

# Adote Amor: um aplicativo para divulgação gratuita de Organizações Não Governamentais em Prol da Proteção Animal

Hellen Susana Izolan\*  
Prof<sup>a</sup>. Dr<sup>a</sup>. Anubis Graciela Moraes Rossetto †

2020

## Resumo

Este trabalho tem como objetivo apresentar o desenvolvimento de uma aplicação web para divulgação de Organizações Não Governamentais em Prol da Proteção Animal, disponibilizando seu uso de forma gratuita. A aplicação foi desenvolvida utilizando Ionic Framework, seguindo os conceitos de PWA (Progressive Web App) e utilizando recursos da Plataforma Firebase. O projeto foi dividido em duas aplicações, sendo uma delas para a Área Administrativa, onde a ONG realiza seu cadastro e outra para a Aplicação Final, onde os dados cadastrados serão exibidos. Com o auxílio da Plataforma Firebase foi possível disponibilizar a opção de autenticação via e-mail e senha e também via Facebook. O resultado do projeto possibilita oferecer de forma gratuita uma aplicação web amigável, responsiva, customizável e que pode ser adicionada a área de trabalho dos usuários sem a necessidade de download em lojas de aplicativo, além de ser executável em qualquer dispositivo, propiciando a divulgação de não apenas uma, mas diversas ONGs.

**Palavras-chave:** PWA. Ionic. Firebase. ONGs. Adote Amor.

---

\* <[hellensusanaizolan@gmail.com](mailto:hellensusanaizolan@gmail.com)>

† <[anubisrossetto@gmail.com](mailto:anubisrossetto@gmail.com)>

# 1 Introdução

Segundo dados oficiais da Organização Mundial da Saúde de 2010, vivem no Brasil aproximadamente 30 milhões de animais abandonados, entre eles 10 milhões de gatos e 20 milhões de cães ([JUSBRASIL, 2013](#)). Em cidades de grande porte, para cada cinco habitantes há um cachorro, sendo que 10% destes estão abandonados. No interior, em cidades menores, a situação não é muito diferente, em muitos casos o número pode chegar a 1/4 da população humana. Buscar soluções para este problema é fundamental, mas é também um desafio a ser enfrentado.

Infelizmente o poder público carece de políticas que possam auxiliar no controle do abandono e no tratamento destes animais, fazendo assim com que não haja amparo financeiro e estrutural para que as Instituições não Governamentais prossigam com seus trabalhos. Por esse motivo e pelo grande número de novos casos todos os dias, a busca por meios atuais de conscientização da população e de divulgação dos animais disponíveis para adoção vem crescendo incessantemente.

Como hoje o poder da internet e seus meios, tais como WebSites, aplicativos e mídias sociais promovem ampla divulgação de qualquer assunto, este trabalho busca desenvolver um aplicativo web para divulgação gratuita de Organizações Não Governamentais em Prol da Proteção Animal. Além disso, durante pesquisas realizadas, foi detectado que nenhuma outra plataforma dentre as analisadas disponibiliza os serviços propostos pelo projeto, existem apenas sites e/ou aplicativos para divulgação de uma ONG específica, mas nenhum aplicativo web configurável para uso de diferentes ONGs.

Em relação a escolha da linguagem de programação e do método de desenvolvimento do aplicativo, para ([CSELLE, 2018](#)), todo e qualquer processo obrigatório, como download, instalação e cadastro/inscrição, que antecede o uso real de um aplicativo, cria oportunidades para que o usuário desista de usá-lo. O autor afirma que “Em um aplicativo para dispositivos móveis do consumidor, todas as etapas adicionadas ao seu fluxo de integração custarão cerca de 20% dos usuários” ([CSELLE, 2018](#)). Sendo assim, a busca por aplicativos práticos, ágeis, de fácil compreensão e execução e que exijam o menor uso de memória possível dos dispositivos é o ponto forte do mercado atual.

Neste contexto, ao desenvolver o projeto utilizando Ionic Framework juntamente com os conceitos de PWA (Progressive Web Apps), que serão descritos na seção 2 deste artigo, foi possível criar um aplicativo web totalmente responsivo, executável em qualquer dispositivo móvel ou desktop e que não requer instalação, o usuário pode simplesmente acessar e usar.

Buscando atender os requisitos acima descritos, a plataforma Adote Amor oferecerá uma Área Administrativa para cadastro dos dados da ONG bem como logotipo padrão da instituição, informações de localização, telefones para contato, contas bancárias para depósito, animais disponíveis para adoção e conteúdos diversos como projetos e campanhas que a ONG possa desejar divulgar, essa possibilidade fará com que diversas ONGs possam se cadastrar e fazer uso da plataforma. Após realizar o cadastro e preencher os dados da ONG, uma aplicação final irá exibir os dados específicos de cada instituição cadastrada, possibilitando que a ONG possa

utilizar o aplicativo para divulgar suas principais informações e projetos.

O artigo está dividido da seguinte forma: a seção 2 especifica as tecnologias utilizadas para codificação do aplicativo e armazenamento de dados, a seção 3 apresenta a relação de trabalhos correlatos analisados, a seção 4 descreve os requisitos de projeto e como sua codificação foi realizada, além de apresentar quais funcionalidades foram implementadas e algumas telas do aplicativo. Por fim, tem-se as considerações finais e as sugestões para trabalhos futuros.

## 2 Tecnologias utilizadas

Nesta seção são descritas as tecnologias que foram utilizadas no desenvolvimento do aplicativo.

### 2.1 Ionic Framework e suas Tecnologias

Hoje, o desenvolvimento de aplicativos para dispositivos móveis é uma das áreas de mais interesse tanto por parte dos jovens programadores, quanto do mercado, que busca soluções práticas, leves e interativas para facilitar o dia a dia dos usuários. Segundo publicação da Educa Mais Brasil, inúmeros são os fatores que tornam o mercado de trabalho de desenvolvimento mobile atrativo e em constante crescimento, um destes é o fato dos brasileiros estarem em primeiro lugar quando o assunto é o uso de aplicativos para as atividades cotidianas, além de, em 2019, o ramo ter atingido a maior perspectiva de aumento salário do ano, chegando a 11,76% de aumento (SILVA, 2018).

A programação utilizando o Ionic framework é especificamente baseada na experiência do usuário com a interface do aplicativo, utilizando diversos componentes visuais para otimizar ao máximo essa interação. Segundo a documentação da página oficial do Framework, "Ionic Framework é um kit de ferramentas de IU (Interface do Utilizador) de código aberto para a construção de aplicativos móveis e de desktop de alta qualidade usando tecnologias da web - HTML, CSS e JavaScript - com integrações para estruturas populares como Angular , React e Vue."(IONIC, 2020c).

O desenvolvimento com Ionic Framework gera aplicações híbridas, ou seja, um único projeto de código-fonte capaz de gerar uma aplicação que pode ser exportada para diversas plataformas. Essa modalidade de programação tem se tornado cada vez mais comum no mercado por não exigir que os desenvolvedores tenham conhecimento específico para cada plataforma de software disponível. Além disso, todas as aplicações híbridas são executadas através de um recurso chamado webview, um tipo especial de browser que é executado quando a aplicação híbrida é requisitada pelo usuário. Apesar de não ficar explícita para o usuário, essa forma de execução possibilita que os usuários façam uso da ferramenta sem a necessidade de realizar a transferência do aplicativo, reduzindo assim o uso de memória do dispositivo.

Como mencionado pela documentação do Ionic, o framework tem integração oficial com o Angular, uma plataforma e framework que facilita a criação de aplicativos web e que tem seus códigos gravados usando TypeScript (ANGULAR, 2020b). O

TypeScript implementa funcionalidades através de um conjunto de bibliotecas que o desenvolvedor importa para seus projetos. É compilado para um código JavaScript (linguagem de script para páginas Web ([DEVELOPER, 2020](#))), executável em qualquer navegador, no Node.js ou em mecanismos JavaScript que suportem o ECMAScript 3 ou mais recente ([TYPESCRIPT, 2020](#)). O ECMAScript é definido como um padrão de linguagens de programação baseadas em Scripts, padronizada pela Ecma International na especificação ECMA-262 ([ECMA, 2020b](#)). A Ecma International (European Computer Manufacturers Association - Associação Europeia de Fabricantes de Computadores), é uma associação fundada em 1961 dedicada à padronização de sistemas de informação ([ECMA, 2020a](#)).

Ou seja, no desenvolvimento com Ionic é possível implementar aplicativos que rodam em várias plataformas, como iOS, Android, desktop, Web, e com o conceito de PWA, onde o resultado final são aplicativos construídos com base nos padrões da Web, HTML (Hypertext Markup Language - Linguagem de Marcação de Hipertexto), CSS (Cascading Style Sheets - Folha de Estilo em Cascatas) e JavaScript, com design simples, amigável, de fácil aprendizagem e capaz de oferecer experiências parecidas para usuários de diferentes plataformas.

## 2.2 Progressive Web Apps (PWA)

Segundo a documentação oficial do Google Developers, os Progressive Web Apps (Aplicativos da Web Progressivos), foram projetados para serem eficientes, confiáveis e instaláveis ([DEVELOPERS, 2020c](#)). Baseado nesses três principais pilares, oferecem uma experiência parecida com a de um aplicativo específico de plataforma, mas através de um aplicativo web, possibilitando o alcance de praticamente qualquer pessoa, em qualquer local, por serem desenvolvidos através de apenas um código fonte que pode ser executado em diferentes dispositivos ([DEVELOPERS, 2020c](#)).

Para que um PWA possa oferecer a melhor experiência possível, o Google Developers oferece uma lista de conferência com algumas recomendações a serem seguidas ([DEVELOPERS, 2020d](#)), são elas:

- Começa rápido, permanece rápido: o desempenho tem um papel muito importante para gerar sucesso em qualquer experiência online, por isso a PWA deve ser otimizada ao máximo;
- Funciona em qualquer navegador: os usuários podem usar qualquer navegador que escolherem para acessar seu PWA;
- Responde a qualquer tamanho de tela: o PWA precisa ser totalmente responsivo, permitindo que todo o conteúdo esteja disponível em qualquer tamanho de tela;
- Fornece uma página offline personalizada: quando offline, manter o usuário na PWA oferece uma experiência mais contínua do que voltar para a página offline padrão do navegador;
- É instalável: precisa ser instalável pois os usuários que instalam ou adicionam aplicativos a seus dispositivos tendem a se envolver mais com esses aplicativos;

- Oferece uma experiência offline: onde a conectividade não for estritamente necessária, seu aplicativo deve funcionar tanto offline quanto online;
- É totalmente acessível: todas as interações do usuário atendem aos requisitos de acessibilidade WCAG 2.0;
- Podem ser encontrados por meio de pesquisa: seu PWA pode ser facilmente descoberto por meio de pesquisas na web;
- Funciona com qualquer tipo de entrada: Seu PWA pode ser usado da mesma forma com um mouse, teclado, caneta ou toque;
- Fornece contexto para solicitações de permissão: ao pedir permissões, forneça contexto simples e pergunte apenas quando for necessário;
- Segue boas práticas para códigos saudáveis: manter sua base de código íntegra torna mais fácil atingir seus objetivos e fornecer novos recursos.

Assim, é possível afirmar a importância e a relevância do desenvolvimento desse projeto utilizando os conceitos de PWA, já que aplicativos progressivos da web podem obter um melhor desempenho utilizando as características de um aplicativo móvel, mas sem implicar as dificuldades normalmente encontradas ao desenvolver e manter aplicativos específicos para cada plataforma.

### 2.2.1 Service Worker

O Service Worker é um arquivo JavaScript que é executado no navegador, independente de seu aplicativo, em resposta a eventos como solicitações de rede, notificações push, mudanças de conectividade e mais, possibilitando através de suas configurações funcionalidade offline, atualização de conteúdo em segundo plano, cache de conteúdo e outros (IONIC, 2020b).

O Service Worker pode também ser descrito como uma espécie de proxy ou middleware para as solicitações feitas a partir dos eventos, através dele é possível ouvir estes eventos e verificar os dados em cache retornando uma resposta imediatamente ou permitindo que a solicitação continue para o servidor remoto (IONIC, 2020b).

### 2.2.2 Instalação e App Manifest

Até pouco tempo os aplicativos web não podiam ser instalados como um aplicativo na tela inicial, o usuário até podia fixar um site em sua tela inicial, mas a experiência era incompleta e não como a esperada em aplicativos nativos. Contudo, recentemente o Chrome adicionou no Android suporte para a instalação de aplicativos da web na tela inicial com um banner de instalação nativo, assim como os banners de aplicativos nativos que os usuários estão acostumados. Para informar ao Chrome que um aplicativo web pode ser instalado como um aplicativo, deve-se escrever um arquivo manifest.json e incluir um link para ele na página HTML principal. Infelizmente, hoje o iOS não tem nenhum recurso adicional além de fixar um atalho a tela inicial, então a experiência não será tão interessante como em sistemas Android (IONIC, 2020b).

## 2.3 Plataforma Firebase

O Firebase é uma plataforma de desenvolvimento para dispositivos móveis que conta com sistema de autenticação, base de dados, envio de mensagens e relatórios de erros, além de possibilitar e facilitar o compartilhamento de informações entre aplicativos integrados ao Firebase. Foi criado sobre a infraestrutura do Google e oferece escalonamento automático, até mesmo para grandes aplicativos. (DEVELOPERS, 2020a).

No período de escrita deste artigo o Firebase dispõe de dezoito diferentes produtos que estão divididos em três grupos, o primeiro se refere aos produtos para criação de aplicativos, o segundo grupo envolve opções para melhorar a qualidade dos aplicativos já implantados e o terceiro grupo dispõe de produtos para expandir os negócios relativos ao aplicativo, tratando de engajamento de usuários, análises de acessos entre outros (DEVELOPERS, 2020b).

Neste projeto os produtos utilizados foram o Authentication para autenticação dos usuários do aplicativo, o Cloud Firestore, banco de dados não relacional baseado em coleções e documentos, utilizado para armazenamento dos dados, o Cloud Storage para armazenamento de mídias (imagens), e o Hosting para hospedagem do projeto.

## 3 Trabalhos Correlatos

Buscando alcançar os objetivos do projeto, foram realizadas pesquisas no quesito aplicativo, analisando projetos já desenvolvidos na área para obter-se, assim, dados relevantes quanto à organização das páginas e definição das funções principais.

Tabela 1 – Aplicativos Analisados

Nome da aplicação	Onde encontrar	Recursos
Adô - doação e adoção	App PlayStore	Relação de pets para adoção
AdotaFácil	App PlayStore	Relação de pets para adoção
Meaudota	App PlayStore	Relação de pets para adoção
AdotaAqui	App PlayStore	Relação de pets para adoção
SOS Pet	App PlayStore	Relação de pets para adoção
PetFinder	App PlayStore	Relação de pets perdidos
Site Arca Brasil	arcabrasil.org.br	Dados Institucionais, campanhas, área para doações, loja virtual
Site SUIPA	suipa.org.br	Dados Institucionais, campanhas, área para doações, loja virtual
Blog Amo Cachorros	amocachorros.com.br	Dicas, lista de ONGs existentes
Site Proanima	proanima.org.br	Dicas, informativos, campanhas
Site Apoie-se	apoia.se	Cmpanhas sociais diversas
Site Atados	atados.com.br	Campanhas sociais diversas

Fonte: Autor, 2019

Segundo a pesquisa realizada e exemplificada na Tabela 1, os aplicativos disponibilizados na PlayStore apenas listam animais para adoção, com exceção do SOS Pet, que lista animais perdidos a procura de seus donos. Já os sites da Arca Brasil e da SUIPA são plataformas que disponibilizam conteúdos institucionais, campanhas, além de áreas específicas para doações e venda de produtos através de uma loja virtual. O Blog Amo Cachorros apenas lista ONGs já existentes, enquanto o site Proanima disponibiliza dicas, informativos e campanhas específicas dessa ONG. Os sites Apoie-se e Atados são plataformas específicas para anúncio de ONG's e campanhas das mais diferentes áreas, não exclusiva e especificamente para a causa animal. Sendo assim, dentre as aplicações analisadas nenhuma outra plataforma que disponibilize os serviços propostos pelo projeto Adote Amor foi encontrada, estão disponíveis apenas sites e/ou aplicativos para divulgação de uma ONG específica, mas nenhum aplicativo web configurável para uso de diferentes ONGs.

## 4 Aplicativo Adote Amor

Esta seção tem como objetivo descrever os requisitos do projeto e quais as funcionalidades implementadas durante seu desenvolvimento, relatar como a persistência de dados foi realizada, bem como apresentar telas exemplificando parte de sua codificação e organização.

### 4.1 Requisitos do Adote Amor

A ideia principal deste projeto firma-se em desenvolver um aplicativo web seguindo os conceitos de PWA que seja capaz de auxiliar gratuitamente na divulgação de não somente uma, mas várias Organizações Não Governamentais em Prol da Proteção Animal. Tendo isso em mente, o projeto foi organizado em duas aplicações diferentes, uma para Área Administrativa e outra para Aplicação Final que irá exibir os dados de cada ONG.

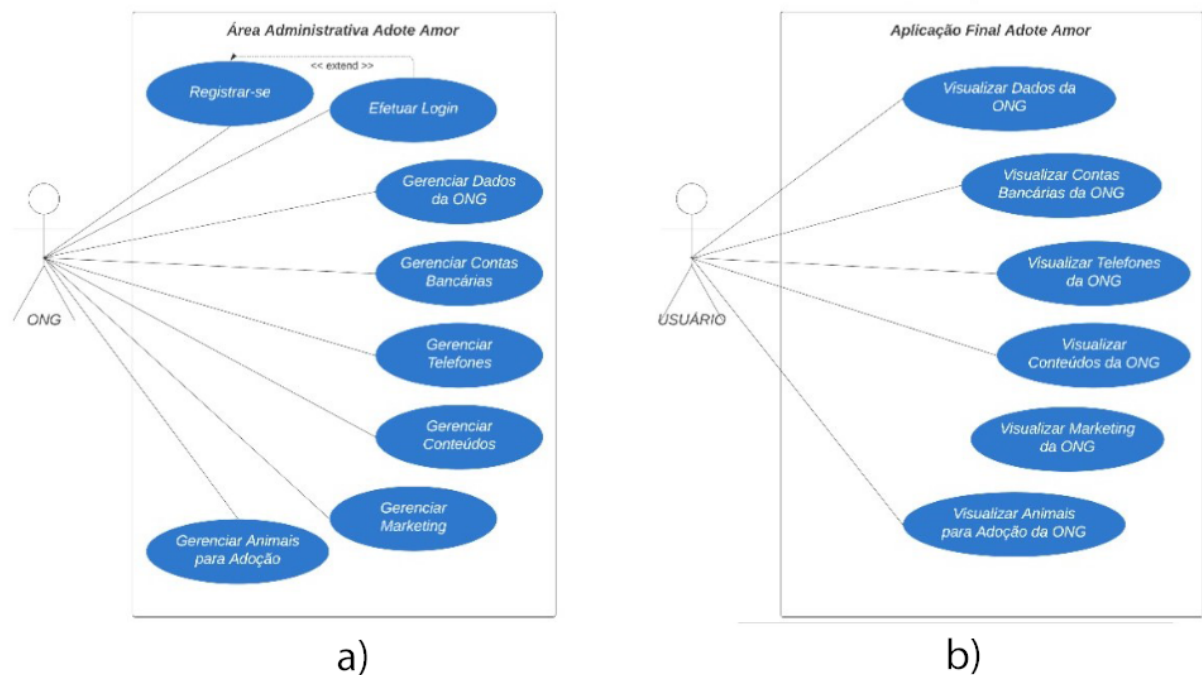
A Área Administrativa gerencia o cadastro de cada ONG, ou seja, cada ONG pode realizar seu cadastro de forma independente, inserindo seus principais dados, bem como logotipo padrão da instituição, informações de localização, telefones para contato, contas bancárias para depósito, animais disponíveis para adoção e conteúdos diversos como projetos e campanhas que a ONG possa desejar divulgar. Já a Aplicação Final será responsável por exibir estes dados que foram anteriormente cadastrados na Área Administrativa.

Para que as ONGs não precisem adquirir domínios próprios e mesmo assim possam dispor de uma página exclusiva para sua ONG, a Aplicação Final exibirá os dados de cada ONG em seu próprio domínio através de um parâmetro passado via Uniform Resource Locator (URL), conforme a URL acessada os dados da ONG correspondente ao parâmetro desta url serão exibidos.

## 4.2 Diagrama de Casos de Uso

Os diagramas de casos de uso buscam auxiliar através de propriedades gráficas na representação de como as funcionalidades do sistema irão interagir entre si, além das interações dos atores (sujeitos que estarão interagindo com o sistema) para com a aplicação. Assim sendo, os casos de uso abaixo abordam as ações que podem ser desempenhadas pelos usuários da Área Administrativa e da Aplicação Final do projeto.

Figura 1 – Casos de Uso do Aplicativo Adote Amor



Fonte: Autor, 2020

A Figura 1(a) exibe as possíveis interações da ONG para com a área administrativa. Como pode ser observado, a ONG poderá realizar seu cadastro, consequentemente logar na aplicação e posteriormente gerenciar o cadastro de seus principais dados, além de telefones, contas bancárias, marketing, conteúdos e animais para adoção. Esse gerenciamento inclui visualização, inserção, edição, e exclusão de dados.

Já na Aplicação final, que é a área pública do projeto, o usuário final poderá apenas visualizar os dados exibidos, mas com a possibilidade de navegar pela página de forma livre, sem restrições de acesso, a Figura 1(b) apresenta os casos de uso da Aplicação Final.

## 4.3 Autenticação de Usuários

No desenvolvimento da aplicação, o gerenciamento de usuários foi realizado através do módulo Authentication do Firebase. Com o auxílio do componente



AngularFireAuth da Biblioteca AngularFire2, que é integração oficial do Firebase com o Angular (ANGULAR, 2020a), foi possível realizar o cadastro e acesso de usuários via e-mail e senha e também via Facebook.

Figura 2 – Configuração do aplicativo no painel do Facebook Developers

**Configurações de OAuth do cliente**

**Sim** **Login no OAuth do cliente**  
Ativa o fluxo do token de cliente do OAuth padrão. Proteja seu aplicativo e evite o abuso usando as opções abaixo para impedir alterações nos URIs de redirecionamento de token permitidos. Desative globalmente se não for usado. [?]

**Sim** **Login do OAuth na Web**  
Ative o login no OAuth do cliente baseado na Web. [?]

**Sim** **Exigir HTTPS**  
Exija o uso de HTTPS para URIs de redirecionamento e o SDK do JavaScript. Altamente recomendado. [?]

**Não** **Forçar reautenticação do OAuth na Web**  
Quando ativado, solicita que as pessoas insiram sua senha do Facebook para fazer o login na Web. [?]

**Não** **Login do OAuth no navegador incorporado**  
Ative os URIs de redirecionamento do webView para login no OAuth do cliente. [?]

**Sim** **Usar modo estrito para URIs de redirecionamento**  
Permita somente redirecionamentos que usam o SDK do Facebook ou que correspondem exatamente aos URIs de redirecionamento do OAuth válidos. Altamente recomendável. [?]

**URIs de redirecionamento do OAuth válidos**

**Não** **Login de dispositivos**  
Ativa o fluxo de login de cliente do OAuth para dispositivos como uma smart TV [?]

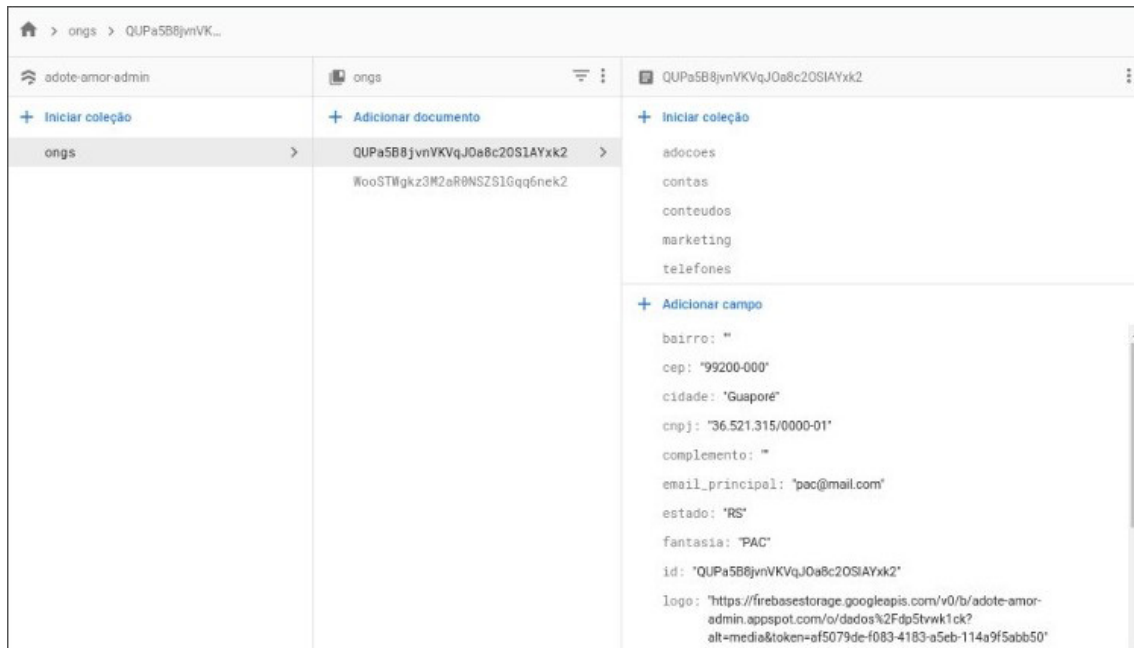
Fonte: Autor, 2020

Através do console do Firebase a autenticação via E-mail e Senha e também via Facebook foi habilitada. Após esse processo, para que a autenticação com Facebook fosse possível, se fez necessário registrar a aplicação através do Facebook Developers e também configurar lá quais seriam as urls padrão do aplicativo, a Figura 2 apresenta as configurações do Aplicativo Adote Amor no Facebook Developers.

#### 4.4 Persistência de Dados

A persistência de dados do projeto foi realizada através do Cloud Firestore, e como a Figura 3 apresenta a estrutura utilizada é composta por uma coleção chamada ONG que armazena os principais dados de cada instituição, atrelada a ela existem subcoleções chamadas adocoos, contas, conteudos, marketing e telefones que armazenam os demais dados das ONGs.

Figura 3 – Estrutura do Cloud Firestore



Fonte: Autor, 2020

Em relação as regras de acesso ao Cloud Firestore, a Figura 4 mostra que o projeto foi configurado para aceitar leitura de dados sem autenticação, mas escrita apenas quando um usuário devidamente cadastrado no Authentication do projeto for verificado.

Figura 4 – Regras de acesso ao Cloud Firestore

```
1 rules_version = '2';
2 service cloud.firestore {
3   match /databases/{database}/documents {
4     match /{document=**} {
5       allow read;
6       allow write: if request.auth != null;
7     }
8   }
9 }
```

Fonte: Autor, 2020

## 4.5 Área Administrativa

A Área Administrativa é responsável por gerenciar o cadastro de cada ONG e todos os seus dados, assim sendo sua construção foi realizada levando em consideração a necessidade de listar dados e também de oferecer telas para criação e edição de registros. Para isso, utilizando elementos Ionic que são tags HTML já estilizadas disponibilizadas pelo próprio Ionic em sua documentação oficial (IONIC, 2020a), foi

possível desenvolver um projeto com código organizado e exibição simples, amigável e responsiva. Nas Figuras 5 e 6 abaixo é possível observar o uso de algumas dessas tags Ionic.

Figura 5 – Componente para listagem de telefones utilizando tags Ionic

```
telefone-item.component.html ×
src > app > admin > components > telefone-item > < telefone-item.component.html > ...
1 <ion-item-sliding #sliding>
2   <ion-item (click)="update.emit(telefone)">
3     {{ telefone.titulo + " - " + telefone.numero }}
4   </ion-item>
5   <ion-item-options side="start">
6     <ion-item-option color="danger" (click)="delete.emit(telefone); sliding.close()">
7       <ion-icon slot="icon-only" name="trash"></ion-icon>
8     </ion-item-option>
9
10    <ion-item-option (click)="update.emit(telefone); sliding.close()">
11      <ion-icon slot="icon-only" name="create"></ion-icon>
12    </ion-item-option>
13  </ion-item-options>
14 </ion-item-sliding>
```

Fonte: Autor, 2020

A Figura 5 apresenta um componente codificado para exibir a listagem de telefones cadastrados que foi construído com uso do elemento Ionic 'ion-item-sliding'. Este elemento lista vários elementos filhos ditos 'ion-item' que em conjunto com o elemento 'ion-item-options' podem conter botões. Ao associar estes elementos, cada item se torna deslizante e assim, ao ser arrastado, dois botões são exibidos, um para edição e outro para exclusão do item.

Figura 6 – Chamada ao componente para listagem de telefones

```
telefones.page.html ×
src > app > admin > pages > telefones > < telefones.page.html > ion-footer
16 <ion-list *ngIf="(telefones$ | async) as telefones">
17   <ng-container *ngIf="telefones.length > 0; else noTelefones">
18     <app-telefone-item
19       *ngFor="let telefone of telefones"
20       [telefone]="telefone"
21       (update)="onUpdate($event)"
22       (delete)="onDelete($event)">
23   </app-telefone-item>
24 </ng-container>
25 <ng-template #noTelefones>
26   <ion-item lines="none">
27     Nenhum telefone cadastrado até o momento...
28   </ion-item>
29 </ng-template>
30 </ion-list>
```

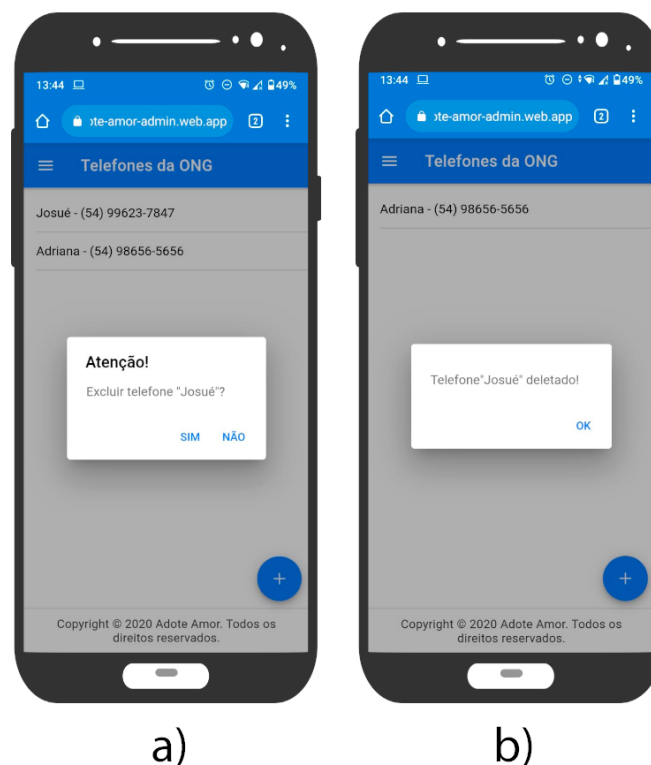
Fonte: Autor, 2020

Ja na Figura 6 é possível visualizar a chamada para o componente de listagem de telefones na página HTML principal, realizada através do elemento 'ion-list'. Como é possível observar, separando a codificação do componente de listagem da página que realiza a chamada para este, obtém-se um código mais limpo e organizado, além de possibilitar a reutilização do componente em outros locais do projeto. Vale ressaltar que todas as listagens da Área Administrativa seguem este padrão.

Em relação ao acesso dos dados no Cloud Firestore, através da biblioteca AngularFire2 (ANGULAR, 2020a) que dispõe de métodos para comunicação entre a aplicação e o Firestore, foi possível realizar a inserção, edição e exclusão de dados. Todas as consultas realizadas ao Firestore foram feitas por meio de Services, grupo de arquivos que foram configurados de modo a serem reutilizados em qualquer parte do projeto.

Outra funcionalidade que ficou concentrada em um Service é a possibilidade de exibir mensagens de alerta, erro e sucesso.

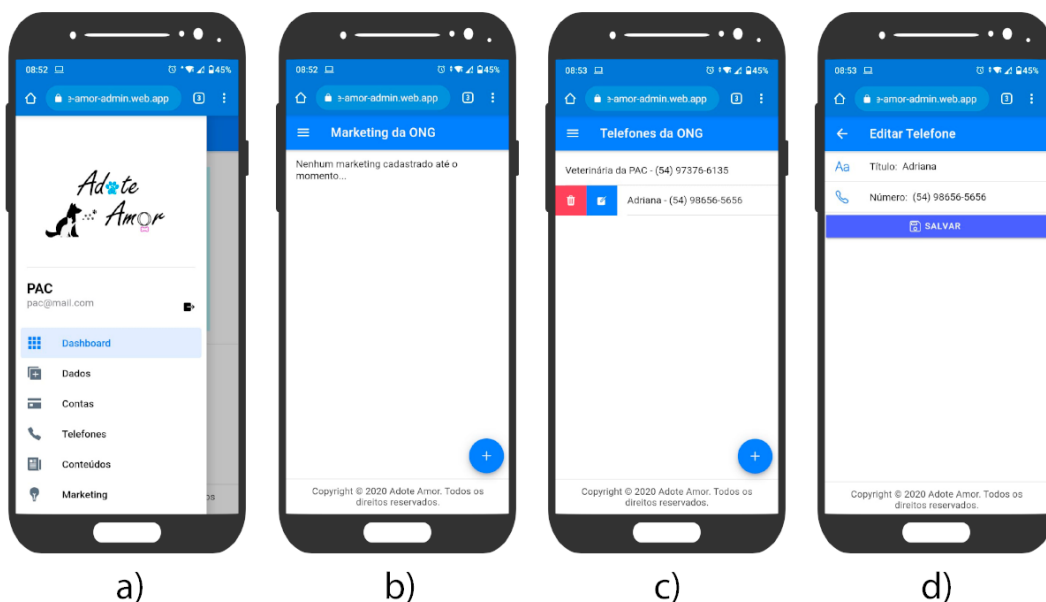
Figura 7 – Telas exibindo modais para exclusão de registro



Fonte: Autor, 2020

Todas as páginas de listagem da Área Administrativa contam com a exibição de modais solicitando confirmação para exclusão de dados e, posteriormente, retornam mensagens correspondentes ao resultado da ação. Nas Figuras 7(a) e 7(b), respectivamente, é possível visualizar a solicitação de exclusão de um telefone, a exibição do modal solicitando a confirmação e em seguida a mensagem de sucesso ao excluir o registro.

Figura 8 – Principais Telas da Área Administrativa



Fonte: Autor, 2020

Em relação ao layout das demais telas da Área Administrativa, a Figura 8(a) apresenta a exibição do menu configurado para permitir a navegação entre as páginas, a Figura 8(b) e 8(c) exibem, respectivamente, uma tela de listagem sem nenhum dado a ser exibido e outra com registros já devidamente cadastrados, por fim a Figura 8(d) apresenta a organização de uma página para edição de registro.

#### 4.5.1 Estrutura do Cloud Storage

Para armazenamento das imagens cadastradas na Área Administrativa e posteriormente exibidas na Aplicação Final, o Cloud Storage foi utilizado. A Figura 9 apresenta sua estrutura que é composta por três pastas para armazenamento de mídias, uma para cada CRUD que possibilita envio de imagens, são elas:

- *adocoes*: armazena imagens dos animais que podem ser cadastrados para adoção;
- *conteudos*: armazena as imagens que podem ser cadastradas no CRUD de conteúdos;
- *dados*: armazena o logotipo da ONG;

Figura 9 – Estrutura do Cloud Storage

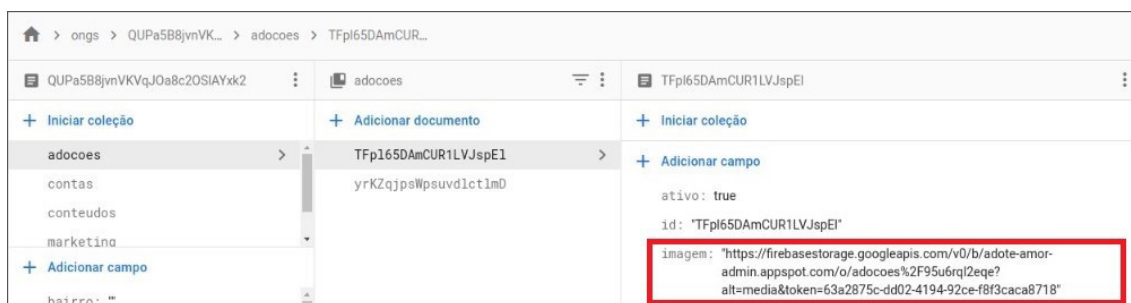


Name	Tamanho	Tipo	Última modificação
adocoes/	-	Pasta	-
conteudos/	-	Pasta	-
dados/	-	Pasta	-

Fonte: Autor, 2020

Em cada um dos documentos do Firestore aos quais estas imagens são pertencentes, seu caminho de acesso no Cloud Storage é armazenado em um campo do tipo string, sendo assim possível recuperar e exibir a imagem posteriormente. A Figura 10 abaixo destaca um dos documentos da subcoleção "adoções" com destaque para o campo "imagem" que armazena o caminho da imagem do animal para adoção em questão.

Figura 10 – Sub-coleção 'adocoes' do Cloud Firestore



Nome	Conteúdo
adocoes	<pre>TFp165DAmCUR1LVJspE1 yrKZqjpsWpsuvd1ctImD ativo: true id: 'TFp165DAmCUR1LVJspE1' imagem: "https://firebasestorage.googleapis.com/v0/b/adote-amor-admin.appspot.com/o/adocoes%2F95u6rq12eqe?alt=media&amp;token=63a2875c-dd02-4194-92ce-f8f3caca8718"</pre>

Fonte: Autor, 2020

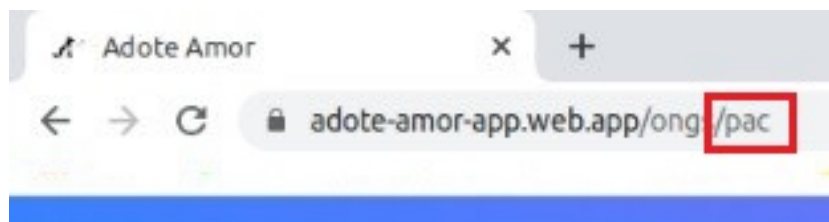
Em relação as regras de acesso ao Cloud Storage, assim como no caso do Cloud Firestore, o projeto foi configurado para aceitar leitura de dados sem autenticação, mas escrita apenas quando um usuário devidamente cadastrado no Authentication do projeto for verificado.

## 4.6 Aplicação Final

Visando exibir os dados cadastrados na Área Administrativa em uma Aplicação Final diferente para cada ONG, um novo projeto IONIC foi criado.

Neste projeto o parâmetro "url\_site" que pode ser configurado pela ONG na Área Administrativa, é passado junto a URL, assim sendo o projeto realiza uma consulta aos dados do Cloud Firestore com base neste parâmetro e exibe os dados da respectiva ONG, a Figura 11 exibe a barra de endereço do navegador acessando a Aplicação Final de uma das ONGs através do parâmetro enviado junto a URL.

Figura 11 – Barra de endereço do navegador acessando a Aplicação Final de uma das ONGs através do parâmetro enviado junto a URL



Fonte: Autor, 2020

Assim como na área administrativa, na aplicação final também foi possível chegar a um layout totalmente responsivo e amigável, fazendo uso dos elementos Ionic já estilizados que são fornecidos pelo Framework (IONIC, 2020a). Um elemento bastante interessante e que foi utilizado na codificação da aplicação final é o "ion-slides", apresentado na Figura 12. O elemento foi utilizado para exibir um slide contendo os animais para adoção cadastrados para a ONG. Como é possível observar, com poucas linhas de código foi possível realizar esta exibição, utilizando o parâmetro "pager=true" a paginação do slide já é automaticamente inserida, além do slider se adaptar perfeitamente a telas de diferentes tamanhos sem a necessidade de outras configurações.

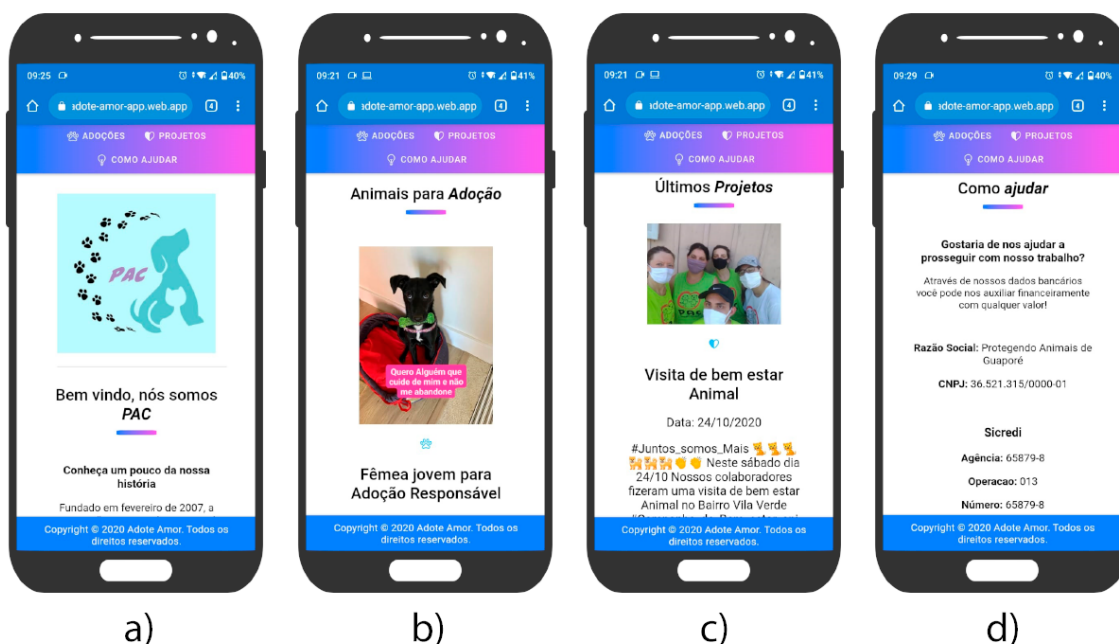
Figura 12 – Tela do arquivo html da Aplicação Final utilizando a tag Ionic "ion-slides" para exibição dos animais para adoção

```
ong.page.html X
src > app > ongs > ongs.page.html > ion-header.ion-text-center > ion-toolbar.header
76 <ion-col size="12" padding *ngIf="(adocoes$ | async) as adocoes">
77   <ion-slides pager="true" [options]="slideOpts">
78     <ion-slide *ngFor="let adocao of adocoes">
79       <ion-col *ngIf="adocao.imagem">
80         <ion-img class="slide-img" [src]="adocao.imagem"></ion-img>
81         <ion-icon name="paw-outline" color="secondary"></ion-icon>
82         <h2>{{ adocao.nome }}</h2>
83         <p>{{ adocao.sobre }}</p>
84       </ion-col>
85     </ion-slide>
86   </ion-slides>
87 </ion-col>
```

Fonte: Autor, 2020

Para este projeto um layout OnePage foi utilizado, onde a mesma página exibe todas as informações necessárias de forma ordenada (ZANETTI, 2018). Na Figura 13 é possível visualizar a exibição dos principais dados da ONG na Aplicação Final, iniciando pelo texto sobre a empresa, Figura 13(a), seguido dos animais para adoção, Figura 13(b), conteúdos postados pela ONG, Figura 13(c), contas para depósito, Figura 15(d) e no rodapé da página todas as informações de contato.

Figura 13 – Aplicação Final desenvolvida com layout OnePage

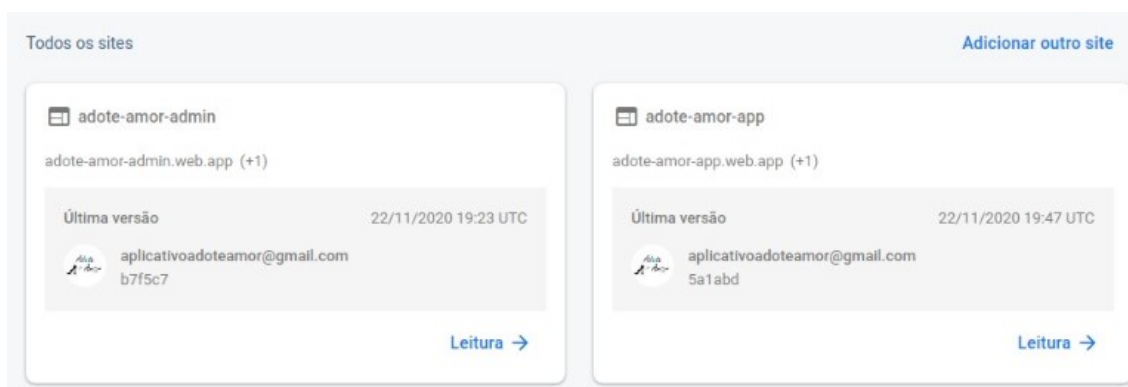


Fonte: Autor, 2020

## 4.7 Hospedagem no Firebase Hosting

Para realizar a hospedagem das aplicações o Firebase Hosting foi utilizado. Todos os recursos do Firebase que foram utilizados estão concentrados em apenas um projeto do Firebase, contudo, o Hosting conta com duas instâncias diferentes, assim, conforme a Figura 14 apresenta, o deploy da Área Administrativa do projeto e da Aplicação Final são feitos em separado, sendo acessados por duas URLS diferentes.

Figura 14 – Configuração do Firebase Hosting



Fonte: Autor, 2020



Devido as regras anteriormente criadas no Cloud Firestore e no Cloud Storage, através da Área Administrativa, quando devidamente logado, é possível que o usuário faça manutenção dos CRUDs tendo acesso a inserção, edição e exclusão. Já na Aplicação Final de cada ONG, somente leituras de dados são permitidas e realizadas, garantindo assim a segurança dos dados do projeto.

## Considerações finais

Conforme apresentado anteriormente, a carência de políticas públicas eficientes que possam auxiliar no controle e tratamento de animais abandonados, junto ao crescente número de novos casos todos os dias, faz com que as Organizações Não Governamentais em Prol da Proteção Animal estejam sempre em busca de novos meios para divulgação de seus projetos e arrecadação de fundos, para que possam assim prosseguir com seus trabalhos. Uma vez que estas ONGs já tem inúmeros custos com o tratamento dos animais que mantém, o objetivo principal do Aplicativo Adote Amor foi propiciar uma aplicação totalmente gratuita capaz de auxiliá-las na busca por recursos.

Buscando atingir este objetivo e seguindo os conceitos de PWA foi possível desenvolver uma aplicação que pode ser acessada de forma simples e rápida em qualquer dispositivo que disponha de navegador, sem a necessidade de download em lojas de aplicativos, facilitando seu uso e alcançando assim mais usuários, já que praticamente qualquer dispositivo poderá fazer uso dela. Além disso, a Área Administrativa do projeto é bastante amigável e de fácil uso, todas as informações da ONG podem ser facilmente cadastradas, sem a necessidade de grandes conhecimentos de informática ou então auxílio de profissionais da área de desenvolvimento web ou afins para isso. O Adote Amor oferece para as ONGs que o utilizam uma aplicação flexível e de fácil configuração, onde a ONG não precisa adquirir domínio próprio, hospedar a aplicação em um servidor ou ter qualquer outra preocupação relativa a manter a plataforma em funcionamento, a ONG simplesmente realiza seu cadastro e já tem disponível a Aplicação Final apresentando seus dados.

Com isso, considera-se que o projeto foi concluído com êxito, uma vez que o objetivo central de oferecer uma plataforma totalmente gratuita que pudesse auxiliar não somente uma, mas várias ONGs em Prol da Proteção Animal foi alcançado.

Tem-se como sugestão para trabalhos futuros a possibilidade de personalizar outros detalhes em relação a estilização da Aplicação Final, como o uso de cores.

# Adote Amor: an application for free dissemination of Non-Governmental Organizations for Animal Protection

Autor

2020

## Abstract

This work aims to present the development of a web application for the dissemination of Non-Governmental Organizations in favor of Animal Protection, making its use free of charge. The application was developed using Ionic Framework, following the concepts of PWA (Progressive Web App) and using resources from the Firebase Platform. The project was divided into two applications, one for the Administrative Area, where the NGO performs its registration and the other for the Final Application, where the registered data will be displayed. With the help of the Firebase Platform, it was possible to make the authentication option available via email and password and also via Facebook. The result of the project makes it possible to offer a free, friendly, responsive, customizable web application that can be added to the users' desktop without the need to download it from app stores, in addition to being executable on any device, allowing the dissemination of not just one, but several NGOs.

**Keywords:** PWA. Ionic. Firebase. ONGs. Adote Amor.

## Referências

ANGULAR. *Angular Fire*. 2020. Disponível em: <<https://github.com/angular/angularfire>>. Acesso em: 28 ago 2020. Citado 2 vezes nas páginas 9 e 12.

ANGULAR. *Visão geral da arquitetura*. 2020. Disponível em: <<https://angular.io/guide/architecture>>. Acesso em: 15 mai 2020. Citado na página 3.

CSELLE, G. *Every Step Costs You 20% of Users*. 2018. Disponível em: <<https://medium.com/gabor/every-step-costs-you-20-of-users-b613a804c329>>. Acesso em: 22 out 2020. Citado na página 2.

DEVELOPER, M. *JavaScript*. 2020. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 06 jun 2020. Citado na página 4.

DEVELOPERS, G. *Firebase*. 2020. Disponível em: <<https://firebase.google.com/>>. Acesso em: 06 jun 2020. Citado na página 6.

DEVELOPERS, G. *Firebase Products*. 2020. Disponível em: <<https://firebase.google.com/products>>. Acesso em: 06 jun 2020. Citado na página 6.

DEVELOPERS, G. *What are Progressive Web Apps?* 2020. Disponível em: <<https://web.dev/what-are-pwas/>>. Acesso em: 20 nov 2020. Citado na página 4.

DEVELOPERS, G. *What makes a good Progressive Web App?* 2020. Disponível em: <<https://web.dev/pwa-checklist/>>. Acesso em: 20 nov 2020. Citado na página 4.

ECMA. 2020. Disponível em: <<http://www.ecma-international.org/>>. Acesso em: 06 jun 2020. Citado na página 4.

ECMA. *ECMAScript*. 2020. Disponível em: <<http://www.ecma-international.org/memento/tc39.htm>>. Acesso em: 06 jun 2020. Citado na página 4.

IONIC. *API Index*. 2020. Disponível em: <<https://ionicframework.com/docs/api/>>. Acesso em: 20 nov 2020. Citado 2 vezes nas páginas 10 e 15.

IONIC. *What are Progressive Web Apps?* 2020. Disponível em: <<https://blog.ionicframework.com/what-is-a-progressive-web-app/>>. Acesso em: 06 mai 2020. Citado na página 5.

IONIC. *What is Ionic Framework?* 2020. Disponível em: <<https://ionicframework.com/docs>>. Acesso em: 06 mai 2020. Citado na página 3.

JUSBRAZIL. *Brasil tem 30 milhões de animais abandonados*. 2013. Disponível em: <<https://anda.jusbrasil.com.br/noticias/100681698/brasil-tem-30-milhoes-de-animais-abandonados>>. Acesso em: 25 mar 2019. Citado na página 2.

SILVA, G. *Desenvolvedor Mobile: como é a carreira de quem cria os aplicativos?* 2018. Disponível em: <<https://www.educamaisbrasil.com.br/educacao/carreira/desenvolvedor-mobile-como-e-a-carreira-de-quem-cria-os-aplicativos>>. Acesso em: 15 dez 2020. Citado na página 3.

TYPESCRIPT. 2020. Disponível em: <<https://www.typescriptlang.org/index.html>>. Acesso em: 06 jun 2020. Citado na página 4.

ZANETTI, V. *O que é site One Page?* 2018. Disponível em: <<https://webrain.com.br/blog/site-one-page/>>. Acesso em: 20 nov 2020. Citado na página 15.