaltado é que, se tratando de cães e gatos, o Brasil é o segundo colocado, ficando atrás apenas dos Estados Unidos (ABINPET, 2021). Área que o sistema procura tratar, pois tanto cães quanto gatos são os animais mais propensos a escapar de seus lares atualmente.

Um dos piores medos de quem tem um animal de estimação em casa está no fato de ele desaparecer em algum momento, qualquer descuido pode ser suficiente para que um cão ou gato escape de seu lar. Em situações onde os animais se encontram afastados de seus tutores, qualquer medida que possa prevenir a fuga ou desaparecimento desses animais e permita unir pessoas e seus animais de estimação é de grande ajuda (MOREIRA et al., 2017).

A fim de demonstrar toda a elaboração que foi utilizada durante a implementação da ferramenta, o artigo foi dividido em alguns tópicos, definidos da seguinte forma:A seção 2 tratará sobre as tecnologias utilizadas no desenvolvimento do projeto, os principais conceitos abordados e referenciais. Na seção 3, vão ser abordados os trabalhos relacionados. A seção 4 ficará responsável por apresentar a metodologia, análise e projeto. Posteriormente na seção 5 irá demonstrar o desenvolvimento e testes da aplicação mobile e API de serviços do sistema. Já na seção 6 é onde serão apresentados os resultados obtidos com o fim da implementação do sistema, demonstrando sua interface e funcionamento. Por fim, com a conclusão do artigo, a seção 7 será responsável por destacar todos os objetivos, pontos positivos e pontos negativos alcançados com o desenvolvimento do projeto, como também projetos futuros previstos para a plataforma.

1.1 Objetivos

Os objetivos desse trabalho se dividem em gerais e específicos.

1.1.1 Objetivos gerais

O objetivo deste projeto é desenvolver uma aplicação para divulgar e gerenciar os dados de acordo com as informações cadastradas pelo usuário, utilizar os recursos de geolocalização dos *smartphones* para divulgar os animais de estimação que estão nas proximidades e melhorar a experiência do usuário durante buscas feitas no aplicativo.

1.1.2 Objetivos específicos

De acordo com o objetivo geral mencionado anteriormente são apresentados os objetivos específicos, que são:

- Implementar o sistema proposto e utilizá-lo na avaliação do projeto.
- Possibilitar o reencontro de animais de estimação e seus tutores.
- Gerenciar os dados do usuário.
- Implementar recursos de geolocalização.
- Permitir contato entre os usuários do sistema.
- Implementar mecanismo de busca .

2 FUNDAMENTAÇÃO TEÓRICA

O trabalho presente neste artigo é uma aplicação mobile para relatar animais perdidos, ressaltando que as temáticas a serem abordadas neste trabalho são: Animais de estimação, uso da tecnologia na causa abordada, ferramentas e tecnologias para auxiliar na busca e relatos.

A aplicação desenvolvida é um facilitador para pessoas empenhadas na causa animal, esses usuários fazem postagens, criam grupos e páginas dedicando seu tempo e suas redes sociais para divulgar animais encontrados, sem nenhum interesse além do afeto pelos animais.

A plataforma irá unir tanto a busca, quanto a divulgação de animais, pois muitas postagens não relacionadas atrapalham e se sobrepõem, dificultando aos interessados visualizá-las. Outro recurso do sistema é informar a localização do animal ou onde ele foi encontrado.

Na plataforma será possível cadastrar seus animais de estimação, informando características como: espécie, raça, cor entre outras características e caso aconteça de o pet fugir o usuário poderá divulgar seu desaparecimento, a partir do momento que o usuário relatar que seu pet se encontra desaparecido, o sistema irá comparar este relato com os relatos de animais encontrados e notificará o dono do animal.

2.1 Animais de estimação

O isolamento social causado pela pandemia da Covid-19, ocasionou o aumento da procura de animais de estimação, para servir de companhia, visto que o isolamento trouxe sentimentos de solidão e a adoção traria um conforto terapêutico (DIVINO, 2020).

O efeito contrário também foi notado, trata-se do abandono de animais devido a falta de condições e o medo infundado de que pudessem ser transmissores do vírus (CAMPBELL, 2020).

Uma época do ano complicada para quem tem animais de estimação é o fim de ano, já que com ele vem o costume do uso de inúmeros tipos de fogos de artifício, trazendo desconforto e incômodo auditivo aos animais, ocasionando o aumento de casos de fuga, entre outros problemas (PADULLA, 2017).

2.2 Uso da tecnologia na causa abordada

A tecnologia e sua evolução possibilitaram uma nova forma de comunicação aos indivíduos da sociedade, proporcionando uma eficiente forma de se destacar no mercado almejado, atribuindo um crescimento às redes de relacionamento - *networking* (CAMPOS; WALCZAK; SOUZA, 2010).

Diversos fatores colaboram com o avanço da tecnologia e trazem mudanças, tanto administrativas, quanto pessoais, possibilitando rendimento e produtividade na realização de qualquer tarefa (CHIAVENATO, 2005).

2.3 Ferramentas e Tecnologias

Nesta subseção, são citadas as ferramentas e tecnologias utilizadas para o desenvolvimento do projeto.

2.3.1 React Native

O React Native combina o que a de melhor da plataforma nativa com o uso do React, uma biblioteca JavaScript criada com a finalidade de desenvolver interfaces para usuários. Sendo possível distribuir uma única tecnologia, mantendo duas plataformas distintas. Em 2018, o time de colaboradores do React Native tinha o segundo maior time para qualquer repositório no GitHub, além de explorar o Android e iOS, está sempre criando novos projetos interessantes para outros sistemas voltados para Windows, MacOS e Web (FACEBOOK, 2021b).

2.3.2 AWS S3

Com os avanços das tecnologias muitas empresas viram a necessidade de terceirizar alguns recursos como os de armazenamento. Com isso, os provedores de armazenamento viram a oportunidade de oferecer estes serviços através da nuvem de forma simples e limpa, abstraindo as complexidades do gerenciamento direto de hardware (PERSICO; MONTIERI; PESCAPE, 2016).

O *Amazon Simple Storage Service*(Amazon S3) serviço oferecido pela AWS para fazer o armazenamento de objetos oferece escalabilidade, disponibilidade de dados,

segurança e performance líderes do setor. Serviço esse que dá suporte a clientes de qualquer tamanho e setor para realizar o armazenamento e proteção de todos os volumes de dados de vários casos de uso (SERVICES, 2021a).

O acesso aos dados armazenados no servidor do Amazon S3 podem ser feitos por meio de solicitações autenticadas ou de modo anônimo, por meio do acesso autenticado, as chamadas REST API que são implementadas diretamente no código do sistema são feitas a partir de uma assinatura que utiliza de credenciais válidas, serviço que pode ser um pouco complicado se optar por não utilizar as recomendações feitas pela própria documentação do AWS (SERVICES, 2021b).

2.3.3 Postman

Postman é uma plataforma de API que simplifica cada etapa do ciclo de vida da API de forma agilizada e colabora para que as APIs possam ser criadas melhores e com mais rapidez. Sendo possível criar e executar quaisquer consultas REST, SOAP e GraphQL de dentro do Postman, facilitando a visualização dos retornos das informações, feitas a partir das requisições (POSTMAN, 2021).

2.3.4 MongoDB

Um banco de dados ideal para trabalhar em conjunto com Node.js são os bancos de dados NoSql, que são considerados fortemente ligados ao Node.js. Vale lembrar que existem módulos de bancos de dados SQL, porém os mais populares nesta plataforma são os módulos NoSQL. Pode se dizer que a grande vantagem de se escolher este modelo de banco de dados está no fato de que ele possui uma ampla compatibilidade e suporte que são mantidos pela própria comunidade Node.js. MongoDB é um banco de dados NoSQL que utiliza para a interface de manipulação de dados Javascript e a persistência dos dados é baseada em documentos e para isso são utilizados objetos JSON (PEREIRA, 2014).

2.3.5 API Rest

Atualmente os usuários se mantêm conectados à internet através de inúmeros dispositivos usufruindo de serviços oferecidos. Dentre eles, os mais populares são *smartphones, tablets, smart TV* e computadores. Então para que o sistema possa cobrir todos esses dispositivos requer o trabalho de construir um *web service* que no caso deste sistema utiliza a API na aplicação mobile e pretendesse desenvolver futuramente uma aplicação web que também irá acessar a API(*Application Program Interface*). Ou seja, essas APIs são os sistemas conhecidos como *back-end* que ficam responsáveis por trabalhar apenas com dados, de forma centralizada, que então assim é permitido a criação de sistemas *front-end* que ficam responsáveis pela interface que geralmente são: *mobile* apps, aplicações *desktop* ou *web* apps (PEREIRA, 2016).

2.3.6 Node.js

Node.js é uma plataforma de desenvolvimento que possui uma arquitetura totalmente não bloqueante, que apresenta uma boa performance referente ao consumo de memória e que utiliza o poder de processamento dos servidores ao máximo e de

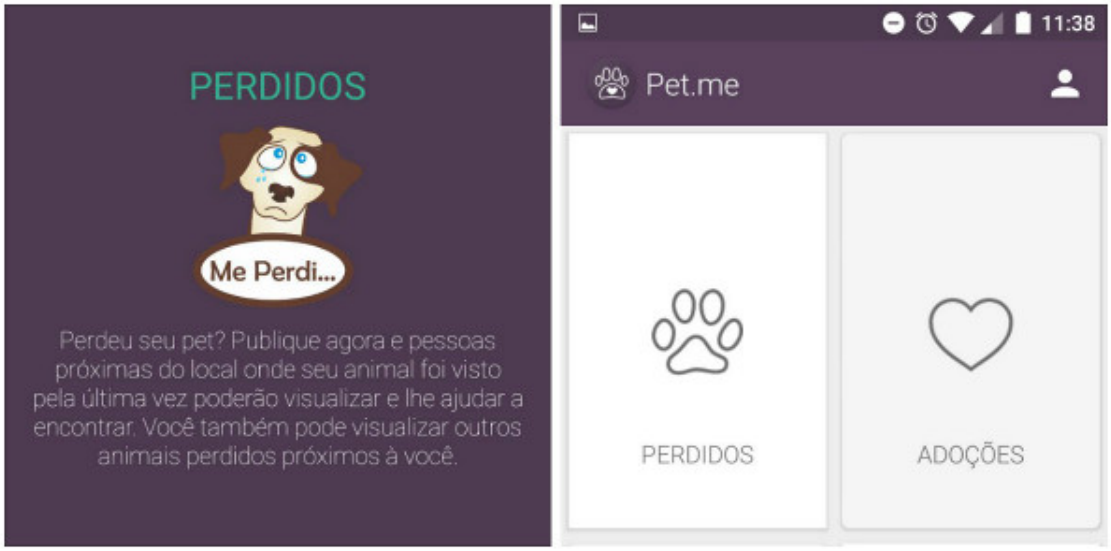
forma eficiente. Muito bem vindo quando aplicado em sistemas que geram um grande volume de requisições ao servidor. Diferentemente como é o caso da grande maioria das linguagens de desenvolvimento que produzem sistemas bloqueantes onde as requisições feitas são processadas uma a uma, enquanto as demais ficam em espera, não permitindo múltiplos processos, assim mantendo uma fila de requisições ociosas. Neste caso pode se dizer que quando se trabalha de forma assíncrona pode se conseguir inúmeros benefícios relacionados a processamento I/O, pois uma chamada I/O é considerada uma tarefa muito custosa para um computador realizar, que se torna perceptível ao usuário quando precisa esperar por uma requisição ser processada e respondida (PEREIRA, 2014).

3 TRABALHOS RELACIONADOS

Para o desenvolvimento do projeto foram realizadas algumas pesquisas por ferramentas que oferecem um papel semelhante, e dentre elas está: Pet.me lançada em 2017, disponibilizada pela empresa de mesmo nome, Cade.pet atualizado pela ultima vez em 2019 e oferecido pela FORBOMB.

Pet.me é um aplicativo que permite compartilhar uma foto e informações de seu *pet* para que pessoas próximas possam ajudá-lo a encontrá-lo, como é possível perceber na Figura 1. O *app* também conta com divulgações de animais para adoção, podendo entrar em contato com o dono para o adquirir. Outro recurso do aplicativo é a divulgação de serviços voltados para animais como clínicas veterinárias (CURIOSAMENTE, 2017). Em relação ao sistema proposto neste projeto, o Pet.me oferece divulgação a serviços voltados a animais, no entanto ele não faz divulgações de animais encontrados, nem permite o usuário ter uma visão geral de todas as publicações feitas no sistema como no “Find My Pet”.

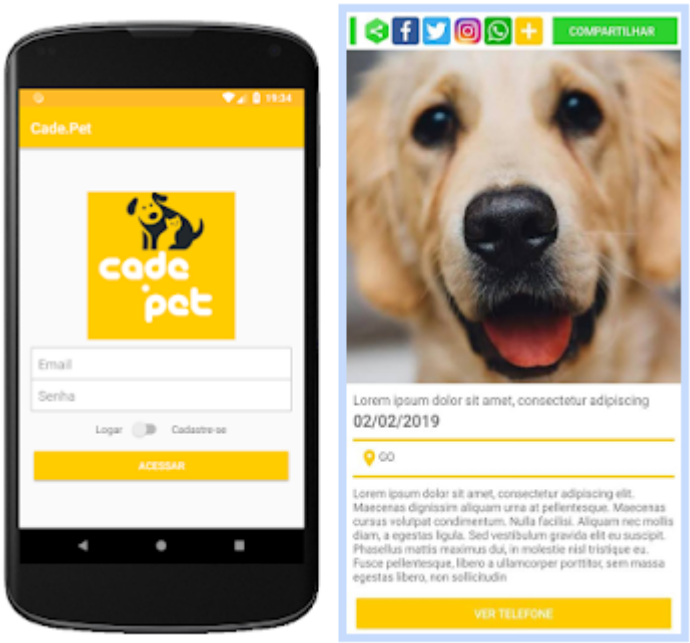
Figura 1: Aplicativo "Pet.me"



Fonte: CURIOSAMENTE (2017)

Cade.pet é um aplicativo para animais perdidos ou em situação de risco, como pode ser visto na Figura 2. O *app* permite ao usuário fazer postagens relacionadas a animais de estimação, como doações, lar temporário e denúncias de maus tratos, para que alguém possa ajudá-lo (FORBOMB, 2019). O Sistema desenvolvido permite ao usuário criar relatos com descrições livres. Porém, não possui nenhum mecanismo de filtro de busca avançado para que o usuário consiga realizar uma busca mais específica.

Figura 2: Aplicativo "Cade.pet"

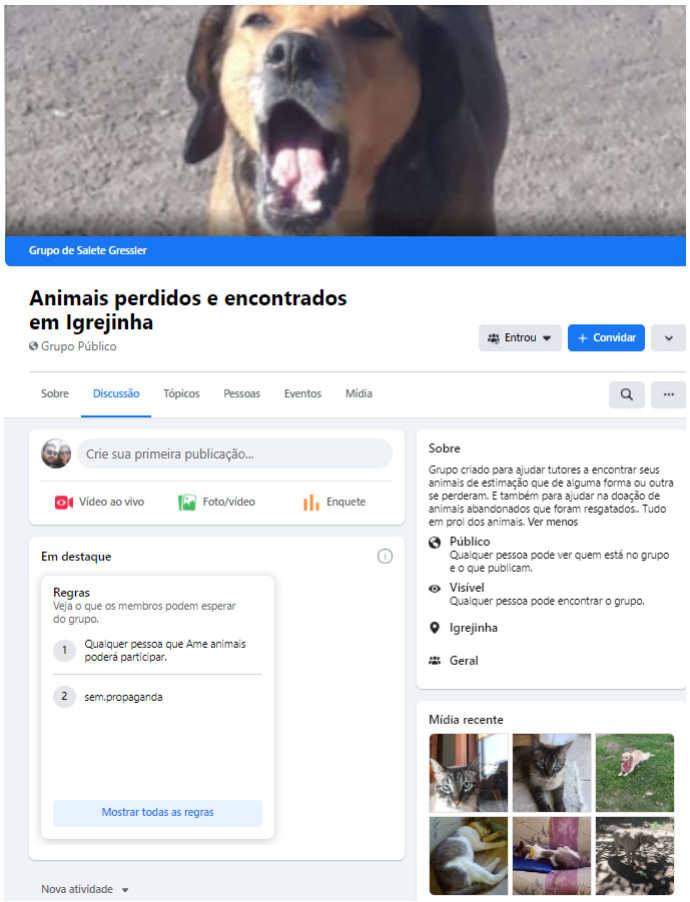


Fonte: FORBOMB (2019)

Em 2019 o Facebook completa 15 anos e com isso atinge o marco de 2,3 bilhões de usuários diários (G1, 2019), rede popularmente utilizada por ONGs para criar postagens com a finalidade de anunciar animais de estimação perdidos, ou para publicar algum animalzinho que foi encontrado e animais para adoção. O Facebook permite criar e compartilhar publicações na *timeline* de modo prático e fácil com apenas um clique, o que lhe favorece. Além do mais, a plataforma conta com diversos grupos criados por usuários da ferramenta com a finalidade de publicar sobre os *pets*, entre eles: Animais Perdidos Taquara e Região (RS) 2[3] , Apata Taquara[4] e a página Animais perdidos e encontrados em Igrejinha[5] apresentada na Figura 3.

[3]https://www.facebook.com/groups/animaisperdidostaquaraeregiao2/
[4]https://www.facebook.com/apatataquara
[5]https://www.facebook.com/groups/616494848756618/

Figura 3: Animais perdidos e encontrados em Igrejinha



Fonte: Facebook (2021a)

Sendo um dos aplicativos mais utilizados por ongs e serviços voltados para animais criando grupos e páginas para relatar animais perdidos, abandonados, animais disponíveis para adoção, entre outros. Usuários da rede fazem publicações, comentam, curtem e compartilham postagens nesses grupos. O Facebook não foi desenvolvido exclusivamente para publicações voltadas a animais de estimação e sim publicações de modo geral. Esta plataforma permite publicações de qualquer interesse social, enquanto o sistema desenvolvido neste projeto tem como foco somente publicações de animais.

Tabela 1: Tabela comparativa de trabalhos relacionados

	FindMyPet	Cade.pet	Pet.me	Redes Sociais
Animal Perdido	✓	✓	✓	✓
Animal Encontrado	✓	X	X	✓
Chat de contato	✓	X	X	✓
Última Localização	✓	X	✓	X
Filtro Avançado	✓	X	X	X
Notificar Relatos Semelhantes	✓	X	X	X

Fonte: Desenvolvido pelo autor

Conforme apresentado na Tabela 1, foram listados os diferencias do projeto em relação aos trabalhos relacionados, como pode ser visto, o principal diferencial da plataforma desenvolvida está no fato de o sistema comparar os relatos criados com os já adicionados.

4 METODOLOGIA E PROJETO

O desenvolvimento deste projeto fez a utilização da FDD, Feature Driven Development, uma metodologia ágil que tem como foco as funcionalidades do sistema e após de desenvolver um modelo e construir uma lista de funcionalidades se começa a planejar, detalhar e construir uma a uma individualmente (PRESSMAN; MAXIM, 2016).

A fim de acompanhar toda a rotina de desenvolvimento do projeto complementando o uso de FDD e buscando um maior controle do desenvolvimento tanto do que está em execução quanto das próximas tarefas, foi adotado o uso do quadro Kanban, método utilizado para gerenciar o fluxo do trabalho a fim de definir, gerenciar e melhorar o desenvolvimento da aplicação. Ajudando na visualização do trabalho e melhorando a eficiência, com o auxílio do trello, para tornar possível a divisão das tarefas e impor suas prioridades.

Realizou-se uma pesquisa para definir qual seria o público-alvo deste estudo, a partir da análise dos dados apresentados pelo Instituto Pet Brasil (IPB), onde a população de animais de estimação chega a 139,3 milhões, dos quais 54,2 milhões são cães e 23,9 milhões são gatos (IPB, 2019). Tendo isto em vista, o público-alvo deste trabalho serão pessoas que possuem animais de estimação no momento cães e gatos. Desta maneira, considerando a pesquisa realizada, o tema central do presente trabalho é a busca por animais desaparecidos. Observa-se que há necessidade de um aplicativo voltado para esta causa, trazendo uma solução para auxiliar usuários e possibilitando o reencontro com seus *pets*.

4.1 Plataforma

O sistema desenvolvido serve para unir tutores a animais resgatados por meio de uma plataforma, que tem como intuito beneficiar o reencontro dos animais e seus tutores, possibilitando o contato entre os usuários da aplicação, de um lado quem busca encontrar seu *pet* perdido e do outro o usuário que encontrou o animal desaparecido, contato esse que vem sendo feito a partir das redes sociais, no entanto, como não é um sistema construído para esta finalidade fica difícil realizar a gestão correta dos dados do sistema.

Responsável pela face do sistema, está o aplicativo que fica encarregado pela adesão de usuários que buscam relatar seus animais perdidos ou animais encontrados, apresentando uma imagens e descrições. A fim de destacar melhor os relatos, o usuário tem a possibilidade preencher informações específicas como, por exemplo, raça, cor, pelagem, tamanho, sexo, dentre outras características pré estabelecidas que o aplicativo restringe no seu desenvolvimento.

Sendo responsável por centralizar todas as informações que forem consumidas pela aplicação que é acessada através de diferentes perfis de usuário, foi desenvolvida uma API REST, um modelo de arquitetura utilizado na comunicação entre sistemas distribuídos (LIMA, 2020), que efetua a manipulação e disponibilização de dados para realizar a integração ao aplicativo, sendo possível suprir então as necessidades da plataforma.

4.2 Análise e projeto

Em função de existirem duas aplicações, foi criado um quadro Kanban, para aplicar uma boa organização e gerenciamento ao projeto da ferramenta construída, desta forma proporcionando uma melhor implementação das funcionalidades do sistema. Desta forma, foi possível separar a implementação de modo que não seja necessário implementar todas as funcionalidades ao mesmo tempo e ainda assim tornando possível verificar o estado em que estão cada uma delas.

Como se pode ver na Tabela 2, foi aplicado o processo de análise de necessidades, onde são separadas por funções e após feito a divisão das tarefas de cada funcionalidade do sistema. A partir desta análise se pode obter uma ampla visão das tarefas a serem realizadas no decorrer de todo o processo de desenvolvimento.

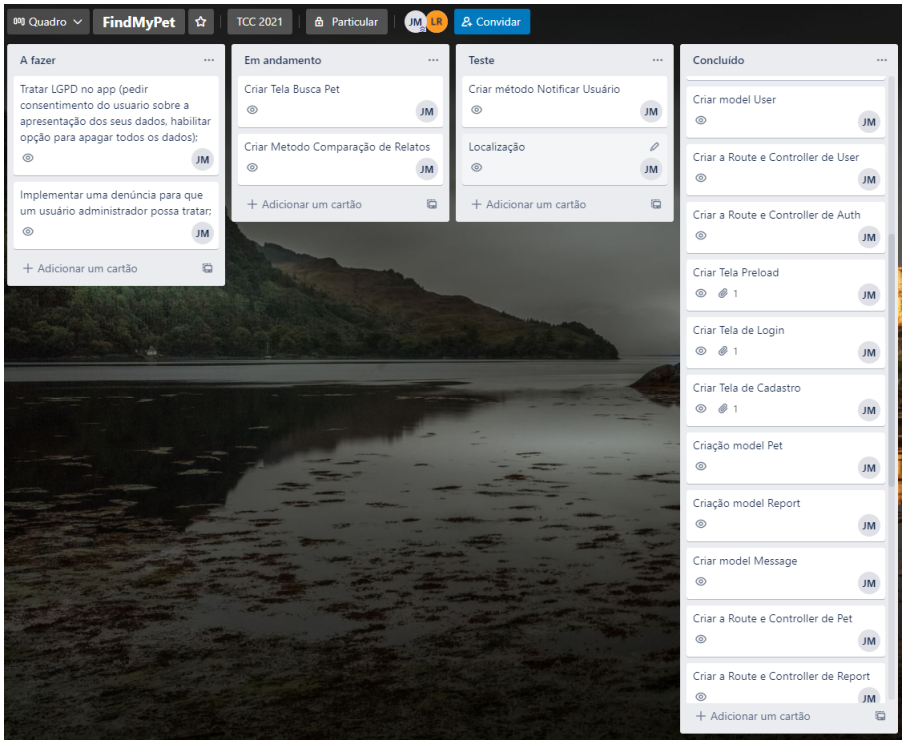
Tabela 2: Relação de tarefas da funcionalidade Apresentar relatos de animais

APLICAÇÃO	FUNCIONALIDADE	TAREFAS
API	Apresentar relatos de animais.	Receber requisição de consulta
		Recuperar dados da requisição
		Validar dados enviados na requisição
		Gerar consulta com base nos dados validados
		Realizar busca no banco de dados
		Gerar objeto JSON de retorno
		Retornar dados da requisição

Fonte: Desenvolvido pelo autor

Após a conclusão das definições, conforme a análise das funcionalidades, utilizando o Trello que é uma ferramenta de gerenciamento de projetos baseado em Kanban, as funcionalidades foram incluídas de uma a uma em cartões e alocadas no quadro para que fosse possível dar início ao desenvolvimento e planejamento, conforme foi previsto, para então dar continuidade na aplicação. Conforme foi descrito acima, para melhor monitorar o desenvolvimento do projeto foi feito o uso de um quadro Kanban, semelhante à da Figura 4, apresentada a seguir.

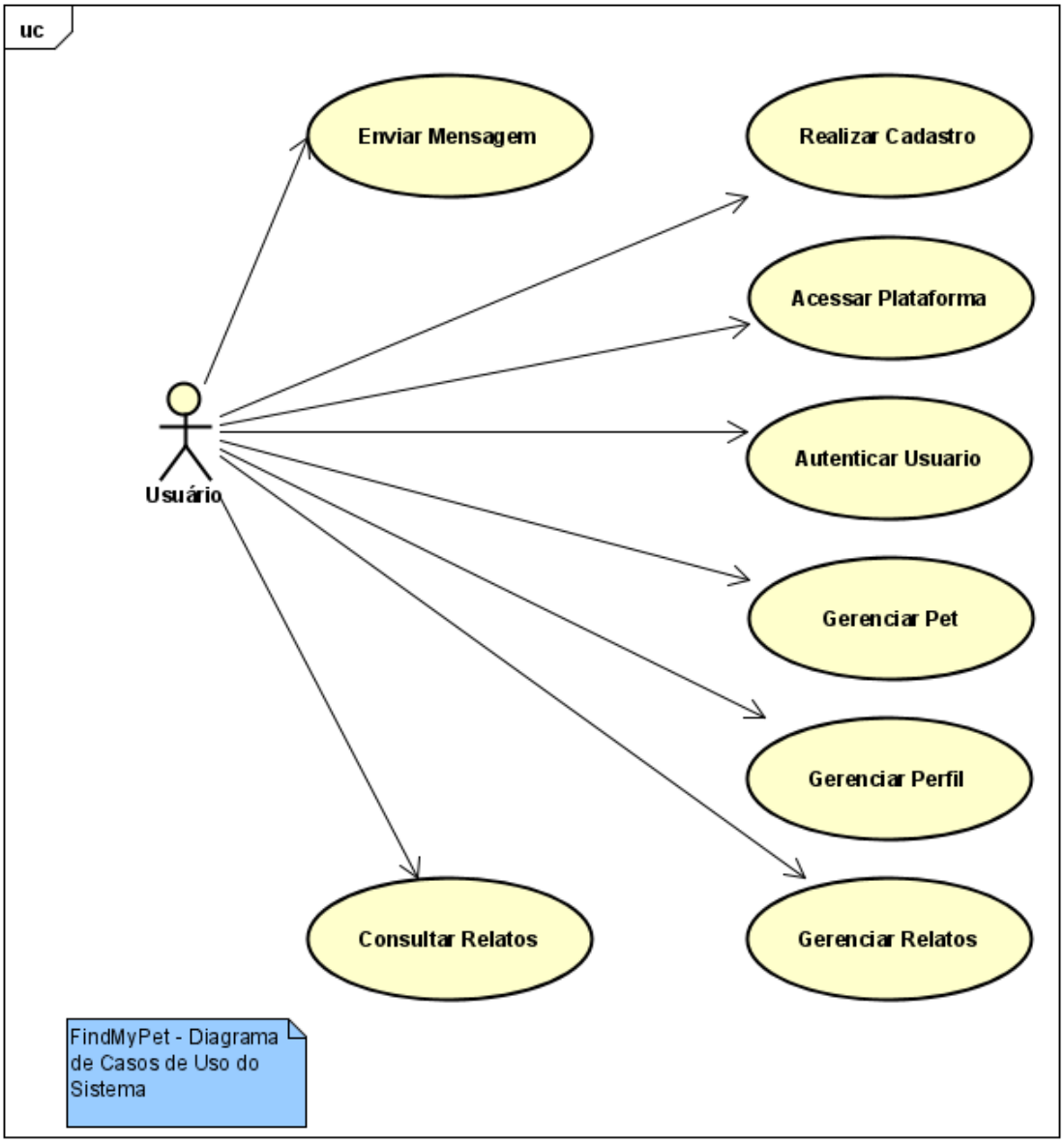
Figura 4: Kanban virtual



Fonte: Desenvolvido pelo autor

Realizada a definição das funcionalidades, foi efetuada a construção do diagrama de casos de uso, um diagrama da linguagem UML (*Unified Modeling Language* - Linguagem Unificada de Modelagem), que serve para representar a aplicação por meio de notações gráficas. Demonstrando todas ações que os perfis atuantes desempenham na ferramenta, na Figura 5 é possível visualizar as informações consumidas pelo usuário do sistema, a partir dos dados que foram geridos por outro usuário. Como pode ser observado abaixo, todos os usuários desempenham as mesmas ações, que resultam em dados que podem ser consumidos por outro atuante.

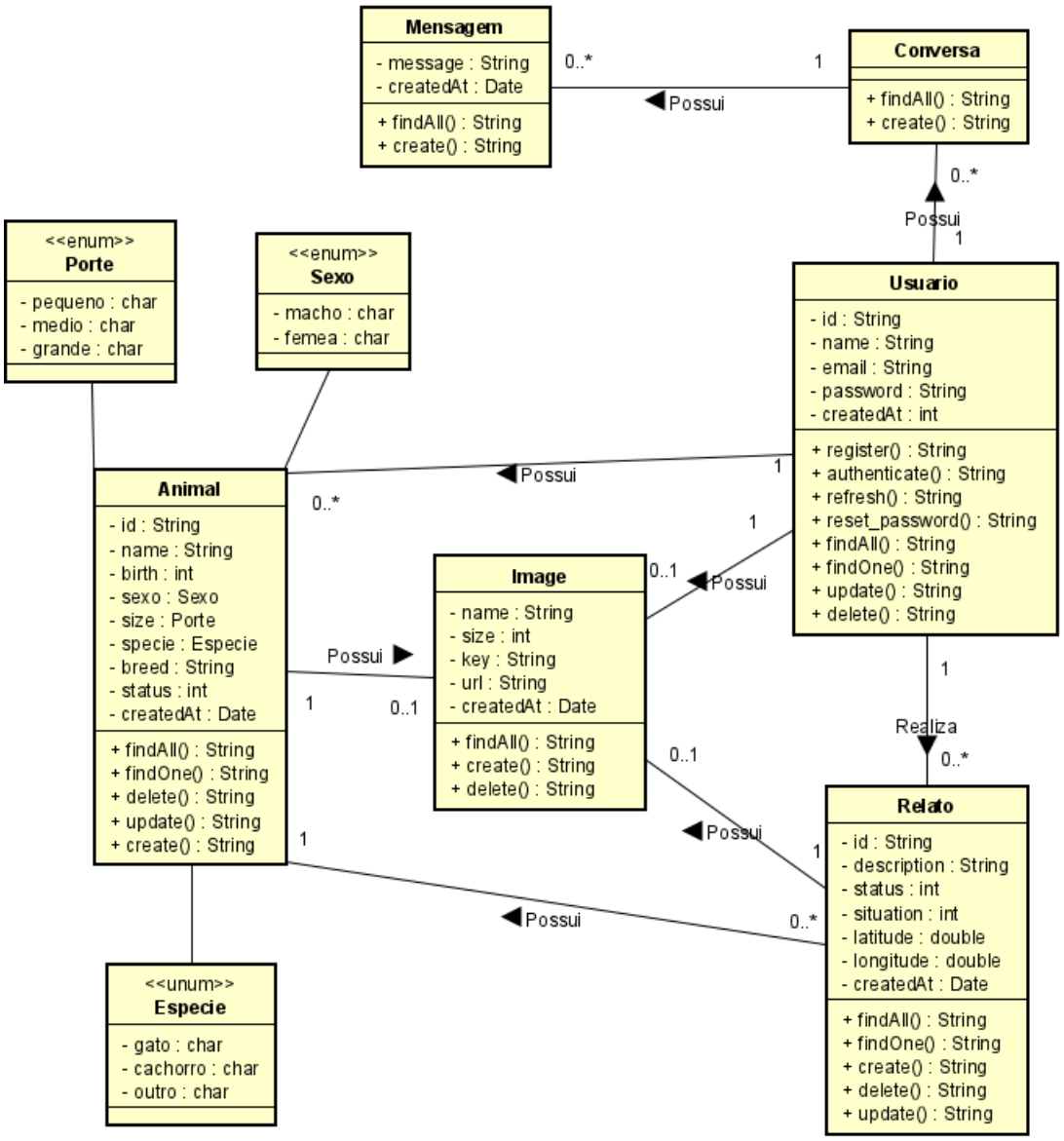
Figura 5: Diagrama de caso de uso



Fonte: Desenvolvido pelo autor

Como apresentado na Figura 6, foi realizada a construção de um diagrama de classes. Deixando de forma mais clara a visualização de quais classes e atributos são necessários para realizar implementação das funcionalidades previstas, assim contribuindo para melhorar a forma como as funcionalidades seriam implementadas.

Figura 6: Diagrama de classes



Fonte: Desenvolvido pelo autor

Conforme foi descrito, com base no que foi definido e detalhado no diagrama de classes, é possível iniciar o desenvolvimento da aplicação seguindo os dados que foram incluídos no quadro de tarefas Kanban.

5 DESENVOLVIMENTO E TESTES

Na seção abaixo, é citado o desenvolvimento e testes utilizados no projeto.

5.1 API REST

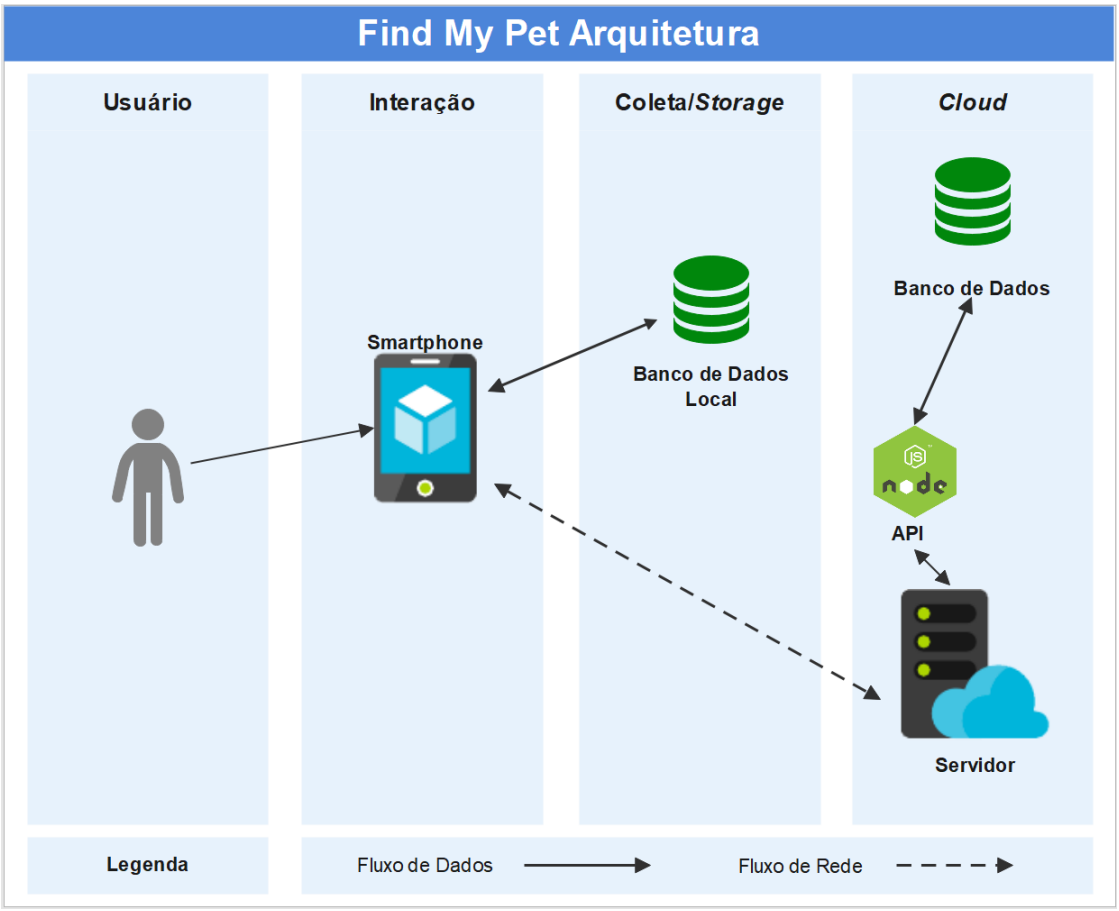
O Find MY PET foi construído de modo que atenda a demanda e obtenha informações sobre animais de estimação, foi desenvolvido um app mobile pensado nos tutores de pets. Para o desenvolvimento do aplicativo foi utilizado o framework React

Native, para não ser necessário desenvolver dois aplicativos, um para Android e outro para IOS, porém não foi gerado uma aplicação para dispositivos IOS. Enquanto que no backend, para a implementação da API, foi utilizado NodeJS. Para o acesso ao banco de dados foi utilizado Mongoose que é uma biblioteca baseada em schemas que serve para modelar os dados da aplicação.

A utilização de uma API de serviços do tipo REST em conjunto com a aplicação vem pela necessidade de que o sistema possa funcionar em qualquer ambiente, neste caso um aplicativo mobile que irá receber as informações e serviços da API a partir de requisições feitas através da plataforma.

Integrado com a plataforma *Amazon Web Services* (AWS) utilizando Amazon S3 um serviço de armazenamento de altos níveis de resiliência, garantindo a disponibilidade dos dados acessados com frequência e performance para acessá los, e adequado a uma grande variedade de casos de uso, como distribuição de conteúdo, aplicativos móveis entre outros, oferecendo soluções simples para o gerenciamento, possibilitando a organização dos dados e configuração dos controles de acesso, refinados para atender a requisitos específicos.

Figura 7: Arquitetura do modelo



Fonte: Desenvolvido pelo autor

Conforme a Figura 7, é possível visualizar a forma de interação no modelo, detalhada de acordo com seu ecossistema de funcionamento e relação de integração das aplicações, onde pode ser identificada a API do sistema, responsável pela realização de operações relacionadas ao armazenamento e manipulação no banco de dados que

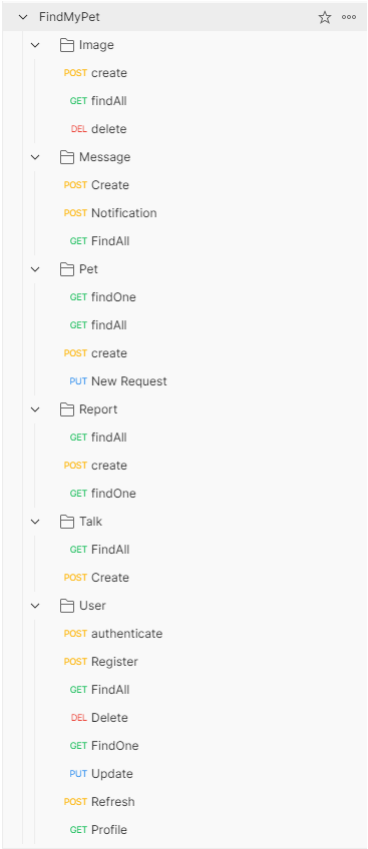
neste caso utiliza-se MongoDB, para então fornecê-las para a aplicação mobile e disponibilizar as mídias para o servidor da plataforma da AWS utilizando Amazon S3. Em relação ao *frontend* ou seja a interface que será responsável por dar origem aos dados e informações consumidas pelo sistema, pois um dispositivo não se comunica diretamente com o outro, as trocas de dados e solicitações apenas são realizadas junto a API, que faz o gerenciamento de todo o processo.

5.1.1 Controle de rotas e serviços

Com inúmeras rotas possíveis, que aumentam a cada novo *Controller* implementado no sistema, sendo estes executados por parte da API. Para fazer o gerenciamento destas rotas e endereços foi necessário utilizar um gerenciador e nele criar uma *Collection* de requisições. Para tornar possível o gerenciamento destas rotas, foi utilizado a ferramenta Postman, onde foram adicionadas as requisições de modo que cada módulo e necessidade fossem estabelecidos.

Como apresentado a seguir, na Figura 8, as pastas e subdivisões dos módulos foram definidas e adicionadas dentro de uma *Collection* denominada "FindMyPet", como por exemplo "*Report*" e "*Message*". E para cada módulo, estão registradas suas respectivas URL's que são utilizadas nas chamadas dos serviços, de acordo com seus formatos, GET, POST, PUT ou DELETE, além das informações passadas na solicitação que são necessárias ao realizar requisições diretamente na API.

Figura 8: Conjunto de requisições - Postman



Fonte: Desenvolvido pelo autor

5.1.2 Implementação

Foi implementada uma API do tipo *Representational State Transfer* ou REST, onde somente são liberadas as requisições solicitadas por usuários autenticados, exceto para o cadastro e login do usuário. Para o usuário ter acesso aos serviços que a API disponibiliza é necessário o envio de um *token*, JSON Web Token (JWT), assim o usuário é identificado e quais são os dados e serviços que este possui.

Figura 9: Retorno de lista de relatos no formato JSON


```
1  [
2      {
3          "images": [],
4          "_id": "6109fd02106f120022004398",
5          "description": "Teste",
6          "status": 1,
7          "situation": "P",
8          "user": {
9              "_id": "60f21dfd610656001bbf9db6",
10             "name": "Jonathan Machado",
11             "email": "jonathan.m@outlook.com",
12             "createdAt": "2021-07-17T00:02:05.470Z",
13             "__v": 0,
14             "image": "60f773ab610656001bbf9dc9",
15             "deviceToken":
                   "fP5hqEQLTeag2qfaFaJlIs:APA91bEnnhE5xkg2U7Dt0XOu_slu4STlseLemcpoBxcwpAjebi
                   Fqj8s_yrAWEVd5x8MuGvx4ExyIA4_zNsdLbTBZ8N62Ia0iwMGXnAptjH5t-gYlNCMSpIyX0cJL
                   0I70_0OJJ5KNeJ7h"
16         },
17         "pet": {
18             "_id": "6109fcfb106f120022004397",
19             "name": "Preta",
20             "birth": "2007-03-03T00:00:00.000Z",
21             "sexo": "F",
22             "size": "P",
23             "specie": "C",
24             "breed": "Pintcher",
25             "createdAt": "2021-08-04T02:35:39.294Z",
26             "user": "60f21dfd610656001bbf9db6",
27             "__v": 0,
28             "status": 1
29         },
30         "createdAt": "2021-08-04T02:35:46.325Z",
31         "__v": 0
32     },
33     {
```


Fonte: Desenvolvido pelo autor

Com a forte integração e a grande facilidade de uso do banco não relacional MongoDB, em conjunto com o ambiente Node.js, utilizando as duas tecnologias para o desenvolvimento da aplicações, tendo em vista a alta escalabilidade e levando em consideração um crescimento futuro, assim não terá a necessidade de enormes investimentos em virtude de aumento no número de servidores e replicação de banco de dados. Prevendo necessidades de melhorias futuras na plataforma, foi optado o uso de um banco NoSQL, já que este torna menos impactante e custosa as alterações em nível de estrutura.

A estrutura de informações presente no banco de dados MongoDB da *collection*,

apresentada na Figura 9, mostra uma estrutura de armazenamento, de relatos onde os campos "_id" que são gerados pelo próprio banco são destacáveis. Desta forma é mantido um identificador único, a fim de facilitar consultas objetivas ao documento e aos campos "user" e "pet", responsáveis por direcionar estes documentos às *collections* que armazenam tanto informações do usuário quanto informações do *pet*. Nesta estrutura, também são alocadas informações de descrição do relato, data de postagem, situação, também com informações relacionadas a localização do *pet*.

5.2 Aplicação móvel

A implementação do aplicativo desenvolvido em React Native, *framework* que está presente nas ferramentas como Facebook, Instagram, Discord, Skype, UberEats entre outros, tem como objetivo facilitar o acesso de usuários na plataforma através de dispositivos móveis Android, que foi escolhido devido ao alto custo de investimento e recursos de que seriam necessários para desenvolver a partir de uma plataforma IOS da Apple, pois um MacBook é obrigatório na implementação para dispositivos da marca (FACEBOOK, 2021b).

Visto que o *framework* gera código nativo Android, podemos garantir que a aplicação não terá perda de desempenho, pois embora seja implementada utilizando o React Native, o aplicativo é gerado com código nativo da plataforma, neste caso Android, diferente de outras tecnologias híbridas que por sua vez tendem a perder desempenho, devido ao fato de serem emuladas dentro do componente *webview*, navegador *web* nativo, da plataforma de dispositivos móveis.

Uma facilidade no uso da ferramenta é de que não precisa compilar o código a cada alteração enquanto está desenvolvendo, já que as mudanças salvas durante as implementações feitas no código são mostradas instantaneamente, por esse motivo se tem um ganho de produtividade em comparação com o tempo de desenvolvimento e os testes que foram realizados na aplicação.

A criação de componentes reutilizáveis na construção de telas no React Native é muito semelhante ao ReactJS, o processo utilizado no desenvolvimento da aplicação através de requisições, os dados recebidos são manipulados e injetados em componentes para serem apresentados na tela dos dispositivos dos usuários.

5.3 Testes

Quando o desenvolvimento da aplicação chega na etapa de finalização, são realizados os testes efetuados pelo desenvolvedor, que irão explorar a aplicação em busca de erros, o teste parte com foco nos requisitos não funcionais, por exemplo usabilidade, desempenho, confiabilidade. A realização dos testes foi pensada com o intuito de cobrir as falhas não encontradas durante os testes realizados no processo de desenvolvimento, a fim de corrigir problemas relacionados à otimização e desempenho.

Os testes de funcionamento básico realizados na aplicação de acordo com algumas necessidades, seguem um roteiro do que se é esperado por cada funcionalidade. Pensando em cobrir erros relacionados a API, mesmo após terem sido testadas durante o seu desenvolvimento, as requisições, chamadas individualmente de cada método, foram testadas novamente com a finalidade de garantir que não haja recursos disponíveis, devido correções feitas na fase de testes.

6 RESULTADOS

Nesta seção, são apresentados os resultados no desenvolvimento do projeto.

6.1 API REST

Com o intuito de manter a segurança de acesso às informações, possibilitando sua manipulação e distribuição de várias maneiras, foi criada uma API *service* do tipo REST, na qual as requisições feitas através das aplicações *front-end* somente são liberadas a usuários autenticados, exceto em requisições de login e cadastro do usuário. Como pode ser visto na Figura 10, o *log* de registros mostra a inicialização da API *service*, onde é destacado o status como online e que as chamadas são recebidas na porta 3000. E por fim os *logs* de requisições que foram realizadas, são exibidos na plataforma do Umbler, servidor onde a API está alocada.

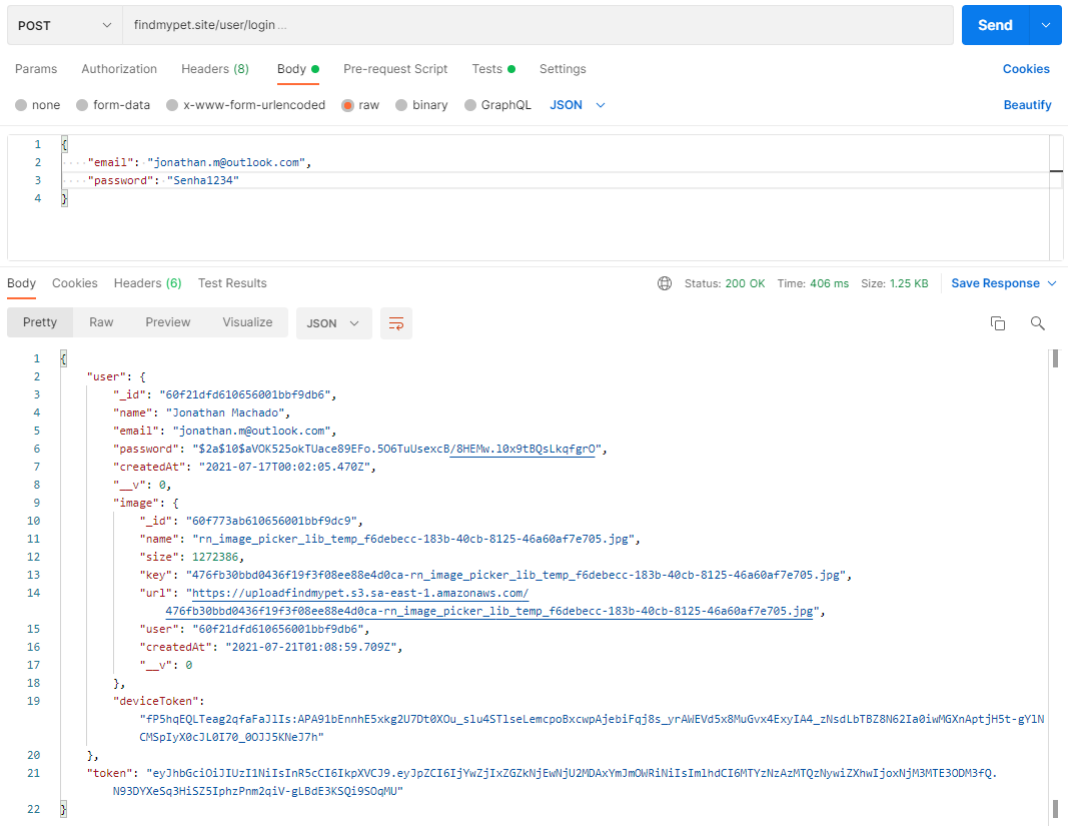
Figura 10: Servidor NodeJS inicializado

```
1  Nov 15 17:38:07   Listening on 3000...
2  Nov 15 17:38:04
3  Nov 15 17:38:04   > node dist/index.js
4  Nov 15 17:38:04   > api_findmypet@1.0.0 start /usr/src/app
5  Nov 15 17:38:04
6
```

Fonte: Desenvolvido pelo autor

Na Figura 11 é apresentada uma requisição feita utilizando a ferramenta Postman, que passa os dados de identificação do usuário ou seja email e senha, que são parâmetros enviados à API, através de um dos métodos HTTP responsáveis por determinar a operação a ser realizada. Um token é gerado pela API e retornado ao aplicativo quando os dados enviados conferem com os digitados no momento do cadastro. Após esse mesmo token será utilizado pelo sistema para autenticação de acesso à aplicação.

Figura 11: Servidor NodeJS inicializado



Fonte: Desenvolvido pelo autor

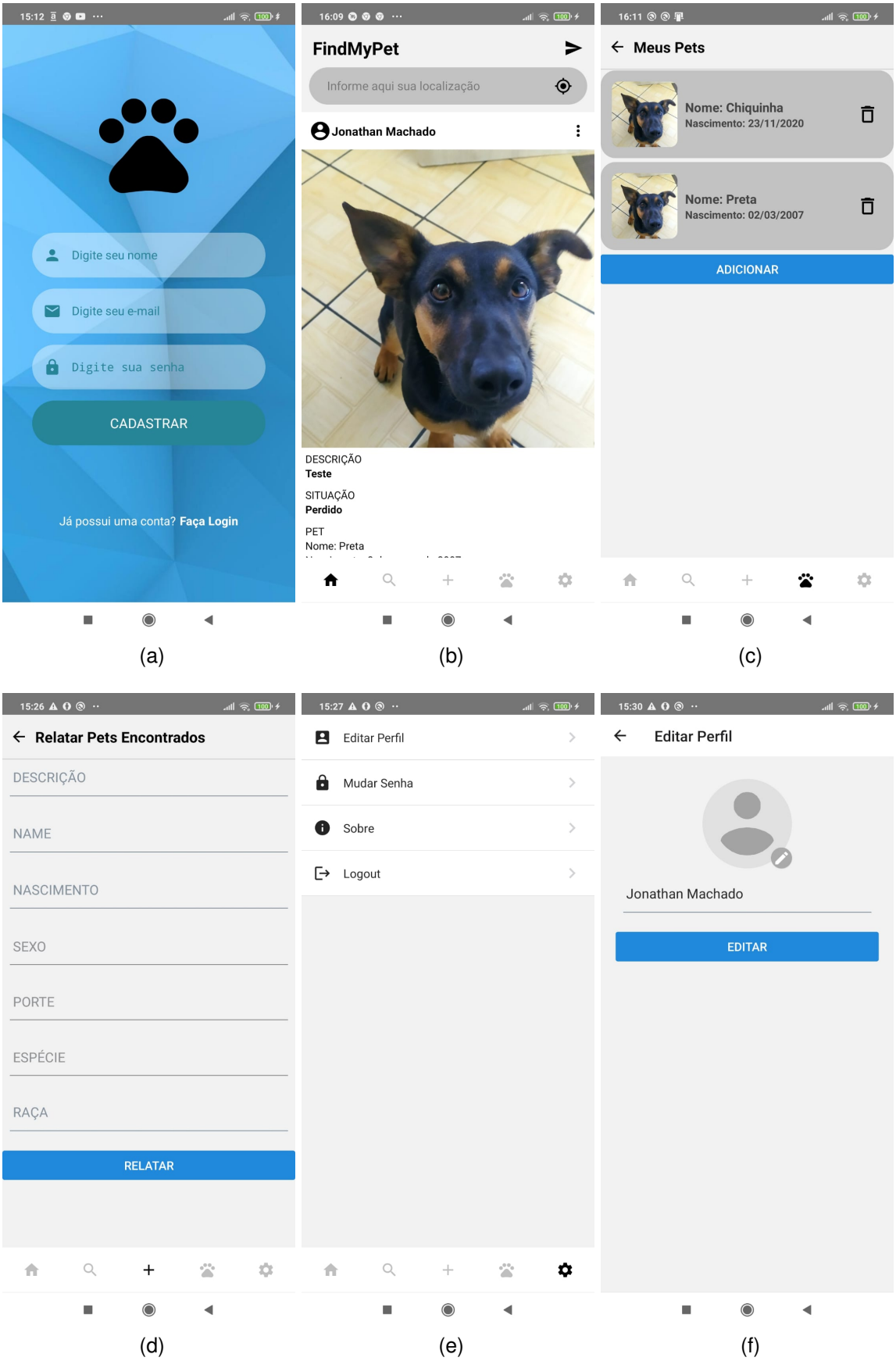
6.2 Aplicação móvel

Para desenvolver o aplicativo, que irá servir como uma ferramenta que ajuda na busca de animais desaparecidos. Foi optado pela plataforma Android, pelo fato de estar presente em diversos dispositivos atualmente utilizados, além de ser uma plataforma muito estável e ter sido construída pelo Google (ANDROID, 2021).

O cadastro que é solicitado ao usuário no primeiro acesso à aplicação, que deverá preencher um formulário com os seguintes dados: nome, e-mail e senha, conforme pode ser visto na Figura 12a.

Finalizado esta etapa de cadastro, os dados informados pelo usuário são enviados para a API, que ficará responsável pelo processo de validação dos dados, do usuário. Estando tudo correto é realizada a inserção do novo usuário no Banco de Dados MongoDB, que realiza a autenticação interna, e retorna um *token* JWT para o aplicativo e redirecionando para a tela principal. Em caso de informações incorretas, uma mensagem de erro é retornada pela API, que em seguida é exibida ao usuário pelo dispositivo móvel, com o erro de validação ocorrido.

Figura 12: Telas do aplicativo



Fonte: Desenvolvido pelo autor

Feito isso, fica a critério do usuário fornecer mais informações ao sistema, como por exemplo: se tem animais de estimação, seus nomes e idades. Desta forma o usuário tem acesso liberado para que possa criar seus próprios relatos de animais, fornecendo informações de características, descrição e localização.

Como apresentado na Figura 12b, na tela principal são exibidos os relatos feitos pelos usuários do sistema, onde são apresentadas informações como: descrição, situação, uma foto, nome, idade, porte, sexo, localização, tanto sobre os animais perdidos, quanto dos animais encontrados. Neste exemplo da Figura 12b podemos ver um relato de um *pet* que está perdido, tem o nome Preta, possui uma foto para servir de ajuda na sua identificação. Neste mesmo *card* no canto superior direito tem um botão *pop up*, com a opção excluir para o usuário apagar o *post*.

Na Figura 12c é apresentado a tela com os pets adicionados pelo usuário, no qual é possível adicionar novos, preenchendo o formulário com alguns dados como: Imagem, nome, data de nascimento, porte, sexo, raça ou editar o perfil dos animais que foram cadastrados no sistema, como adicionar uma nova foto ou alterar algum erro dos dados que foram informados. Na Figura 12d pode-se visualizar a tela para relatar animais encontrados, podendo preencher algumas informações no formulário presente na tela.

A Figura 12e apresenta a tela que exibe as de configurações do aplicativo, no qual é possível editar o perfil, mudar a senha da conta, o sobre, que apresenta as políticas do aplicativo, e por fim a opção para sair (fazer logout) do aplicativo. Na Figura 12f é possível visualizar a tela para edição do perfil, podendo colocar uma foto sua, selecionando uma existente ou tirar uma foto atual, também é permitido realizar a troca de informações do usuário.

7 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho de pesquisa evidenciado no presente artigo, apresentou-se a aplicação Find My Pet, um aplicativo implementado e desenvolvido que irá conectar tutores com animais resgatados por meio de uma plataforma integrada, que tem como intuito beneficiar o reencontro dos animais e seus tutores, possibilitando o contato entre os usuários da aplicação, de um lado quem busca encontrar seu *pet* perdido e do outro o usuário que encontrou o animal desaparecido, contato esse vinha sendo feito a partir das redes sociais.

O FindMyPet foi desenvolvido utilizando a abordagem de API REST, utilizando Node JS e também um aplicativo utilizando React Native, uma biblioteca JavaScript usada no desenvolvimento de aplicativos para os sistemas Android e iOS.

Durante o desenvolvimento deste projeto, identificou-se inúmeros benefícios na utilização de uma API REST. Facilitando a disponibilização, uso e manipulação das funcionalidades do web service, como também, os recursos disponíveis para o aplicativo.

Do mesmo modo que o web *service* facilita na adesão de novas aplicações, a escolha pela biblioteca React Native, deu-se ao fato de que a tarefa para o desenvolvimento de uma aplicação voltada para a plataforma IOS, acaba por se tornar mais simples, com o uso da biblioteca em uma melhoria futura.

Entre os resultados obtidos, é destacado que utilizando um sistema mobile atrelado a recursos de geolocalização, foi possível otimizar a apresentação dos relatos e o desempenho da aplicação. Em relação às informações coletadas pelo sistema, se

obtém um reaproveitamento e customização do sistema através de cadastros realizados pelos usuários, visto que são comparados relatos novos com os que já haviam sido publicados, o sistema após armazenar os dados compara este relato com relatos ativos no sistema e notifica cada usuário que um novo relato com informações semelhantes foi postado por outro usuário. A aplicação mostrou ter efetividade no seu intuito, visto que o uso do sistema irá contribuir para que o usuário não tenha o retrabalho de fazer múltiplas postagens da publicação, como ocorre em outras plataformas.

Estuda-se a implementação de um projeto futuro, que por questões de prazo não seria possível implementar uma solução neste momento. Dentre elas. destaca-se: Oferecer aos usuários o uso de tecnologias vestíveis com o uso de *wearables*(em português, computação vestível), que nesse caso seriam coleiras inteligentes que fariam o monitoramento constante do *pet* e caso ele se afaste do dono ou da casa, o sistema iria emitir um sinal de aviso ao usuário; Promover um sistema de anúncios, para empresas de *pet* shop e/ou clínicas veterinárias, que serão oferecidos a usuários de suas proximidades que seria a forma de renda ao sistema; Criar um espaço para divulgar animais para adoção; Oferecer um espaço a Organizações Não Governamentais (ONGs) de animais para divulgarem seus espaços e fazerem pedidos de doação.

REFERÊNCIAS

ABINPET. **Informações gerais do setor Pet**. 2021. Disponível em: <http://abinpet.org.br/infos_gerais/>. Acesso em: 12 out. 2021.

ANDROID. **Android Open Source Project**. 2021. Disponível em: <https://source.android.com/>. Acesso em: 30 out. 2021.

CAMPBELL, C. 'they are overwhelmed.'china's animal shelters can't cope with the number of pets abandoned due to covid-19. **Time**, 2020.

CAMPOS, C. I.; WALCZAK, F. S.; SOUZA, A. O. de. A tecnologia como ferramenta estratégica para a formação das competências e do marketing pessoal. **Revista de Ciências Gerenciais**, v. 14, n. 19, p. 239–248, 2010.

CHIAVENATO, I. **Gestão de pessoas**. [S.l.]: Elsevier Brasil, 2005.

CURIOSAMENTE. **Brasileiro cria aplicativo para encontrar animais perdidos**. 2017. Disponível em: <http://curiosamente.diariodepernambuco.com.br/project/jovem-paulista-cria-aplicativo-para-encontrar-animais-perdidos/>. Acesso em: 25 out. 2021.

DIVINO, L. Pandemia e o crescente aumento na adoção de animais domésticos. **Revista Gestão & Tecnologia**, v. 1, n. 30, p. 33–35, 2020.

FACEBOOK. **Animais perdidos e encontrados em igrejinha**. 2021. Disponível em: <https://www.facebook.com/groups/616494848756618>. Acesso em: 18 out. 2021.

FACEBOOK, I. **Learn once, write anywhere.** 2021. Disponível em: <https://reactnative.dev/>. Acesso em: 12 out. 2021.

FORBOMB. **CADE.PET ANIMAIS PERDIDOS OU DESAPARECIDOS**. 2019. Disponível em:

<https://play.google.com/store/apps/details?id=com.cade.pet.cadepet&hl=pt_BR&gl=US>. Acesso em: 25 out. 2021.

G1. **Facebook completa 15 anos com 2,3 bilhões de usuários: Forte crescimento da empresa se reflete nos lucros e também nas crescentes polêmicas..** 2019. Disponível em: <https://g1.globo.com/economia/tecnologia/noticia/2019/02/04/facebook-completa-15-anos-com-23-bilhoes-de-usuarios.ghtml/>. Acesso em: 25 out. 2021.

IPB. **Censo Pet: 139,3 milhões de animais de estimação no Brasil**. 2019. Disponível em: <http://institutopetbrasil.com/imprensa/censo-pet-1393-milhoes-de-animais-de-estimacao-no-brasil/>. Acesso em: 18 ago. 2021.

IPB. **População pet mundial cresce liderada por gatos**. 2021. Disponível em: <http://institutopetbrasil.com/fique-por-dentro/populacao-pet-mundial-cresce-liderada-por-gatos/>. Acesso em: 18 ago. 2021.

LIMA, G. **REST: Conceito e fundamentos**. 2020. Disponível em: <https://www.alura.com.br/artigos/rest-conceito-e-fundamentos?gclid=Cj0KCQiA-eeMBhCpARIsAAZfxZD0FpYsnjlcMAX0TAFkcAIUsBCHZ41WpvdKSV-Dq9dqYklzRLv5fxQaAis_EALw_wcB>. Acesso em: 20 nov. 2021.

MOREIRA, T. P. et al. Where is my puppy? retrieving lost dogs by facial features. **Multimedia Tools and Applications**, Springer, v. 76, n. 14, p. 15325–15340, 2017.

PADULLA, P. L. F. L. **Pelo fim dos fogos de artifício!** 2017. Disponível em: <https://biologosocialista.wordpress.com/2017/01/06/pelo-fim-dos-fogos-de-artificio/>. Acesso em: 25 out. 2021.

PEREIRA, C. R. **Aplicações web real-time com Node. js**. [S.l.]: Editora Casa do Código, 2014.

PEREIRA, C. R. **Construindo APIs REST com Node. js: Caio Ribeiro Pereira**. [S.l.]: Editora Casa do Código, 2016.

PERSICO, V.; MONTIERI, A.; PESCAPE, A. On the network performance of amazon s3 cloud-storage service. In: IEEE. **2016 5th IEEE International Conference on Cloud Networking (Cloudnet)**. [S.l.], 2016. p. 113–118.

POSTMAN. **Postman API Plataform**. 2021. Disponível em: <https://www.postman.com/>. Acesso em: 12 out. 2021.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software - 8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016.

SERVICES, I. A. W. **Amazon S3 REST API Introduction**. 2021. Disponível em: <https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html>. Acesso em: 25 out. 2021.

SERVICES, I. A. W. **Amazon Simple Storage Service**. 2021. Disponível em: <https://aws.amazon.com/pt/s3/>. Acesso em: 25 out. 2021.