



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

FILIPE PIRES GUIMARÃES

**ADOÇÃO PET CG:
UM APLICATIVO PARA SIMPLIFICAR O PROCESSO DE ADOÇÃO DE
ANIMAIS EM CG**

CAMPINA GRANDE - PB

2021

FILIFE PIREES GUIMARÃES

**ADOÇÃO PET CG:
UM APLICATIVO PARA SIMPLIFICAR O PROCESSO DE ADOÇÃO
DE ANIMAIS EM CG**

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. José Antão Beltrão Moura

CAMPINA GRANDE - PB

2021



G963a Guimarães, Filipe Pires.

ADOÇÃO PET CG: Um aplicativo para simplificar o processo de adoção de animais em CG. / Filipe Pires Guimarães. - 2021.

13 f.

Orientador: Prof. Dr. José Antão Beltrão Moura.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Adoção de animais. 2. Desenvolvimento de aplicação. 3. React-Native. 4. Aplicativo para adoção de animais. 5. Cadastro de Pets. 6. Biblioteca de javascript. I. Moura, José Antão Beltrão. II. Título.

CDU:004.415.2(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

FILIPPE PIRES GUIMARÃES

**ADOÇÃO PET CG:
UM APLICATIVO PARA SIMPLIFICAR O PROCESSO DE ADOÇÃO
DE ANIMAIS EM CG**

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. José Antão Beltrão Moura

Orientador – UASC/CEEI/UFCG

Professor Dr. Tiago Lima Massoni

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 25 de Maio de 2021.

CAMPINA GRANDE - PB

RESUMO (ABSTRACT)¹

Um grande problema das cidades brasileiras é o alto índice de animais abandonados, que segundo uma pesquisa da OMS existem mais de 20 milhões de cães sem lares no Brasil [1]. Além desse número muito alto, há um outro problema na adoção que é adotar um animal, não se adaptar ao seu comportamento, temperamento daquele animal, então algumas pessoas acabam abandonando esses animais, novamente. Existem ONGs e pessoas que gostariam de fazer a diferença nesse meio. Nessa plataforma, auxiliaremos de algumas formas, como: ajudar na avaliação do adotante e do pet, mostrar os animais disponíveis para adoção podendo classificar e/ou filtrar, além disso abrir um canal de comunicação entre as organizações e os adotantes. Para o desenvolvimento desta plataforma, será utilizada uma biblioteca de javascript bem conhecida no mercado, o React-Native [2]. Espera-se que o aplicativo sirva para disseminar a ideia da adoção de animais, além de ajudar a evitar a reintrodução desse animal ao abandono e assim aumentar o número de animais adotados.

¹ A major problem in Brazilian cities is the high rate of abandoned animals, which, according to a OMS survey, has more than 20 million homeless dogs in Brazil [1]. In addition to this very high number, there is another problem with adoption, which is adopting an animal, not adapting to its behavior, that animal's temperament, so some people end up abandoning these animals, again. There are NGOs and people who would like to make a difference in this environment. In this platform, we will assist in some ways, such as: helping in the evaluation of the adopter and the pet, showing the animals available for adoption, being able to classify and/or filter, in addition to opening a communication channel between the organizations and the adopters. For the development of this platform, a well-known javascript library, React-Native [2], will be used. It is hoped that the application will serve to disseminate the idea of adopting animals, in addition to helping to prevent the reintroduction of that animal to abandonment and thus increase the number of adopted animals.

ADOÇÃO PET CG: UM APLICATIVO PARA SIMPLIFICAR O PROCESSO DE ADOÇÃO DE ANIMAIS EM CG

Filipe Pires Guimarães
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil
filipe.guimaraes@ccc.ufcg.edu.br

RESUMO

Um grande problema das cidades brasileiras é o alto índice de animais abandonados, que segundo uma pesquisa da OMS existem mais de 20 milhões de cães sem lares no Brasil [1]. Além desse número muito alto, há um outro problema na adoção que é adotar um animal, não se adaptar ao seu comportamento, temperamento daquele animal, então algumas pessoas acabam abandonando esses animais, novamente. Existem ONGs e pessoas que gostariam de fazer a diferença nesse meio. Nessa plataforma, auxiliaremos de algumas formas, como: ajudar na avaliação do adotante e do pet, mostrar os animais disponíveis para adoção podendo classificar e/ou filtrar, além disso abrir um canal de comunicação entre as organizações e os adotantes. Para o desenvolvimento desta plataforma, será utilizada uma biblioteca de javascript bem conhecida no mercado, o React-Native [2]. Espera-se que o aplicativo sirva para disseminar a ideia da adoção de animais, além de ajudar a evitar a reintrodução desse animal ao abandono e assim aumentar o número de animais adotados.

PALAVRAS-CHAVES

Adoção, pets, animais, React Native, aplicativo, mobile, android

1. INTRODUÇÃO

Com a crescente alta de animais abandonados, uma das opções para a resolução desse problema é a adoção de animais.

Mas com um número tão alto de animais, fica quase que impossível que as ONGs, que realizam esse processo de adoção, conheçam todas as características dos animais que estão ali para adoção. E assim, a adoção dos pets virará uma grande surpresa, para bem ou para mal, já que essas pessoas que querem adotar não saberão se aquele animal se adapta bem ao novo lar ou se aquele sonho de ter um animal em casa irá virar um pesadelo, podendo chegar ao ponto de o animal ser abandonado, novamente. Segundo uma pesquisa realizada pelo Centro de Zoonoses do Distrito Federal [3], com a pandemia do covid-19, o número mais do que dobrou entre o período de janeiro e setembro de 2020, em relação ao ano anterior.

E para tratar dessa carência, este projeto visa produzir uma ferramenta (app móvel) para apoiar tutores, animais e as ONGs. O app irá conter uma base de dados para os animais das ONGs, com suas características únicas, e que indicará a compatibilidade entre os tutores e animais para que ajudem nesse processo de adaptação.

2. SOLUÇÃO

Com os problemas apresentados anteriormente, a proposta para o projeto é aproximar os usuários de organizações, para aumentar o número de animais adotados, além de maximizar a compatibilidade entre os animais e os adotantes, para evitar a reintrodução desse mesmo animal ao abandono.

Segundo um levantamento feito pela *Brain & Company* [4], 9 em cada 10 pessoas, que utilizam celulares, usam o sistema operacional *Android*, portanto, visando um maior número de usuários, foi inicialmente escolhida essa plataforma para o desenvolvimento da aplicação.

Para o levantamento dos requisitos, foram consultadas ONGs e pessoas com perfil de adotantes. Com base nos requisitos vistos, foi definido um escopo com as funcionalidades mais essenciais, que são:

Contas de usuário organização e usuário adotante: Algumas funcionalidades foram definidas baseadas na autenticação, ou não, no aplicativo. Dessa forma, tomamos como base 2 tipos de usuários, um autenticado (tido como usuário organização) e o não-autenticado (tido como usuário adotante).

Cadastro de organização: Uma organização pode cadastrar-se no app, através de um formulário *in-app*;

Atualização dos dados do usuário organização: Uma organização pode atualizar os seus dados, como: endereço, foto, telefone, entre outros dados;

Cadastro de pets disponíveis para adoção: O usuário organização pode cadastrar "anúncios" de novos pets disponíveis para adoção;

Remoção de pets disponíveis para adoção: O usuário organização pode retirar o seu anúncio, anteriormente adicionado;

Lista com os pets disponíveis para adoção: O usuário adotante pode ver a lista completa dos pets disponíveis, enquanto o usuário organização pode somente ver sua lista de pets;

Mapa com endereços e contatos das organizações: É possível que os dois usuários visualizem os endereços das organizações cadastradas no sistema, além de visualizar alguns dados, como: telefone, email;

Envio de notificação quando um pet for cadastrado: Ao ser criado um novo "anúncio", deve ser enviado a todos os usuários do tipo adotantes, para ficarem sabendo sobre esse novo pet.

2.1 ARQUITETURA

O sistema foi dividido em 2 sub-sistemas: o *back-end* e o *front-end*. O primeiro é responsável pela persistência dos dados da aplicação, incluindo o *storage* das imagens. Enquanto o *front-end* é responsável pela interação com o usuário final, onde se fazem requisições para o *back-end* através da API do *SDK*.

O *back-end* utiliza um banco de dados de nuvem *NoSQL*, para a estruturação dos dados.

Seguindo um modelo arquitetural C4 [12], temos nosso contexto [Figura 1], nossos containers e nossos componentes, os quais serão apresentados mais à frente em cada subtópico. O código que implementa a arquitetura pode ser visto no [repositório](#).

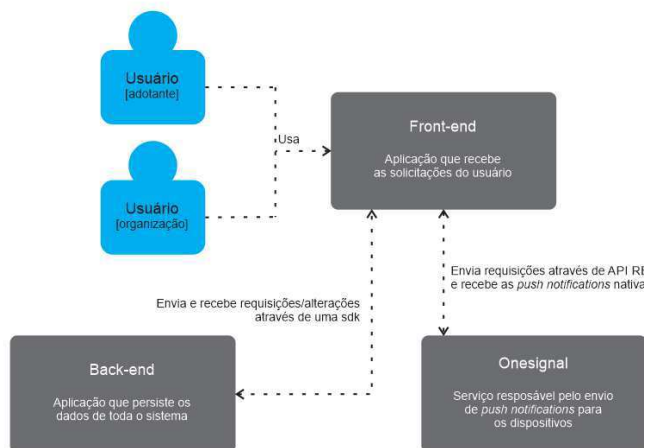


Figura 1 - Diagrama de contexto sistema

2.1.1 BACK-END

Para o desenvolvimento do *back-end*, foi utilizado o serviço do *Onesignal* [10] e alguns do *Firebase* [11], vide Figura 2.

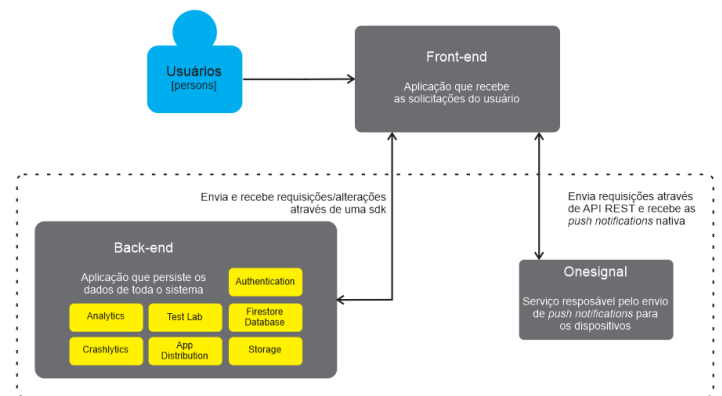


Figura 2 - Diagrama de container back-end

Onesignal: Foi utilizado a API Rest, para o recebimento de um novo evento para disparar um *trigger* interno para disparar as *push notifications* para os dispositivos;

Authentication: Serviço ao qual é possível fazer diferentes tipos de login, como: por email e senha, por algumas redes sociais, por número de celular, entre outros. Nesse projeto foi utilizado o método por "login e senha". Ele se relaciona diretamente com dois outros serviços, o *Firestore Database* e o *Storage*, pois só é possível fazer inserção de novos dados se o usuário estiver cadastrado;

Firestore Database: Serviço para persistência dos dados do sistema, onde tem-se como regra que: qualquer usuário pode ler, porém apenas os usuários logados podem fazer a criação, atualização e remoção dos dados;

Storage: Serviço para persistir arquivos do sistemas, estes podem ser imagens dos pets e/ou das organizações. Tem a mesma regra de leitura/gravação que o *Firestore*;

Analytics: Serviço para analisar o aplicativos em tempo real, para tomada de ações para aumentar engajamento e/ou outras métricas;

Crashlytics: Serviço para monitoramento de problemas para quando o produto estiver em versão *release*;

Test Lab: Serviço para teste de usabilidade em diversos tipos de dispositivos móveis;

App Distribution: Serviço para distribuição do aplicativo sem a necessidade de fazer o *deploy* na *PlayStore*;

2.1.1.1 TECNOLOGIAS DO BACK-END

No *back-end*, foi utilizado o serviço do *Firebase*, para agilizar o desenvolvimento do sistema.

É um serviço baseado em um *SDK*, que possui algumas ferramentas, entre elas o *authentication* (serviço para auxiliar no cadastro/login do sistema, nele é armazenado o token *JWT* para a autenticação), o *firebase database* (serviço para armazenamento

de dados não-relacionais, como os dados dos pets e dos usuários cadastrados) e o *storage* (serviço para armazenamento dos arquivos da aplicação, nessa aplicação foi utilizada para armazenar as imagens enviadas pelas organizações).

2.1.1.2 ESTRUTURA DO BACK-END

Com a ferramenta *authentication*, foi utilizado o método por "email/senha", esse método que irá diferenciar o usuário organização do usuário adotante. Quando feito o cadastro, o usuário envia os seus dados para esse serviço, após isso é criado um *bucket* no *storage* com a imagem da organização e também é criado um novo documento no *firestore database*, com esses dados e a *url* da imagem, caso possua.

Para a exibição dos dados de pets [Figura 3] e das organizações [Figura 4], é feito uma chamada ao *SDK* e é retornado os dados, nas seguinte estrutura:

```
interface Organization {
  email: string;
  name: string;
  phone: string;
}

interface Pet {
  age: string;
  breed: string;
  deleted_at?: string;
  description: string;
  organization: Organization;
  picture?: string;
  type: string;
}
```

Figura 3 - Estrutura de dados da coleção Pets

```
interface User {
  coordinate: {
    latitude: number;
    longitude: number;
  };
  description: string;
  name: string;
  phone: string;
  picture?: string;
}
```

Figura 4 - Estrutura de dados da coleção Users

Também foi utilizado um serviço externo para notificações *push* chamado *Onesignal*, que a comunicação é feita por uma requisição à sua *API REST*, e que dispara as notificações para os dispositivos: *android*, *iOS* e até mesmo *web*.

2.1.2 FRONT-END

Já para o desenvolvimento do aplicativo, foi utilizada a modelagem abaixo [Figura 5], e que será explicado posteriormente.

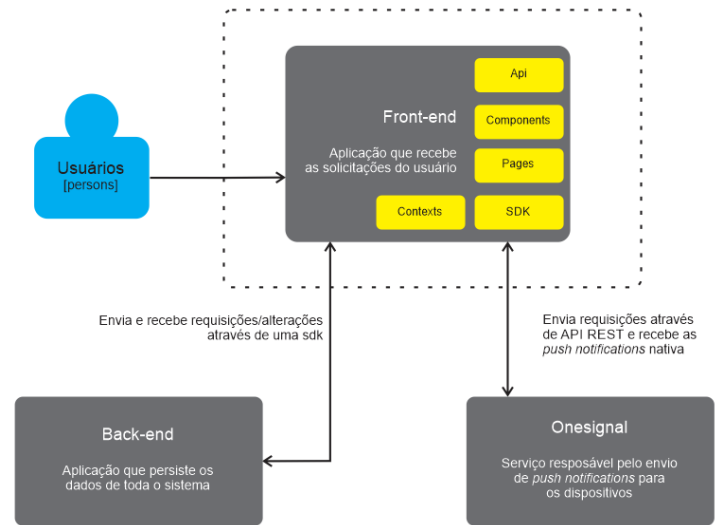


Figura 5 - Diagrama de container do front-end

2.1.2.1 TECNOLOGIAS DO FRONT-END

Foi utilizado o *React-Native* para o *front-end*, por sua principal característica de criar aplicativos nativos híbridos para *iOS*, *android* e até mesmo *web*, com a mesma *codebase*, além de que sua comunidade é numerosa e o mercado está aquecido por ser utilizada por grandes empresas como Facebook, Instagram, entre outras. O escopo do projeto foi definido somente como para *android*, pois para *iOS*, temos limitações referentes ao SO, como o build da aplicação somente é disponível quando se possui uma licença de desenvolvimento, que é paga anualmente, e isso tornaria inviável financeiramente nesse ponto da pesquisa. O design todo foi feito com uma ferramenta *styled-components* e construída em *css*. A navegação foi desenvolvida com *react-navigation*, onde algumas telas têm restrição de uso e algumas telas mudam até o design baseado no tipo de usuário. Além disso, foi utilizado o *Google Maps* e o serviço de geolocalização, também da *Google*, para o posicionamento dos usuários.

Também foram utilizadas ferramentas para padronização de código e de commit, e também utilizado *gitflow*, para estruturação do *git*.

2.1.2.2 ESTRUTURA DO FRONT-END

Esse projeto foi estruturado seguindo ajustes feitos no padrão sugerido em [7], conforme [Figura 6].

Em **api**, temos um arquivo que faz requisições a *API REST* do *Onesignal*.

Em **assets**, temos o diretório em que ficam localizadas as imagens e ícones utilizados na aplicação.

Em **components**, temos o diretório para componentes em que podem ser reutilizados globalmente na aplicação, seguindo o padrão de componentização sugerido [8].

Em **contexts**, temos os arquivos para gerenciamento de estados da aplicação, como um todo.

Em **pages**, temos as telas do aplicativo, que substitui *layouts*, do padrão sugerido.

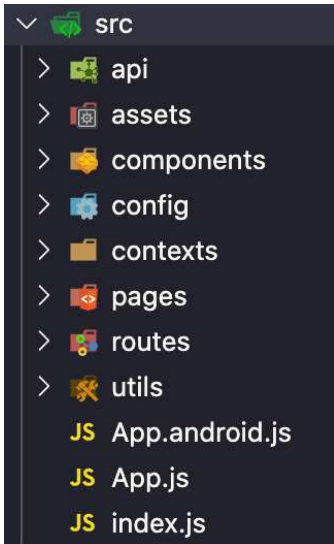


Figura 6 - Estrutura de pastas

2.1.3 LÓGICA DA APLICAÇÃO

Mapa: Ao abrir o aplicativo, será visualizado, diretamente, o mapa com a localização atual do dispositivo além de marcações das organizações já cadastradas no sistema [Figura 7].

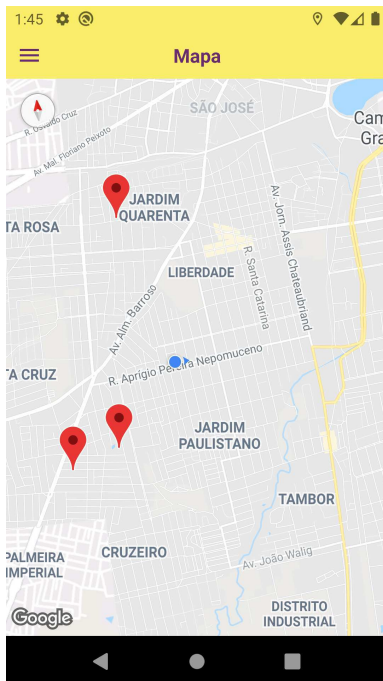


Figura 7 - Tela de mapa do usuário adotante

Para os usuários do tipo organização, temos o mesmo mapa porém com uma *tab bar*, onde tem-se outras funcionalidades, como adicionar um pet, visualizar o mapa e/ou visualizar/alterar os dados cadastrais [Figura 8].

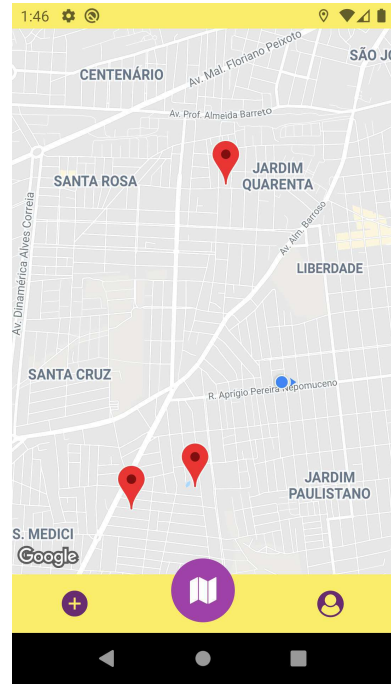


Figura 8 - Tela de mapa do usuário organização

Barra lateral: A barra lateral também é personalizada para cada tipo de usuário. Para os usuários do tipo adotantes, temos a opção de visualizar o mapa e visualizar toda a lista de pets disponíveis para adoção, além dessas opções temos um botão para redirecionar para a tela de login [Figura 9].

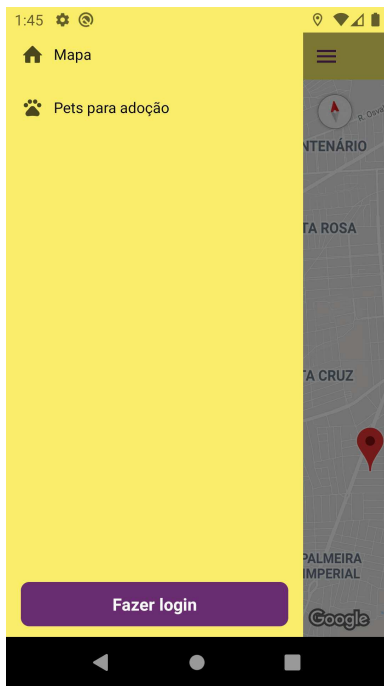


Figura 9 - Barra lateral do usuário adotante

Quando o usuário do tipo organização estiver ativo, teremos a opção de *Home*, onde voltaria para o mapa inicial, e também temos a opção de *Meus anúncios* que exibe a lista de pets adicionadas por esse usuário. E temos um botão para fazer o logout da aplicação [Figura 10].

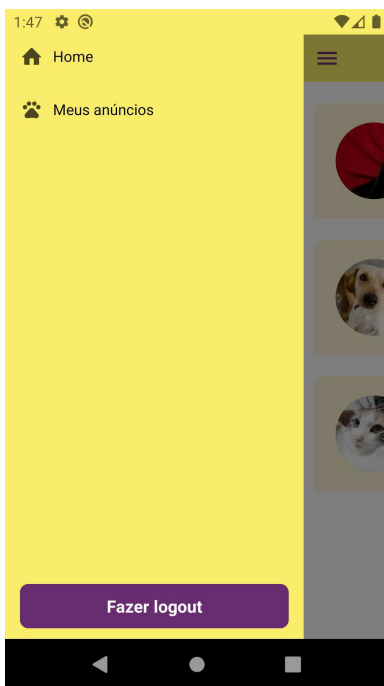


Figura 10 - Barra lateral do usuário organização

Lista de pets: Na listagem de pets, ainda temos a restrição no sistema baseada no tipo de usuário.

Para os usuários adotantes, pode-se visualizar algumas informações mais relevantes de cada pet, além de visualizar dados da organização que cadastrou o pet e/ou até mesmo ligar para essa organização [Figura 11].



Figura 11 - Lista de pets do usuário adotante

O usuário do tipo organização pode visualizar os pets cadastrados por si, além de que também pode deletar do sistema, para casos de adoção ou qualquer outro motivo [Figura 12].



Figura 12 - Lista de pets do usuário adotante

Login: É destinado para organizações, onde vão possuir acesso às funcionalidades desse tipo de usuário. É necessário preencher os campos e pressionar o botão. Nessa tela, também é possível continuar a utilizar o app sem logar ou de fazer o cadastro no sistema [Figura 13].

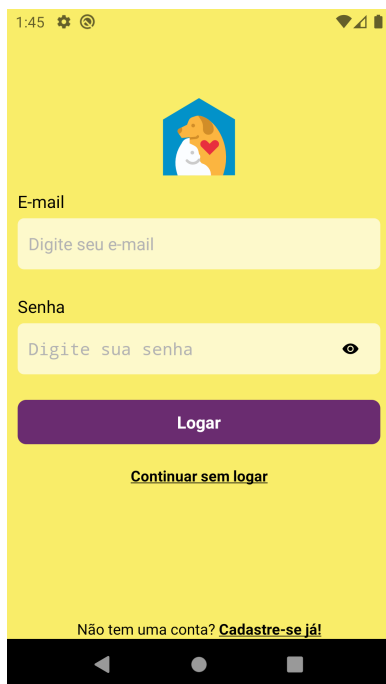


Figura 13 - Login do usuário

Cadastro: Para criar uma conta do tipo organização é necessário preencher alguns dados, como: nome, endereço, telefone, uma breve descrição que será exibida para os usuários, e três campos para autenticação, o e-mail, uma senha e uma confirmação de senha. Além disso, é possível também enviar uma foto da organização, ao pressionar a foto no primeiro campo da tela [Figura 14].



Figura 14 - Cadastro do usuário organização

Perfil: Com o usuário logado, ou seja o usuário organização, ele pode visualizar seus dados cadastrados, além disso pode editar os seus dados cadastrais [Figura 15].



Figura 15 - Atualização de dados cadastrais do usuário organização

Adicionar pet: Nessa funcionalidade, o usuário organização pode inserir novos dados como o tipo do animal, imagem, nome, idade, entre outros e enviar para o sistema, que deve enviar um *push notification* para todos os usuários adotantes [Figura 16].



Figura 16 - Adicionar "anúncio" de pet disponível

2.1.4 AMBIENTE DE PRODUÇÃO

Como foi utilizado um projeto *firebase* para o back-end, temos uma gama de ferramentas que podem ser utilizadas, e algumas realmente foram, como o *crashlytics* (que é uma forma para debugar possíveis erros no aplicativo em modo release), o *analytics* (para analisar os dados dos usuários em tempo real, para aumentar o engajamento na aplicação), e outras para o monitoramento do mesmo.

O aplicativo foi disponibilizado pela ferramenta chamada *App Distribution* do *Firebase* para o compartilhamento entre alguns usuários, por ser um *MVP* não houve ainda a necessidade de hospedar o aplicativo na *play store*.

3. EXPERIÊNCIA

3.1 PROCESSO DE DESENVOLVIMENTO

O projeto e desenvolvimento do aplicativo seguiu um processo iterativo e incremental para desenvolvimento ágil de software (SCRUM [13]).

Para montar o escopo, foi feita uma pesquisa em outros aplicativos [5][6], para definir quais funcionalidades seriam mais essenciais para que sejam desenvolvidas no tempo estimado.

Foi definido um cronograma, para o desenvolvimento das funcionalidades gradativamente.

Inicialmente, foi feito um levantamento das tecnologias para serem utilizadas no aplicativo, e escolhido o que mais daria agilidade no processo de desenvolvimento.

O projeto foi dividido em 2 sprints. Para a primeira sprint, foi feito o levantamento dos requisitos, com os usuários com perfis de adotantes e com os usuários com perfis de organização, para ampliar os conhecimentos. Para a segunda etapa, foi feito o desenvolvimento da aplicação, foi feito a inicialização/configuração do projeto, criado as telas para uma melhor satisfação visual do usuário, foi feito o desenvolvimento das telas e em paralelo com a integração da API, e por último foi feito a integração com o *google maps*, para exibição do mapa e o *autocomplete* do endereço cadastrado para as organizações.

Ao final, foi feito o teste de usabilidade pela ferramenta *Test Lab*, após isso foi feito *build* e criado uma *release* [9] do aplicativo para em seguida ser feito o deploy do aplicativo na ferramenta *App Distribution*, todas essas ferramentas pertencem ao *firebase*.

3.2 DESAFIOS

O maior desafio foi fazer contato com as organizações para tentar adquirir informações sobre adoção e, após a conclusão do desenvolvimento, a avaliação por parte deles do aplicativo. Isso impossibilitou o desenvolvimento da funcionalidade de avaliação comportamental por parte dos pets, para que tenhamos uma combinação melhor entre os adotantes e os pets. Então por esse

motivo, essa funcionalidade ficará para versões posteriores do aplicativo.

4. AVALIAÇÃO

A aplicação depois de finalizada, foi avaliada por 7 pessoas, como usuários adotantes (com idades entre 20 e 40 anos, moradores de Campina Grande-PB, de todos os níveis de instrução).

Foi feito um formulário de satisfação, onde foram feitos questionamentos. Foram eles:

1. Você ficou satisfeito com o aplicativo?
2. Você concorda que o aplicativo possui as funcionalidades essenciais?
3. Como você avaliaria o aplicativo quanto à...
 - 3.1. Usabilidade
 - 3.2. Performance
 - 3.3. Funcionalidades
 - 3.4. Design
4. Qual nota você daria para a utilidade (i.e., vai ajudar na conscientização para adoção e/ou facilitar a adoção de pets) do aplicativo?
5. Tem algo em que podemos melhorar no aplicativo? (opcional)
6. Tem alguma funcionalidade que você gostaria de ver no aplicativo? (opcional)
7. Nome e endereço de e-mail (opcional)

Nas perguntas de 1 a 3, com escalas Likert de 1 a 5, onde 1 significa "pouco satisfeito" e/ou "pouco útil", e 5 significa "muito satisfeito" e/ou "muito útil". Para a 4ª pergunta, foi colocado uma escala de 1 a 10, para avaliação da utilidade do aplicativo de um modo geral. As perguntas 5 e 6, que são opcionais, são perguntas para melhoria do aplicativo. E por fim, na 7ª pergunta, que também é opcional, o nome para possível identificação do usuário.

Para avaliação, foi enviado uma versão do aplicativo por meio da funcionalidade *App Distribution* do *Firebase*, ao qual pode ser enviado diretamente para o e-mail ou por um link compartilhável.

Com base nos resultados da pesquisa, vimos que o design da aplicação é o principal item a ser melhorado, visto que na 3ª pergunta de item 4, foi o menor avaliado e na 5ª pergunta foi respondida por 2 pessoas que o design seria um ponto a ser melhorado.

Como você avaliaria o aplicativo quanto à...

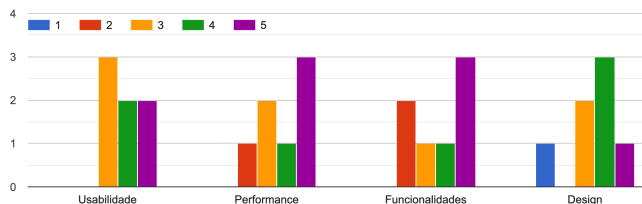


Figura 17 - Avaliação quanto ao perfil geral do aplicativo

Por outro lado, vimos que a aplicação é bem útil para o usuário adotante, de modo que todas as respostas à 1ª pergunta foram avaliadas como 4 e 5 da escala apresentada, além da nota de um modo geral ficou em média 9.

Você concorda que o aplicativo possui as funcionalidades essenciais?
7 respostas

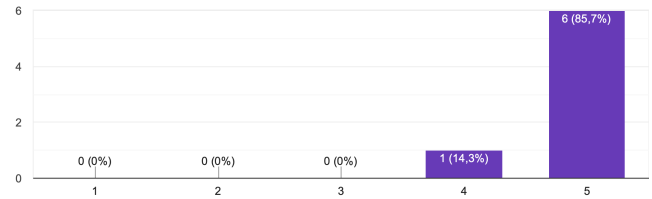


Figura 18 - Resultados das funcionalidades para os usuários

Qual nota você daria para o aplicativo?
7 respostas

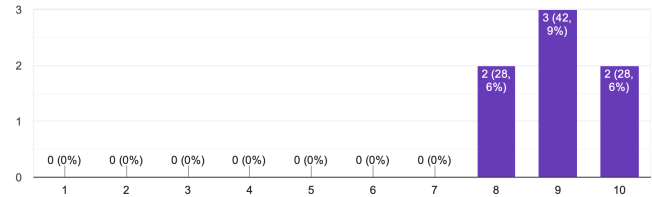


Figura 19 - Nota geral dada pelos usuários ao aplicativo

Outro ponto importante a ser comentado é que não foi possível ter uma avaliação por organizações do meio, pois nenhuma das mesmas responderam ao contato em tempo hábil para apresentar nesse trabalho. Porém, foram feitas avaliações com veterinários(as) em que ajudam nesse processo de adoção, ao qual representam essas organizações.

5. TRABALHOS FUTUROS

Para essa plataforma, podemos destacar algumas novas funcionalidades, para deixar ele ainda mais completo, como:

Fazer o build para dispositivos *iOS*;

Cadastro para o usuário final;

Criação da aplicação web;

Melhorias no design da aplicação;

Mostrar no mapa os próximos eventos na região;

Atualizar a cadastro de pets para adicionar mais dados;

Adicionar adsense para monetizar o app;

Fazer análises dos perfis comportamentais dos pets e dos usuários, para que combinem;

6. REFERÊNCIAS

- [1] MOL, Samylla. Mesmo sem transmitir o coronavírus, cães e gatos têm sido alvo de abandono. **Semad, 2020**. Disponível em: <http://www.meioambiente.mg.gov.br/noticias/4135-mesmo-s-em-transmitir-o-coronavirus-caes-e-gatos-tem-sido-alvo-de-abandono>. Acesso em: 23 abr. 2021 .
- [2] **React Native**. Disponível em: <https://reactnative.dev/docs/components-and-apis>. Acesso em: 27 de mai. de 2021.
- [3] PEDUZZI, Pedro. Adoção e abandono de animais domésticos aumentam durante a pandemia. **Agência Brasil, 2020**. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2020-10/adoca-o-e-abandono-de-animais-domesticos-aumentam-durante-pandemia>. Acesso em: 20 abr. 2021.
- [4] CARDOSO, Bruna. 9 em cada 10 brasileiros usam celular Android, diz relatório do Google. **TechTudo, 2020**. Disponível em: <https://www.techtudo.com.br/noticias/2020/09/9-em-cada-10-brasileiros-usam-celular-android-diz-relatorio-do-google.ghl>. Acesso em: 18 abr. 2021.
- [5] FILHO, Gerson Luis Ferreira Da Silva. Desenvolvimento de aplicativo para adoção de animais abandonados utilizando a linguagem de programação kotlin e programação reativa. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/8462/1/C_T_COENC_2017_1_4.pdf. Acesso em: 23 abr. 2021.
- [6] LOPES, Alice; CARACAS, Juliana; BRANCO, Luana Castelo; FIGUEIRA, Mariana. Projeto experimental. união protetora dos animais carentes - UPAC. Disponível em: http://www.repositorio.ufc.br/bitstream/riufc/26539/1/2013_tcc_avlopesjsleite.pdf. Acesso em: 23 abr. 2021.
- [7] MENISCK, Marcelo. React Native: Organizando o projeto. **Medium, 2017**. Disponível em: <https://medium.com/@menisck/react-native-organizando-o-projeto-20f141d587e4>. Acesso em: 19 abr. 2021.
- [8] FERNANDES, Diego. React do zero: componentização, propriedades e estado. **Rocketseat, 2017**. Disponível em: <https://blog.rocketseat.com.br/react-do-zero-componentizaca-o-propriedades-e-estado/>. Acesso em: 29 jan. 2021.
- [9] Kriss. How to Generate a React Native Release Build APK for Android. **Instamobile.io, 2020**. Disponível em: <https://www.instamobile.io/android-development/generate-react-native-release-build-android/>. Acesso em: 17 mar. 2021.
- [10] **Onesignal**. Disponível em: <https://documentation.onesignal.com/docs>. Acesso em: 25 abr. 2021.
- [11] **Firebase**. Disponível em: <https://firebase.google.com/docs>. Acesso em: 25 abr. 2021.
- [12] BROWN, Simon. **C4 Model, 2021**. The C4 model for visualising software architecture. Disponível em: <https://c4model.com>. Acesso em: 25 abr. 2021.
- [13] PAULA, Gilles B. de. Tudo sobre Metodologia Scrum: o que é e como essa ferramenta pode te ajudar a poupar tempo e gerir melhor seus projetos. **Treasy, 2016**. Disponível em: <https://www.treasy.com.br/blog/scrum/>. Acesso em: 25 abr. 2021.