



**CENTRO UNIVERSITÁRIO CHRISTUS**  
**SISTEMAS DE INFORMAÇÃO**

**IGOR COSTA PORDEUS**

**DESENVOLVIMENTO DE UM APLICATIVO PARA ADOÇÃO DE PETS**  
**UTILIZANDO FLUTTER E FIREBASE**

**FORTALEZA**

**2021**

IGOR COSTA PORDEUS

DESENVOLVIMENTO DE UM APLICATIVO PARA ADOÇÃO DE PETS UTILIZANDO  
FLUTTER E FIREBASE

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Sistemas de Informação do Centro Universitário Christus, como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof. Ms. Tiago Guimarães Sombra

FORTALEZA

2021

IGOR COSTA PORDEUS

DESENVOLVIMENTO DE UM APLICATIVO PARA ADOÇÃO DE PETS UTILIZANDO  
FLUTTER E FIREBASE

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Sistemas de Informação do Centro Universitário Christus, como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

Aprovada em:

BANCA EXAMINADORA

---

Prof. Ms. Tiago Guimarães Sombra (Orientador)  
Centro Universitário Christus (Unichristus)

---

Prof. Ms. Maurício Moreira Neto  
Centro Universitário Christus (Unichristus)

---

Prof. Ms. Euristenho Queiroz de Oliveira Júnior  
Centro Universitário Christus (Unichristus)

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais, pela educação que me proporcionaram desde o início da minha vida, pelo amor, incentivo e apoio incondicional.

Aos meus irmãos e tios que me ajudaram e motivaram a continuar essa caminhada.

A todos os professores que fizeram parte da minha vida acadêmica.

E a todos que fizeram parte de alguma forma da minha formação, obrigado a todos.

“A persistência é o caminho do êxito.”

(Charles Chaplin)

## RESUMO

O número de pessoas interessadas em adotar um animal de estimação no Brasil cresce a cada ano, como o número de animais abandonados também cresce, apesar de existir algumas leis que protegem os animais de estimação. Muitos desses animais são abandonados pois os tutores não possuem condições de manter o animal, por doenças, mudança de lar, questões financeiras, dentre outras situações. Fazendo com que seja necessário a criação de um aplicativo para dispositivos móveis para ajudar as pessoas que querem doar a encontrar as pessoas interessadas em adotar, pois cada vez mais os usuários da internet estão utilizando os celulares com forma de acesso a internet em vez de computadores. O objetivo geral desse trabalho consiste exatamente em criar um aplicativo multi-plataforma que ajude pessoas a adotarem e doarem animais de estimação. Já os objetivos específicos são: criar o aplicativo, usar histórias de usuário afim de encontrar os requisitos, utilizar o *Flutter* e *Firebase* no desenvolvimento, disponibilizar uma primeira versão e ajudar os primeiros usuários a utilizar o aplicativo. Para fazer isso a metodologia que será utilizado é a pesquisa quanto a natureza a abordagem qualitativa, do tipo aplicada, de caráter exploratória e técnica de pesquisa bibliográfica. Afim de gerar conhecimento para desenvolver o aplicativo que utilizará o *framework Flutter* para o desenvolvimento do *front-end* e comunicação com a *api* do *Firebase* e as ferramentas *Storage*, *Firestore* e *Authentication* do *Firebase* onde esse oferece serviços como *back-end*. Depois de desenvolvido o aplicativo será exposto dificuldades encontradas como integração do *Firebase* com o aplicativo e funções do desenvolvimento. E por fim é apresentado as conclusões e melhorias que podem ser efetuadas em trabalhos futuros.

**Palavras-chave:** Adoção de pets. *Flutter*. *Firebase*. Multi-plataforma. Aplicativo *mobile*.

## ABSTRACT

The number of people interested in adopting a pet in Brazil grows every year, as the number of abandoned animals also grows, despite some laws that protect pets. Many of these animals are abandoned because the guardians are unable to keep the animal, due to illnesses, change of home, financial issues, among other situations. Making it necessary to create an application for mobile devices to help people who want to donate to find people interested in adopting, as more and more internet users are using cell phones to access the internet instead of computers . The general objective of this work is exactly to create a cross-platform application that helps people to adopt and donate pets. The specific objectives are: to create the app, use user stories in order to find the requirements, use Flutter and Firebase in development, make a first version available and help the first users to use the app. To do this, the methodology that will be used is research regarding the nature of the qualitative approach, the applied type, exploratory character and bibliographic research technique. In order to generate knowledge to develop the application that will use the Flutter framework for the development of the front-end and communication with the Firebase api and the Firebase Storage, Firestore and Authentication tools, where it offers services as a back-end. Once developed the application will be exposed to difficulties encountered such as Firebase integration with the application and development functions. And finally, the conclusions and improvements that can be made in future work are presented.

**Keywords:** Adoption of pets. Flutter. Firebase. Multiplatform. Mobile app.

## LISTA DE FIGURAS

Figura 1 – Número de estrelas no <i>Github</i> . . . . .	12
Figura 2 – Número de pesquisa por termo no <i>Google trends</i> . . . . .	12
Figura 3 – Prós e contras do desenvolvimento nativo. . . . .	18
Figura 4 – Diferença entre híbrido e nativo. . . . .	19
Figura 5 – Arquitetura <i>Flutter</i> . . . . .	20
Figura 6 – Hierarquia de <i>widjets</i> no <i>Flutter</i> . . . . .	21
Figura 7 – Instalação <i>Android Studio MacOS</i> . . . . .	26
Figura 8 – Instalação <i>Plugin Flutter</i> no <i>Android Studio</i> . . . . .	26
Figura 9 – Instalação <i>Plugin Dart</i> no <i>Android Studio</i> . . . . .	26
Figura 10 – História de Usuário 1. . . . .	27
Figura 11 – História de Usuário 2. . . . .	27
Figura 12 – História de Usuário 3. . . . .	27
Figura 13 – História de Usuário 4. . . . .	28
Figura 14 – História de Usuário 5. . . . .	28
Figura 15 – História de Usuário 6. . . . .	28
Figura 16 – História de Usuário 7. . . . .	28
Figura 17 – História de Usuário 8. . . . .	28
Figura 18 – História de Usuário 9. . . . .	28
Figura 19 – História de Usuário 10. . . . .	29
Figura 20 – História de Usuário 11. . . . .	29
Figura 21 – Protótipo de tela <i>login</i> . . . . .	30
Figura 22 – Protótipo de tela esqueci minha senha. . . . .	31
Figura 23 – Protótipo de tela Cadastrado de Usuário. . . . .	32
Figura 24 – Protótipo de tela inicial. . . . .	33
Figura 25 – Protótipo de tela menu. . . . .	34
Figura 26 – Protótipo de tela detalhe pet visão para adotar. . . . .	35
Figura 27 – Protótipo de tela meus pets. . . . .	36
Figura 28 – Protótipo de tela detalhe pets visão edição. . . . .	37
Figura 29 – Protótipo de tela edição de pet. . . . .	38
Figura 30 – Protótipo de tela minha conta. . . . .	39
Figura 31 – Protótipo de tela cadastro de pet. . . . .	40

Figura 32 – Configuração do <i>Firebase</i> utilizando a plataforma <i>Android</i> . . . . .	41
Figura 33 – Adicionar chave <i>SHA1</i> . . . . .	42
Figura 34 – Estrutura do <i>Firestore</i> . . . . .	43
Figura 35 – Estrutura do <i>Storage</i> . . . . .	43
Figura 36 – Etapas para desenvolvimento. . . . .	44
Figura 37 – Estrutura do projeto. . . . .	45
Figura 38 – Código de controle de acesso a tela <i>Minha conta</i> . . . . .	46
Figura 39 – Código do componente <i>Image dialog</i> . . . . .	46
Figura 40 – Código do método <i>login</i> . . . . .	47
Figura 41 – Tela <i>login</i> . . . . .	48
Figura 42 – Tela cadastro de usuário. . . . .	49
Figura 43 – Tela inicial. . . . .	50
Figura 44 – Tela menu. . . . .	51
Figura 45 – Tela detalhe pet visão adotar. . . . .	52
Figura 46 – Tela meus pets. . . . .	53
Figura 47 – Tela detalhe meu pets edição e remoção. . . . .	54
Figura 48 – Tela cadastro de pets. . . . .	55
Figura 49 – Tela minha conta. . . . .	56
Figura 50 – Tela editar minha conta. . . . .	57
Figura 51 – Tela filtro de pesquisa. . . . .	58

## LISTA DE ABREVIATURAS E SIGLAS

<i>IOS</i>	<i>Iphone Operating System</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>APK</i>	<i>Android application pack</i>
<i>BaaS</i>	<i>Bankend As A Service</i>
<i>DMG</i>	<i>Disk Image File</i>
<i>GPS</i>	<i>Global Positioning System</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>JSON</i>	<i>Javascript Object Notation</i>
<i>MacOS</i>	<i>Sistema Operacional Mac</i>
<i>NoSQL</i>	<i>Not Only Standard Query Language</i>
<i>URL</i>	<i>Uniform Resource Locator</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Contextualização e delimitação do tema</b>	<b>11</b>
<b>1.2</b>	<b>Problematização e Objetivos</b>	<b>13</b>
<i>1.2.1</i>	<i>Objetivo geral</i>	<i>13</i>
<i>1.2.2</i>	<i>Objetivos específicos</i>	<i>14</i>
<b>1.3</b>	<b>Estrutura do trabalho</b>	<b>14</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>15</b>
<b>2.1</b>	<b>Abandono e adoção de pets</b>	<b>15</b>
<b>2.2</b>	<b>Desenvolvimento de aplicativos</b>	<b>17</b>
<b>2.3</b>	<b>Histórias de usuário</b>	<b>19</b>
<b>2.4</b>	<i>Flutter</i>	<i>20</i>
<b>2.5</b>	<i>NoSQL e Firebase</i>	<i>22</i>
<i>2.5.1</i>	<i>NoSQL</i>	<i>22</i>
<i>2.5.2</i>	<i>Firebase</i>	<i>22</i>
<b>3</b>	<b>METODOLOGIA</b>	<b>24</b>
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>25</b>
<b>4.1</b>	<b>Ambiente de desenvolvimento</b>	<b>25</b>
<b>4.2</b>	<b>Requisitos e prototipação</b>	<b>27</b>
<b>4.3</b>	<b>AdotaPet</b>	<b>40</b>
<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>59</b>
	<b>REFERÊNCIAS</b>	<b>61</b>

# 1 INTRODUÇÃO

## 1.1 Contextualização e delimitação do tema

O número de animais de estimação, também chamado de pets, no Brasil tem crescido a cada ano, segundo o (BRASIL, 2019) até o ano de 2018 o crescimento foi de aproximadamente 5% , chegando em 2018 a possuir mais de 54 milhões de cães e 23 milhões de gatos. Com esses números o Instituto Pet Brasil confirmou uma tendência de crescimento do número de animais de estimação, pois cada vez mais as pessoas e famílias buscam por esses, para fazer companhia, receber e dar atenção.

Essa busca cada vez maior por animais de estimação também tem aumentado o número de animais abandonados segundo estimativa da Organização Mundial da Saúde no Brasil existem 30 milhões de animais abandonados, sendo 10 milhões de gatos e 20 milhões de cães. Provocando vários problemas com animais de rua e com a sociedade, como diz a presidente da Comissão de Saúde Pública do Conselho Regional de Medicina Veterinária de São Paulo Adriana Vieira (PAULO, 2019).

"O contato direto com esses animais pode expor o indivíduo a riscos elevados de zoonoses, ou seja, de doenças que são transmissíveis aos seres humanos. Outro risco é desse animal, estando arreado, acabar atacando. Sem falar de outros agravos ao ser humano como, por exemplo, os acidentes de trânsito, quando o motorista tenta desviar de um animal."

Atualmente no Brasil existem algumas leis que protegem os animais de maus tratos e abandono são elas: Constituição Federal artigo 225, inciso VII, lei 9605/95 no seu artigo 32 e a lei 14064/20 que modifica a pena da lei 9605/95 quando o maus tratos se trata de cães e gatos.

Apesar das leis ainda se tem muito abandonos e uma das soluções para diminuir o número de animais abandonados é a adoção. Segundo (COMPANHIA, 2020) em pesquisa realizada pelo Instituto H2R, existe uma tendência de crescimento no número de adoção no Brasil, onde 33% das pessoas que adotaram um cão, 17% acharam o animal abandonado, 9% adotaram em feiras de adoção, já dos 59% de pessoas que adotaram um gato 44% acharam abandonados e 9% em feiras de adoção.

Existem vários sites que ajudam as pessoas a adotarem pets, alguns deles são: clube dos vira latas, adote um gatinho, olhar animal e adote petz.

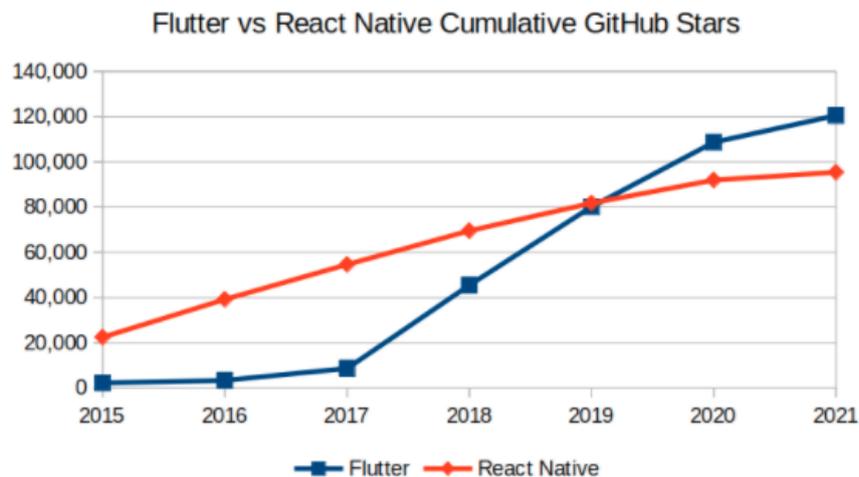
Porém, segundo pesquisa do Comitê Gestor da Internet no Brasil referente ao ano de 2019 (BRASIL, 2020) existem 134 milhões de usuários de internet no Brasil, sendo que os

brasileiro estão utilizando mais os celulares para acessarem a internet do que os computadores, onde 99% das pessoas que tem acesso utilizam o celular. A pesquisa também informa que o número de acesso através de computadores tem caído ano após ano.

Esse aumento no número de utilização dos celulares para acessar a internet cria a necessidade de aplicativos com as mais diversas funcionalidades, por isso nesse trabalho será realizado o desenvolvimento de um aplicativo para adoção de pets utilizando o *Flutter* e o *Firebase* ferramentas desenvolvidas pelo *Google*.

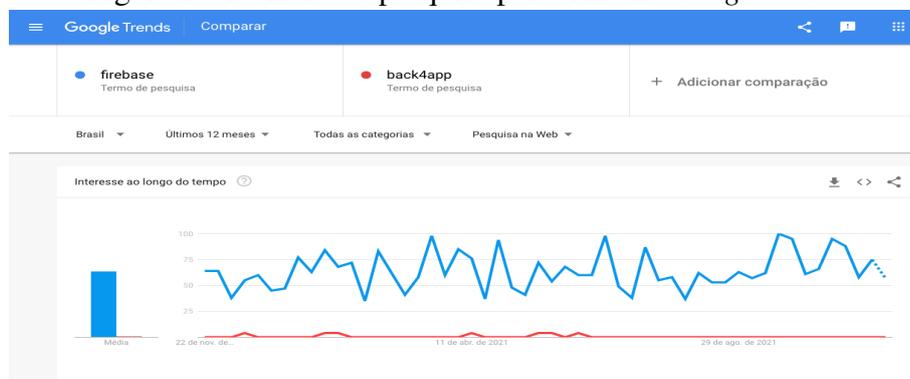
Nesse projeto será utilizado o *Flutter* como *framework* para o desenvolvimento e o *Firebase*, pois atualmente essas ferramentas são as mais utilizadas como é mostrado na figuras abaixo.

Figura 1 – Número de estrelas no *GitHub*.



Fonte: Jayakumar (2021).

Figura 2 – Número de pesquisa por termo no *Google trends*.



Fonte: Trends (2021).

Na figura 1 é mostrado o interesse ou agradecimento dos desenvolvedores em relação

a *Flutter* e *React native*, já na figura 2 é mostrado o número de pesquisa realizada por termos que no caso são *Firebase* e *Back4app*, onde a linha azul representa o *Firebase* e a linha vermelha o *Back4app*.

Entretanto essas ferramentas não foram escolhidas apenas por serem as mais utilizadas, mas também por terem outros benefícios como, documentação robusta e comunidade ativa.

## 1.2 Problematização e Objetivos

Na fase de elaboração desse projeto foi detectado que existia um grande número de animais de estimação abandonados e muitos donos de pets querendo conseguir uma outra pessoa interessada para doar seu pet, a motivação para efetuar essa doação é por diversos motivos como por exemplo: a mudança de moradia do dono, fazendo com que o novo local não fosse adequado para o animal de estimação. Outros problemas são a lacuna que existe entre o doador e a pessoa que adota animais e a falta de um ambiente de fácil acesso e uso, que possua uma interface de fácil entendimento.

Para tentar resolver estes problemas será realizado nesse trabalho o desenvolvimento de um aplicativo de adoção de pets onde as pessoas poderão adotar ou doar um pet. Além de ser livre, ou seja, qualquer pessoa pode ter acesso ao aplicativo, tanto pessoas físicas como instituições. Para desenvolver esse aplicativo será utilizado o kit de ferramentas do *Google* para criar aplicativos multiplataforma para celulares a partir de um único código fonte, conhecido como *Flutter* Valle (2020). E também será utilizado o *Firebase*, que é um *back-end* como serviço que serve para facilitar o uso de alguns recursos, tais como: autenticação, armazenamento, escalonamento, etc (ORLANDI, 2018).

O aplicativo será desenvolvido da seguinte maneira com uma listagem de todos os pets que estiverem disponível para adoção, com filtros para facilitar a busca, fotos dos pets e contato tanto de quem está colocando o pet a disposição como de quem quer adotar.

### 1.2.1 Objetivo geral

Este trabalho tem como objetivo construir um protótipo de aplicativo multi-plataforma para a adoção de animais de estimação, que pode beneficiar tanto as pessoas que pretendem adotar ou doar um pet.

### 1.2.2 *Objetivos específicos*

Os objetivos específicos deste trabalho são:

- Desenvolver um aplicativo que mostre os animais disponíveis para a adoção e que ajude as pessoas a disponibilizar animais no aplicativo.
- Utilizar o método de histórias de usuários para capturar os requisitos do aplicativo.
- Utilizar o kit de desenvolvimento *Flutter*, e as ferramentas do *Firebase* como *back-end* da solução proposta.
- Disponibilizar uma primeira versão.
- Realizar a inclusão dos primeiros animais e ajudar os primeiros usuários a utilizar o aplicativo.

### 1.3 **Estrutura do trabalho**

Esse trabalho está dividido em cinco capítulos, são eles: o capítulo 1 apresenta a problemática, objetivos gerais, e específicos, no capítulo 2 é apresentado o referencial teórico onde é apresentado os problemas do abandono e uma solução que é adotar os pets, os sistemas operacionais e os tipos de desenvolvimento, as ferramentas *Flutter* e *Firebase*, no capítulo 3 é mostrado a metodologia com sua abordagem, natureza, pesquisa e técnica utilizadas nesse projeto, no capítulo 4 é apresentado o desenvolvimento do projeto, com a preparação do ambiente de desenvolvimento, a coleta dos requisitos e a prototipação e por último o desenvolvimento do aplicativo e por fim é apresentada a conclusão do projeto e a perspectiva para trabalhos futuros no capítulo cinco.

## 2 REFERENCIAL TEÓRICO

Neste capítulo será mostrado o contexto de abandono e adoção de pets, os dois sistemas operacionais mais utilizados do mundo, como fazer para desenvolver para essas plataformas, o *framework* que será utilizado no desenvolvimento do aplicativo e as ferramentas que serão utilizadas.

### 2.1 Abandono e adoção de pets

O número de animais de estimação no Brasil é de 139,3 milhões segundo os dados do Instituto Pet Brasil Brasil (2019). Ainda segundo o Instituto o número de pets cresceu cerca de 5% desde a última consulta feita pelo Instituto Brasileiro de Geografia e Estatística- IBGE em 2013.

O aumento da população pet causa um aumento no número de abandono de animais de estimação. Isso faz com que se tenha um grande número de cães e gatos nas ruas das cidades brasileiras, gerando um problema, pois esses animais podem causar: agressões, acidentes de trânsito e transmitir doenças, dentre elas a raiva e a toxoplasmose (SCHERER *et al.*, 2021).

Entre as principais causas para o abandono dos pets estão a falta de planejamento para ter um animal de estimação, questões financeiras, disponibilidade de tempo e cuidado envolvido (PRADO *et al.*, 2015). Já (OURIQUES, 2018), diz que a principal causa de abandono é a mudança de residência dos tutores do pet.

Já em relação aos maus tratos, podemos destacar que na constituição brasileira tem em seu meio a preocupação com proteção dos animais e preservá-los, conforme está no artigo 225 inciso VII:

"Todos têm direito ao meio ambiente ecologicamente equilibrado, bem de uso comum do povo e essencial à sadia qualidade de vida, impondo-se ao poder público e à coletividade o dever de defendê-lo e preservá-lo para as presentes e futuras gerações.

§ 1º Para assegurar a efetividade desse direito, incumbe ao poder público: VII - proteger a fauna e a flora, vedadas, na forma da lei, as práticas que coloquem em risco sua função ecológica, provoquem a extinção de espécies ou submetam os animais a crueldade Brasil (1988)."

Existem também leis federais que protege os animais, sendo a mais importante para os protetores dos animais a Lei federal 9605/98 no seu artigo 32 Strazzi (2015). A lei 9605/98 tem uma pena para maus tratos de detenção de três meses a um ano, e multa. Como é dito no artigo 32 Brasil (1998):

"Praticar ato de abuso, maus-tratos, ferir ou mutilar animais silvestres, domésticos ou domesticados, nativos ou exóticos: Pena - detenção, de três meses a um ano, e multa Brasil (1998).".

A lei federal 14064/20 modifica a pena da lei 9605/98 quando se trata de maus tratos de cães e gatos, houve um aumento na pena para reclusão de dois a cinco anos, multa e proibição da guarda Brasil (2020).

Porém alguns municípios encontram dificuldades de executar a Lei 9605/98, pois para se fazer valer as determinações legais é preciso um fiscal ambiental, e muitas vezes não é possível para as prefeituras, por questões financeiras ou por falta de pessoal, de acordo com (CISCATO *et al.*, 2016).

Uma das maneiras de reduzir o aumento de animais abandonados seria adoção, pois além de tirar o animal da rua, ainda tira da zona de risco da zoonoses e outras doenças e dá a chance ao animal de ter uma qualidade de vida Unesp (2015).

"A escolha de um cão ou gato não é um processo fácil, embora muitas pessoas não acreditem nisso. Estas pessoas normalmente escolhem um animal apenas considerando beleza e porte, esquecendo-se das especificidade de cada raça. Pelos longos que necessitam escovação diária, predisposições a doenças degenerativas temperamento mais ou menos agitado, comportamento destrutivos em casa, tamanho maior do que o esperado, necessidade de interação e socialização, adaptabilidade com outros animais, são fatores que dificilmente são levados em consideração, e que mais tarde são responsáveis por altos índices de abandono ao longo da vida do animal SCHULTZ (2013, apud OLIVEIRA *et al.* (2016))."

Segundo Schultz (2013, apud (OLIVEIRA *et al.*, 2016)) a adoção não muda apenas a vida do animal adotado, mas também promove mudanças na rotina e vida do novo tutor, pois o animal adotado exige cuidado e responsabilidade, como atenção veterinária, tempo, paciência, companheirismo.

Para facilitar essa adoção foi criado em Fortaleza a Lei municipal 11704/21 como é dito no artigo 1, de acordo com (FORTALEZA, 2021):

Art. 1º-Fica instituída, no âmbito do Município de Fortaleza, a Semana da Adoção Responsável de Animais, a ser realizada anualmente na primeira semana do mês de outubro, que obedecerá às disposições previstas nesta Lei e terá como objetivos: (NR); I - estimular a adoção e posse responsável dos animais, conforme legislação vigente; Fortaleza (2021)

## 2.2 Desenvolvimento de aplicativos

Segundo Franco (2021), os dois sistemas operacionais para dispositivos móveis mais populares do mundo são o *Android* e *Iphone Operating System (IOS)*, sendo o primeiro o mais utilizado.

O *Android* foi criado em 2003 por Andy Rubin, Rich Miner, Nick Sears e Chris White, esses fundaram a empresa *Android Inc* em Palo Alto, Califórnia. Em 2005 essa empresa foi comprada pela *Google*, e foi definido que seria utilizado o *Linux* como base para o sistema operacional *Android*, com isso o *Android* poderia ser disponibilizado para fabricantes de celulares e terceiros gratuitamente Lima (2017). Com essa decisão muitas fabricantes passaram a utilizar o *Android*, tornando assim o sistema operacional mais utilizado no mundo. Hoje o *Android* é mantido pela *Open Handset Alliance*, um grupo de 84 empresas que desenvolvem o *Android Alliance* (2021).

Já o *IOS* criado pela *Apple Inc.* foi lançado em 2007, quando lançado era um projeto inovador com funcionalidades jamais vistas Magalhães (2018). O *IOS* foi criado para ser utilizado apenas nos dispositivos da *Apple* como, *Iphones, Ipad, Ipod touch e Apple watch* mais recentemente *Passos* (2019). Diferentemente do *Android* onde várias empresas têm acesso e contribuem para o seu desenvolvimento no *IOS* apenas a *Apple* faz o seu desenvolvimento e só ela utiliza o sistema operacional.

Para fazer o desenvolvimento de aplicativos para dispositivos móveis existem duas abordagens principais, a nativa e a híbrida BUENO (2021).

Os aplicativos nativos são desenvolvidos utilizando as ferramentas, *Application Programming Interface (API)* e linguagens de programação específicas fornecidas pelas plataformas El-Kassas *et al.* (2017).

As aplicações nativas são definidas por Monde (2011):

"As aplicações nativas são desenvolvidas para uma única plataforma, com a capacidade de executar e explorar todas as funções que tal plataforma em questão dispõe. Ou seja, os aplicativos que são criados exclusivamente para iOS, Android e/ou Windows Phone, entre outras, conseguem acessar todo o potencial do dispositivo através da própria arquitetura do sistema operacional, como câmera, calendário, álbum de fotos, GPS, entre outros. É dito que um aplicativo nativo é todo programa construído sob medida para uma única plataforma, com o intuito de funcionar sob medida para os dispositivos em conjunto com suas especificidades. O desenvolvimento de aplicações móveis nativas teve início juntamente com a popularização dos smartphones na segunda metade da década passada e se transformando numa tendência para década seguinte e até o presente momento. apud Reis (2019)"

As plataformas mais utilizadas são o *Android* e *IOS*, para desenvolver nesses sistemas operacionais são utilizadas as linguagens de programação *Java* e *Kotlin* no primeiro e *Objective-c* e *Swift* no último Madureira (2020).

A programação nativa tem seus prós e contras como é visto na imagem a seguir El-Kassas *et al.* (2017).

Figura 3 – Prós e contras do desenvolvimento nativo.

Prós	Contra
Tenha APIs completas para acessar todos os recursos do dispositivo móvel, como câmera, sensores, acesso à rede, GPS, armazenamento de arquivos, banco de dados, SMS e e-mail	Os aplicativos nativos são mais difíceis de desenvolver e exigem um alto nível de experiência
Aparência e comportamento nativos da interface do usuário	Eles precisam ser desenvolvidos separadamente para cada plataforma, aumentando assim o tempo, o custo e os esforços de desenvolvimento
Melhor desempenho	Restrições e custos associados ao desenvolvimento e implantação em certas plataformas

Fonte: El-Kassas *et al.* (2017).

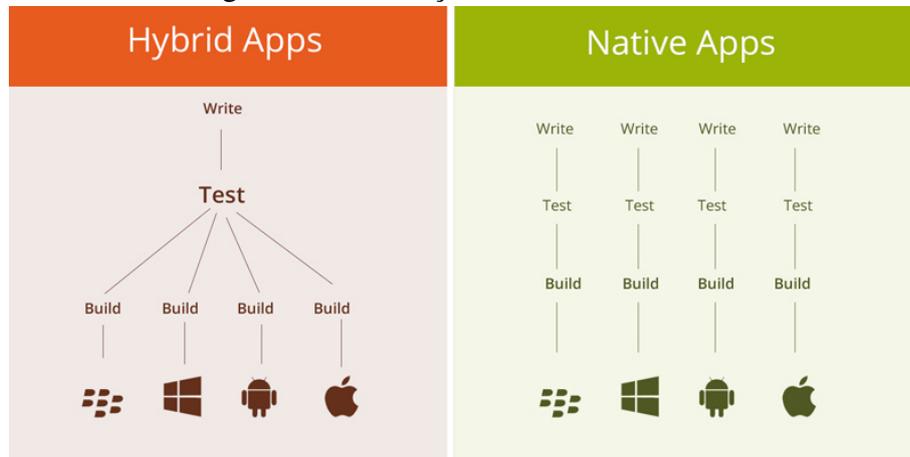
Taqtile (), afirma que no desenvolvimento híbrido com apenas um código fonte é possível desenvolver para os dois sistemas operacionais *Android* e *IOS*. Seguindo três abordagens, o encapsulamento de código em contêineres nativos. Desenvolvimento utilizando uma linguagem nativa ou não, e que depois o *framework* compila para as plataformas. E por último o desenvolvimento misto que ocorre entre a segunda abordagem e código nativo.

De acordo com Biørn-Hansen *et al.* (2018), a adoção dessas ferramentas para o desenvolvimento de aplicativos móveis tem crescido nos últimos anos, principalmente por reduzirem o custo, tempo e conhecimento no desenvolvimento desses.

A imagem a seguir mostra a diferença do desenvolvimento de um aplicativo nativo

para um aplicativo multi plataforma.

Figura 4 – Diferença entre híbrido e nativo.



Fonte: Minds (2021).

Segundo Luetic (2021) os cinco principais frameworks de desenvolvimento multi plataforma são: *Flutter*, *React native*, *Xamarin*, *Ionic* e *Adobe PhoneGap*.

### 2.3 Histórias de usuário

As histórias de usuário (*user stories*) foram criadas pelo processo ágil *Extreme Programming* (xp) na década de 80. Depois de anos as histórias foram incorporadas pelo *framework* ágil *Scrum* fazendo com que muitas pessoas pensem que as histórias são práticas do *Scrum* e esquecendo da origem delas no xp (CAPELLÃO, 2020).

As histórias de usuário são uma representação informal dos requisitos como diz (LONGO; SILVA, 2014):

A história de usuário é uma declaração informal de um requisito de usuário em vez de um grande documento de requisitos. A intenção real da história de usuário é fornecer à equipe uma capacidade para responder rapidamente o que o usuário quer e precisa. A história de usuário cria menos sobrecarga de documentação e mostra de forma rápida a evolução das necessidades do mundo real ou a descoberta de novos requisitos baseados no trabalho em andamento (LONGO; SILVA, 2014).

De acordo com (CAPELLÃO, 2020) histórias de usuário são a representação das necessidades do usuário, podendo ser usada para organizar e definir os requisitos. Descrevendo de maneira simples e curta as funcionalidades, apenas com o essencial para se fazer uma estimativa de risco mesmo com baixa qualidade e focando nas necessidades e benefícios para o usuário.

## 2.4 Flutter

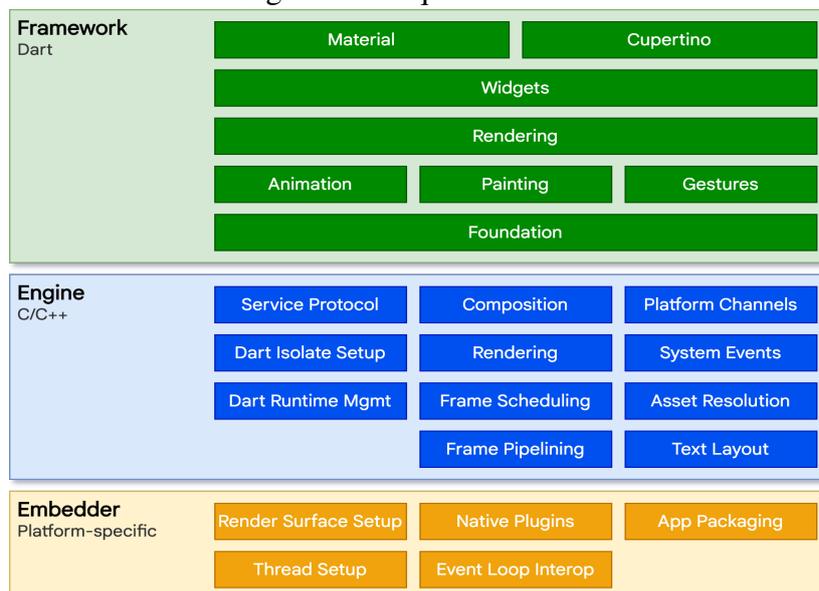
Em 2014 foi iniciado um projeto com o codinome de *Sky*, com o intuito de encontrar a melhor forma de construir *interface mobile*, em 2015 esse projeto mudou de nome para *Flutter*. Sua primeira aparição em grandes eventos foi em 2017 no *Google IO*, em 2018 foi lançado a primeira versão estável para os desenvolvedores Melo (2019).

O *Flutter* é definido na documentação como: "kit de ferramentas interface de usuário do *Google* para construir aplicativos bonitos e nativamente compilados para dispositivos móveis, *web*, *desktop* e incorporados a partir de uma única base de código"Flutter (2021a).

A linguagem de programação utilizada pelo *Flutter* é o *Dart*, sendo essa criada pelo *Google* em 2011, para ser rápida, reativa, adaptável a multi plataformas e possuir sintaxe clara e consistente Dantas (2020)

A arquitetura do *Flutter* é dividida em três camadas o *framework*, *Engine*, *Embedder*. Onde na camada do *framework* é fornecido um moderno *framework* reativo escrito na linguagem *Dart*. Essa camada possui um rico conjunto de plataforma, *layout* e bibliotecas básicas, formado por uma série de camadas. Já a camada de *engine* é escrita em *c++*, e oferece o suporte aos primitivos necessários para da suporte a todos os aplicativos *Flutter*. Por fim o *embedder* é o responsável por fornecer um ponto de entrada para aplicações e por coordenar acesso aos serviços dos sistemas operacionais como superfícies de renderização, acessibilidade e entrada. E gerencia o *loop* de eventos de mensagens (FLUTTER, 2021c).

Figura 5 – Arquitetura *Flutter*.



Fonte: Flutter (2021c).

O *Flutter* possui alguns recursos importantes, como o *hot reload*, bibliotecas internacionalizadas e *widgets*. O *hot reload* é uma funcionalidade que atualiza o aplicativo imediatamente após alterações no código, servindo para correções rápidas de *bugs* bem como para testar novas funcionalidades e visual. As bibliotecas internacionalizadas servem para fazer a internacionalização do aplicativo sem ser preciso muito trabalho Luetic (2021).

O desenvolvimento em *Flutter* é baseado na criação de *widgets*, como diz Corazza (2018):

"existem widgets para definir elementos estruturais (botões, menus...), elementos de estilo (fontes, cores...), aspectos de layouts (margens, espaçamentos...), além de widgets com design específico para a plataforma Android (Material Components) e iOS (Cupertino). Além disso, o Flutter foi projetado para facilitar a criação de novos widgets e a personalização dos existentes Corazza (2018)."

De acordo com Leal *et al.* (2019) os *widgets* são ordenados de forma hierárquica, assim eles são agrupados sob um outro *widget* pai, herdado características desse. Normalmente os *widgets* são compostos por outros *widgets* menores, que definem suas características, como cor, tamanho da fonte, alinhamento e outros.

Figura 6 – Hierarquia de *widgets* no *Flutter*.



Fonte: Flutter (2021c).

Como exibido na figura 6, existem duas categorias de *widgets* no *Flutter*, o *Stateless Widget* e os *Stateful Widget*. Onde o *Stateless* são *widgets* que não mudam o estado, depois de criadas permanecem do mesmo modo durante todo seu ciclo de vida. Já as *Stateful* são *widgets* que possuem mudança no seu estado, seja essa mudança causada pelo usuário por qualquer outro fator Leal *et al.* (2019).

## 2.5 *NoSQL e Firebase*

### 2.5.1 *NoSQL*

O termo *Not Only Standard Query Language* (NoSQL) surgiu em 2009 em uma reunião promovida por Johan Oskarsson que tinha como finalidade falar sobre bancos de dados não relacionais, distribuídos e de código aberto. Esse termo depois passou a descrever soluções de armazenamento de dados não relacionais (SADALAGE; FOWLER, 2019). A criação do *NoSQL* se deve ao fato do aumento do volume e variedade dos dados (CUER, 2015).

Segundo (MOURA; AMORIM, 2014) a sociedade está cada vez mais conectada, passando a enviar e receber dados através da internet como, vídeos, fotos, mensagens de texto, dados de *Global Positioning System* (GPS), redes sociais, entre outros, esses dados fazem parte de um grande banco de dados não estruturado.

O *NoSQL* pode ser dividido em 4 grupos principais, sendo eles: chave valor, documentos, orientado a colunas e orientado a grafos. Onde o primeiro possui uma chave com um valor atribuído a ela, o segundo utiliza o método de armazenamento em modo de coleções e documentos em formato padronizados como *Javascript Object Notation* (JSON), o terceiro se caracteriza por armazenar e processar um grande volume de dados distribuídos, já o último utiliza um modelo que tem três componentes, nós, relacionamentos e as propriedades, onde cada nó se conecta com diversas arestas, sendo muito utilizado para realizar consultas complexas Muniz *et al.* (2018).

Algumas das características do *NoSQL* são: escalabilidade horizontal, ausência de esquema, suporte nativo a replicação, API simples para acesso aos dados. Onde a ausência de esquema proporciona a facilidade da escalabilidade e aumento da disponibilidade, mas não garante a integridade dos dados. As soluções *NoSQL* não estão preocupadas em como os dados são armazenados e sim em recuperar os dados de maneira eficiente afim de gerar alta disponibilidade e escalabilidade, por isso as API simples são desenvolvidas para facilitar o acesso aos dados por qualquer aplicação de forma rápida e eficiente (CUER, 2015).

### 2.5.2 *Firestore*

O *Firestore* é uma plataforma digital utilizada para facilitar o desenvolvimento de aplicações web e mobile, esse serviço oferecido pelo *Firestore* é conhecido como *Backend As A Service* (BaaS). O *Firestore* foi criado por James Tamplin e Andrew Lee no ano de 2011, em

2014 foi adquirido pelo *Google* Coodesh (2021).

O BaaS é definido por Batschinski (2016) como:

"um serviço de computação em nuvem que serve como middleware. O mesmo fornece aos desenvolvedores uma forma para conectar suas aplicações mobile e web a serviços na nuvem a partir de APIs e SDKs Batschinski (2016)."

Segundo Andrade (2020), o *BaaS* é um tipo de serviço que provê a infraestrutura e o *back-end* de uma aplicação de forma simplificada, assim não é preciso o desenvolvimento dessa solução manualmente. Algumas das funcionalidades mais comuns oferecidas pelo *BaaS* são: autenticação, armazenamento, notificações, escalabilidade e outros. Com isso os desenvolvedores podem focar apenas na construção do *front-end*.

O *Firebase* possui diversos recursos, como *realtime database*, *google analytics*, *cloud firestore*, *authentication*, *cloud storage* e outros *Firebase* (2021a). Dentre estes destacamos, *cloud firestore*, *authentication*, *cloud storage* que serão utilizado nesse trabalho. O *cloud firestore* é um banco de dados *NoSQL* flexível e escalonável, com sincronia com os aplicativos clientes em tempo real, além de oferecer suporte *off-line* *Firebase* (2021b). O *Authentication* é o responsável por fazer a autenticação segura e de melhor experiência de *login* para o usuário final, esse *login* pode ser feito de diversas maneiras como, senhas, números de telefone, *login* do *google*, *facebook*, *twitter*, *github*, *apple* e outros *Firebase* (2021c). Já o *cloud storage* é um serviço de armazenagem, fotos e vídeos, simples e econômico criado para a escala do *Google*. Com ele é possível que o usuário final tenha a capacidade de fazer *download* e *upload* com a segurança do *Google* *Firebase* (2021d).

### 3 METODOLOGIA

A necessidade de se criar meios para facilitar a adoção e doação de pets será desenvolvido um aplicativo multi-plataforma que utilizará o *framework Flutter* e os serviços do *Firebase* de forma que ao final do desenvolvimento será possível a doação e adoção de pets pelo aplicativo.

Para isso a pesquisa será quanto a abordagem do tipo qualitativa, pois não tem preocupação com a representatividade numérica e sim com o aprofundamento da compreensão de um grupo social (GERHARDT; SILVEIRA, 2009).

Quanto a natureza será do tipo aplicada. A pesquisa aplicada segundo (GERHARDT; SILVEIRA, 2009) "objetiva gerar conhecimentos para aplicação prática, dirigidos à soluções de problemas específicos."

Quanto aos objetivos a pesquisa é de caráter exploratória, pois como (GERHARDT; SILVEIRA, 2009) diz: "este tipo de pesquisa tem como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses."

A técnica utilizada nessa pesquisa foi a pesquisa bibliográfica, onde (FONSECA, 2002 apud (GERHARDT; SILVEIRA, 2009)) diz: "A pesquisa bibliográfica é feita a partir do levantamento de referências teóricas já analisadas, e publicadas por meios escritos e eletrônicos, como livros, artigos científicos, páginas de *web sites*."

O desenvolvimento da pesquisa está dividido em:

- Revisão bibliográfica e pesquisa.
- Escolha das ferramentas utilizadas no desenvolvimento do aplicativo.
- Estudo das ferramentas.
- Desenvolvimento do aplicativo.
- Resultado obtido.

## 4 DESENVOLVIMENTO

Nesse capítulo será mostrado o desenvolvimento do aplicativo AdotaPet. O desenvolvimento será dividido em três etapas: instalação do ambiente de desenvolvimento, coleta de requisitos e prototipação do aplicativo e apresentação do aplicativo desenvolvido.

### 4.1 Ambiente de desenvolvimento

Para desenvolver um aplicativo em *Flutter* não é preciso utilizar um *Integrated Development Environment* (IDE), pode-se usar qualquer editor de texto combinado com algumas ferramentas de linha de comando, mas para se ter uma melhor experiência de desenvolvimento é recomendado que se utilize uma IDE, pois essas fornecem várias funcionalidades que irão ajudar no desenvolvimento como, autocompletar, destaque de sintaxe, assistências de edição de *widget*, suporte para execução e depuração dentre outras (FLUTTER, 2021b).

A IDE que será utilizada para o desenvolvimento do aplicativo nesse projeto é o *Android Studio*, que é a IDE oficial para desenvolvimento para aplicativos na plataforma *Android* (STUDIO, 2021). Além de possuir suporte a vários recursos que ajudam no desenvolvimento.

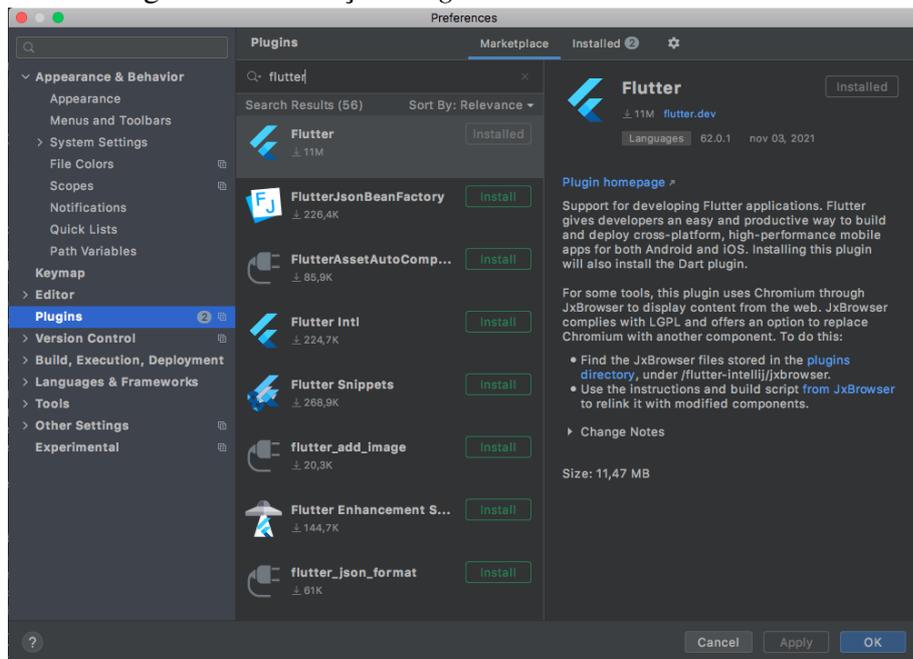
Para ter o *Android Studio* no seu computador é fácil, basta baixar gratuitamente no site oficial e depois é só instalar. No Sistema Operacional Mac (MacOS) que é o sistema operacional utilizado no computador para realizar esse projeto, é só executar o arquivo *Disk Image File* (DMG) arrastar o *Android Studio* para a pasta Aplicativos como na figura abaixo, depois é só executar e instalar o que se pede.

A instalação do *Flutter* é bem simples, é só baixar no site oficial, depois irá extrair o arquivo no local desejado e por fim irá criar a variável de ambiente como no tutorial do site <https://flutter.dev/docs/get-started/install/macos>.

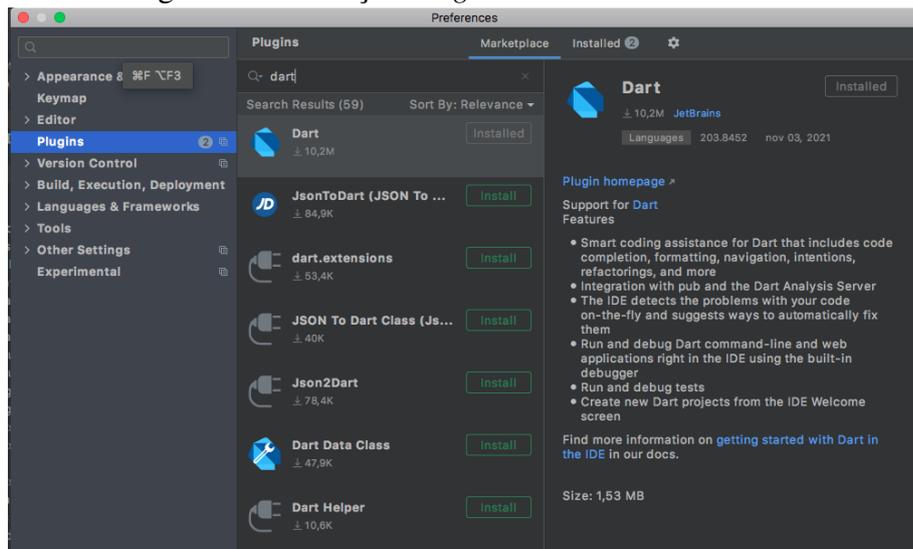
Por último para ajudar no desenvolvimento do aplicativo será instalado os *plugins* que irão auxiliar nessa tarefa, para isso é necessário abrir o *Android Studio*, ir em preferências, *plugins* e pesquisar e instalar o *Flutter e dart*.

Figura 7 – Instalação *Android Studio MacOS*.

Fonte: Próprio autor (2021).

Figura 8 – Instalação *Plugin Flutter no Android Studio*.

Fonte: Próprio autor (2021).

Figura 9 – Instalação *Plugin Dart no Android Studio*.

## 4.2 Requisitos e prototipação

De acordo com (SOUSA, 2018) o levantamento de requisito é o início de qualquer projeto de software, pois é através dos resultados obtidos nessa etapa que será feito a definição das próximas etapas do desenvolvimento do projeto. Nessa etapa é onde são definidos as funcionalidades e o escopo do projeto.

Para Fazer o levantamento de requisitos nesse projeto foi utilizados o método das histórias de usuários (*user stories*). Pois com diz (HELM; WILDT, 2014):

é totalmente possível especificar software de uma forma muito mais efetiva, simples e até divertida do que o mercado normalmente tem feito ao longo dos anos. Essa forma de especificação de requisitos mais eficiente se chama Histórias de Usuário (HELM; WILDT, 2014).

As histórias de usuários nesse projeto seguiram o modelo de (COHN, 2008): Como um, eu quero, de modo a, sendo essa última cláusula opcional. A seguir estão as histórias de usuários criadas nesse trabalho.

Figura 10 – História de Usuário 1.

Como um: Responsável pelo aplicativo
Eu quero: Que os usuários possam visualizar os pets disponíveis para adoção mesmo sem está logado
De modo a:

Fonte: Próprio autor (2021).

Figura 11 – História de Usuário 2.

Como um: Responsável pelo aplicativo
Eu quero: Que os usuários do sistema façam o login
De modo a: Poder cadastrar pets e ou adotar pet

Fonte: Próprio autor (2021).

Figura 12 – História de Usuário 3.

Como um: Responsável pelo aplicativo
Eu quero: Que o usuário seja capaz de se cadastrar no aplicativo
De modo a: Guardar email, nome, senha, telefone do usuário

Fonte: Próprio autor (2021).

Figura 13 – História de Usuário 4.

Como um: Responsável pelo aplicativo
Eu quero: Que o usuário possa recuperar a senha
De modo a:

Fonte: Próprio autor (2021).

Figura 14 – História de Usuário 5.

Como um: Usuário
Eu quero: Ver detalhes dos pets disponíveis para adoção
De modo a: Ver imagens e os dados

Fonte: Próprio autor (2021).

Figura 15 – História de Usuário 6.

Como um: Usuário
Eu quero: Cadastrar pet no aplicativo
De modo a: Que esses possam ficar visíveis para outras pessoas poderem adotar

Fonte: Próprio autor (2021).

Figura 16 – História de Usuário 7.

Como um: Usuário
Eu quero: Ver detalhes dos meus pet
De modo a: Ver imagens e os dados

Fonte: Próprio autor (2021).

Figura 17 – História de Usuário 8.

Como um: Usuário
Eu quero: Editar o perfil dos meus pets cadastrados
De modo a: Poder alterar os dados

Fonte: Próprio autor (2021).

Figura 18 – História de Usuário 9.

Como um: Usuário
Eu quero: Remover o perfil do meu pet cadastrado
De modo a: Apagar o perfil do aplicativo

Fonte: Próprio autor (2021).

Figura 19 – História de Usuário 10.

Como um: Usuário
Eu quero: Editar meu perfil
De modo a: Poder alterar telefone, senha

Fonte: Próprio autor (2021).

Figura 20 – História de Usuário 11.

Como um: Usuário
Eu quero: Poder ver o telefone do Usuário que está disponibilizando o pet
De modo a: Poder se candidatar a adoção

Fonte: Próprio autor (2021).

Depois das histórias de usuário foi realizado uma prototipação. De acordo com (PIAZZA, 2021) uma prototipação de baixa fidelidade é simples de ser realizada com baixo nível de detalhes, porém representa as funcionalidades, além de ser barato e fácil de se fazer, uma maneira de realizar essa prototipação é desenhando a tela do aplicativo em uma folha de papel, nesse projeto foi realizado a prototipação utilizando a ferramenta Figma.

A seguir será mostrado a prototipação das telas do aplicativo AdotaPet. A primeira tela que será exibida é a tela de login que mostra os campos de *email* e senha, *links* para recuperar a senha e se cadastrar e um botão para efetuar o login na aplicação como se ver na figura 21.

Figura 21 – Protótipo de tela *login*.

login



Email

Senha

[Esqueci minha senha](#)

Entrar

[cadastre-se](#)

Detailed description: This is a wireframe of a login screen. At the top left, the word 'login' is written in a small, grey font. Below it, centered, is a blue icon of a person sitting on the ground. The main content is contained within a light blue rectangular area. Inside this area, there are two white input fields. The first field is labeled 'Email' in blue text. The second field is labeled 'Senha' in blue text. To the right of the password field, there is a link that says 'Esqueci minha senha' in a smaller, grey font. Below the input fields is a solid blue button with the white text 'Entrar'. At the bottom of the light blue area, there is a link that says 'cadastre-se' in a smaller, grey font.

Fonte: Próprio autor (2021).

A tela de esqueci minha senha mostrado na figura 22, possui um campo de *email* e um botão para confirmar a solicitação de recuperação de senha.

Figura 22 – Protótipo de tela esqueci minha senha.

O protótipo da tela 'esqueci minha senha' apresenta um layout simples e funcional. No topo, há um ícone de uma pessoa sentada com uma seta apontando para cima, indicando uma ação de recuperação. Abaixo do ícone, há um campo de entrada de texto rotulado 'Email' em azul. Logo abaixo do campo, há um botão de ação rotulado 'Confirmar' em um fundo azul escuro com o texto em branco. O título da tela, 'esqueci minha senha', está no topo da interface.

Fonte: Próprio autor (2021).

Na figura 23 é mostrado a tela de cadastro que possui um formulário com os campos nome, *email*, senha, confirmar senha e data de nascimento, e tem um botão para o usuário se cadastrar no sistema.

Figura 23 – Protótipo de tela Cadastrado de Usuário.



cadastro de usuario



Nome

Email

Senha

Confirmar Senha

Data de Nascimento

Cadastrar

Detailed description: This is a wireframe of a user registration screen. At the top, there is a header 'cadastro de usuario' and a blue icon of a person hugging a dog. Below this is a light blue container with five white input fields stacked vertically, labeled 'Nome', 'Email', 'Senha', 'Confirmar Senha', and 'Data de Nascimento'. At the bottom of this container is a dark blue button with the text 'Cadastrar' in white.

Fonte: Próprio autor (2021).

A figura 24 que será exibida a seguir mostra a tela inicial do aplicativo, onde todos os usuários têm acesso, estando *logados* ou não, a tela é composta de um botão de menu, nome da aplicação e de blocos clicáveis que são compostos de uma imagem do animal de estimação e seus dados.

Figura 24 – Protótipo de tela inicial.



Fonte: Próprio autor (2021).

A tela de menu é dividida em duas partes, onde na superior é apresentado a logo da aplicação e ao lado os dados do usuário se ele estiver *logado* e um *link* para fazer o *login* se ele não estiver *logado*. Na parte inferior tem os botões para o usuário navegar entre as telas da aplicação e o botão sair. A figura 25 apresenta essas funções.

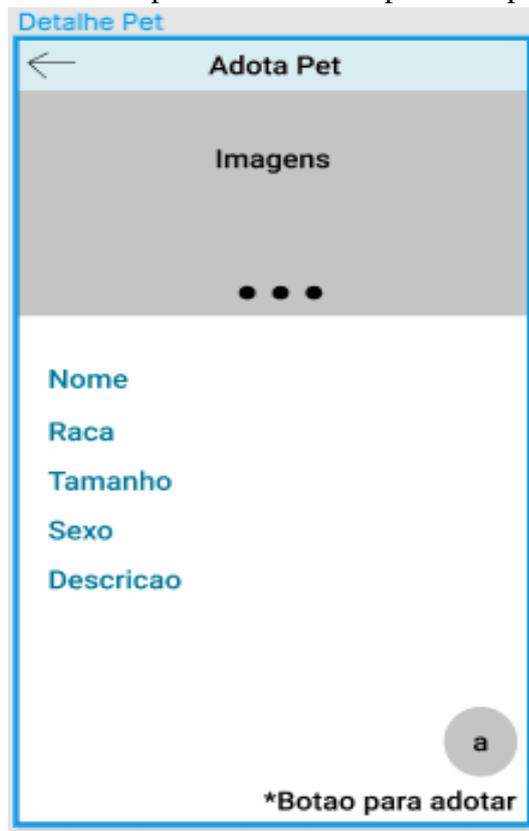
Figura 25 – Protótipo de tela menu.



Fonte: Próprio autor (2021).

A tela de prototipação a seguir é a de detalhe do pet e ela mostra um campo no formato carrossel com algumas imagens, logo abaixo os campos com os dados sobre o pet e no canto inferior direito um botão que tem a função de levar o usuário para o *Whatsapp* abrindo uma conversa com o tutor do pet. A figura 26 mostrará o protótipo dessa tela.

Figura 26 – Protótipo de tela detalhe pet visão para adotar.



Fonte: Próprio autor (2021).

Na figura 27 será exibido o protótipo de tela que mostra os pets do usuário, onde essa possui blocos com imagem e dados do pets, sendo esse bloco clicável com a função de ir para uma página detalhada do pet.

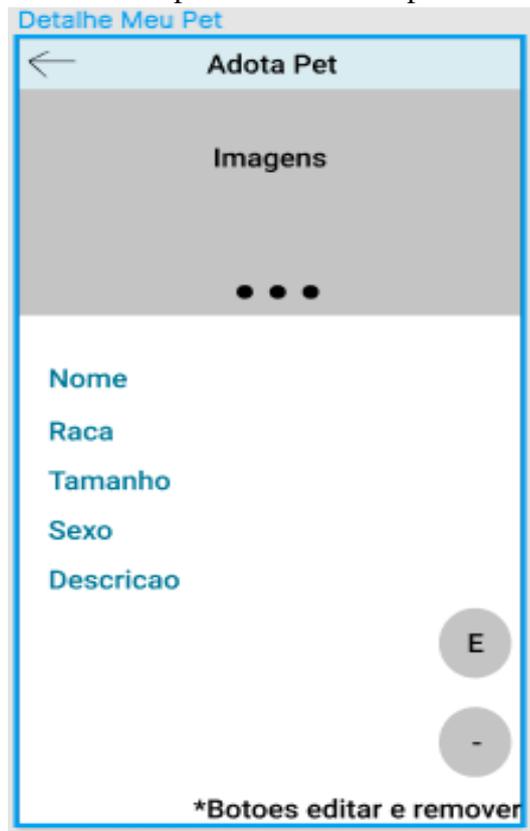
Figura 27 – Protótipo de tela meus pets.



Fonte: Próprio autor (2021).

A tela a seguir apresenta um campo no formato carrossel com algumas imagens, logo abaixo os campos com os dados sobre o pet e no canto inferior direito dois botões, um que tem a função de levar o usuário para a edição do pet e outro que tem a função de remover o pet. A figura 28 apresentará o protótipo dessa tela.

Figura 28 – Protótipo de tela detalhe pets visão edição.



Fonte: Próprio autor (2021).

No protótipo de tela abaixo será exibido a tela de edição do pet que possui os seguintes campos, um botão de para adicionar imagens, formulário com os dados de cadastro do pet e botão para salvar a edição do animal de estimação. Na figura 29 a seguir será mostrado a tela.

Figura 29 – Protótipo de tela edição de pet.



editar pet

← Adota Pet

Editar Pet

+

Nome

Raca

Idade

Tamanho

pequeno medio grande

Sexo

femea macho

Descricao

Salvar

Detailed description: This is a mobile app prototype for editing a pet's profile. The screen is titled 'Adota Pet' with a back arrow on the left. Below the title is a sub-header 'Editar Pet' and a plus sign icon. The form contains several input fields: 'Nome', 'Raca', and 'Idade'. There are two sections of radio buttons: 'Tamanho' with options 'pequeno', 'medio', and 'grande'; and 'Sexo' with options 'femea' and 'macho'. A larger text area is labeled 'Descricao'. At the bottom, there is a blue 'Salvar' button.

Fonte: Próprio autor (2021).

Nesse protótipo de tela será mostrado um campo com a logo da aplicação, um campo com os dados do usuário e botão pra editar os dados. A figura 30 mostra a prototipação da tela minha conta.

Figura 30 – Protótipo de tela minha conta.



Fonte: Próprio autor (2021).

A última tela de prototipação é a tela de cadastro de pet, onde essa possui um botão para adicionar imagens, um campo de formulário dos dados do pet e um botão para cadastrar o animal de estimação. A figura 31 mostra essa tela.

Figura 31 – Protótipo de tela cadastro de pet.



O protótipo da tela de cadastro de pet, intitulada "Adota Pet", apresenta um formulário para registrar um novo animal. O formulário contém os seguintes campos e opções:

- Um ícone de "+" para adicionar um novo pet.
- Campos de texto para "Nome", "Raca" e "Idade".
- Seção "Tamanho" com botões de opção para "pequeno", "medio" e "grande".
- Seção "Sexo" com botões de opção para "femea" e "macho".
- Um campo de texto maior para "Descricao".
- Um botão azul "Cadastrar" para finalizar o registro.

Fonte: Próprio autor (2021).

### 4.3 AdotaPet

Nessa secção será apresentado as dificuldades encontradas para desenvolver o aplicativo, estrutura do banco de dados e o desenvolvimento do aplicativo.

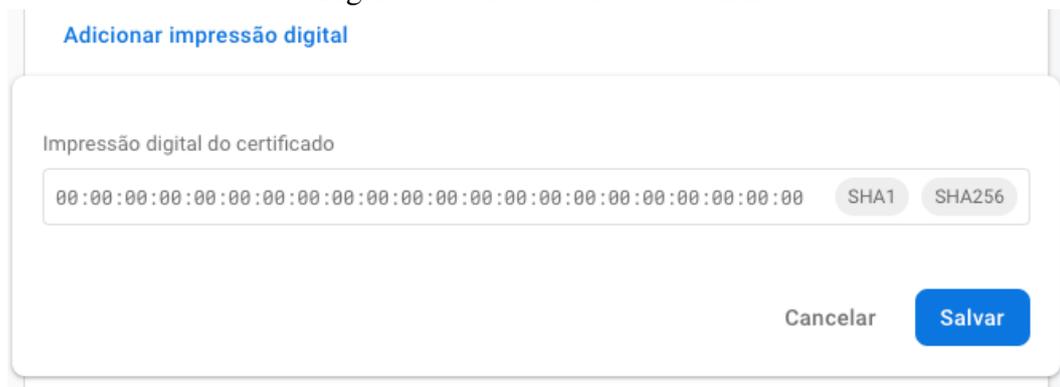
A primeira dificuldade encontrada para desenvolver esse aplicativo foi a configuração do *Firebase* com o aplicativo. As imagens a seguir mostrarão o passo a passo de como configurar o *Firebase*.



*build.gradle*.

Outra dificuldade encontrada foi na utilização do *Firebase* por dois computadores para o desenvolvimento, pois era preciso compartilhar as chaves *sha1* dos dois computadores no *Firebase* para que esses pudessem utilizar, para que os computadores pudessem gerar a *build* de depuração de forma que essas conseguissem ter acesso ao *Firebase*. A seguir será mostrado o compartilhamento das chaves.

Figura 33 – Adicionar chave *SHA1*.



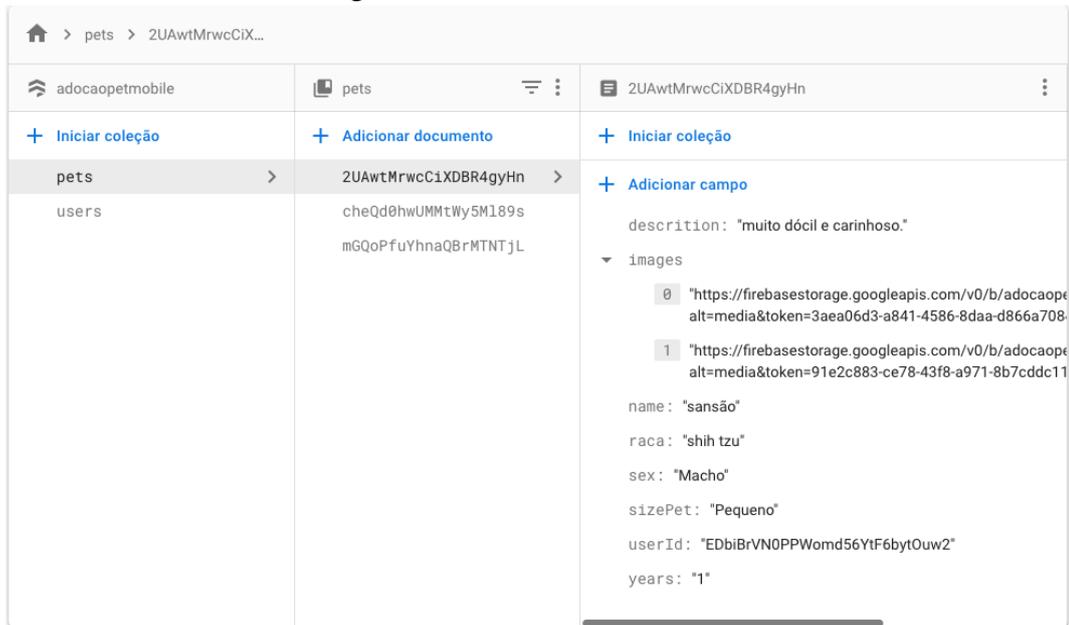
Fonte: Próprio autor (2021).

A adição de uma chave *sha1* é bem simples basta ir nas configurações, na aba geral e clicar no botão adicionar impressão digital e colocar a *sha1* do computador que irá ser utilizado para fazer a depuração do aplicativo.

Algumas funções do projeto também causaram dificuldades como a atualização do dados do pet, cadastro do pet e mensagem de erro para o usuário, pois na atualização é preciso salvar as novas imagens e apagar as imagens antigas do *Storage* o que deu um certo trabalho, no cadastro foi a realização da tela e como salvar as imagens no *Storage*, já nas mensagens de erro por não ter muito conhecimento em tratamento de erros acabou causando dificuldades.

O tempo para desenvolver esse aplicativo também foi uma dificuldade, pois o tempo era dividido em estudo, trabalho e a realização da implementação desse projeto, fazendo com que se tivesse poucas horas disponível para o desenvolvimento.

A estrutura criada no *Firestore* é muito simples com apenas duas coleções a *pets* e a *users*, onde a primeira guarda os dados dos pets como nome, raça, sexo, idade, tamanho, id do usuário que esta disponibilizando pra adoção, a *Uniform Resource Locator* (URL) das imagens e a descrição. Já a última guarda uma coleção onde é salvo os Ids dos pets e os dados do usuário como nome, *email*, telefone, data de nascimento e *URL* da imagem. Na imagem a seguir será mostrado a estrutura do *Firestore*.

Figura 34 – Estrutura do *Firestore*.

Fonte: Próprio autor (2021).

Já *Storage* que é banco onde fica salvo as imagens do nosso projeto, a estrutura é simples, pois as imagens são salvas na pasta inicial do banco de dados. A seguir é mostrado a estrutura do *Storage*.

Figura 35 – Estrutura do *Storage*.

Name	Tamanho	Tipo	Última modificação
1636239280717	71.44 KB	image/jpeg	6 de nov. de 2021
1636239286573	30.04 KB	image/jpeg	6 de nov. de 2021
1636239417739	33.47 KB	image/jpeg	6 de nov. de 2021
1636239544102	59.88 KB	image/jpeg	6 de nov. de 2021
1638065665506	31.19 KB	image/jpeg	27 de nov. de 2021
1638066224613	32.86 KB	image/jpeg	27 de nov. de 2021
1638068166393	56.49 KB	image/jpeg	27 de nov. de 2021

Fonte: Próprio autor (2021).

O desenvolvimento foi realizado em etapas onde essas eram definidas de acordo com os objetivos. Os objetivos eram definidos de acordo com as histórias de usuário e com a prototipação das telas. Depois de ter o objetivo definido era então estipulado o prazo para

conclusão daquela etapa.

A seguir será exibido uma imagem com todas as etapas do desenvolvimento:

Figura 36 – Etapas para desenvolvimento.

Etapas	Objetivos	Prazo	Entrega
1	Criar conexão entre o app e o Firebase	1 dia	atraso
2	Criar coleções do banco de dados e forma de login no Firebase	1 dia	ok
3	Criar tela de login com sua funcionalidade de logar	3 dias	ok
4	Criar tela de cadastro de usuário, salvar no Firestore	3 dias	ok
5	Criar tela e funcionalidade de esqueci minha senha	2 dia	ok
6	Criar a tela home	2 dia	ok
7	Criar menu e Logout do app	3 dias	antecipado
8	Criar tela e funcionalidade do cadastro de pets	5 dias	atraso
9	Criar funcionalidade da tela home	3 dias	antecipado
10	Criar tela de detalhe do pet	3 dias	ok
11	Redirecionar o usuário para o whatsapp quando ele solicita a adoção do pet	1 dia	ok
12	Criar tela meus pets e funcionalidades	3 dias	ok
13	Criar tela de edição de pet e remoção e sua funcionalidade	3 dias	atraso
14	Criar tratamento de erros	2 dias	atraso
15	Criar tela minha conta e funcionalidades	2 dias	antecipado
16	Criar tela de edição da minha conta	2 dia	ok
17	Criar tela de filter e funcionalidades	6 dias	ok
18	Criar controle de acesso na telas	4 dias	ok

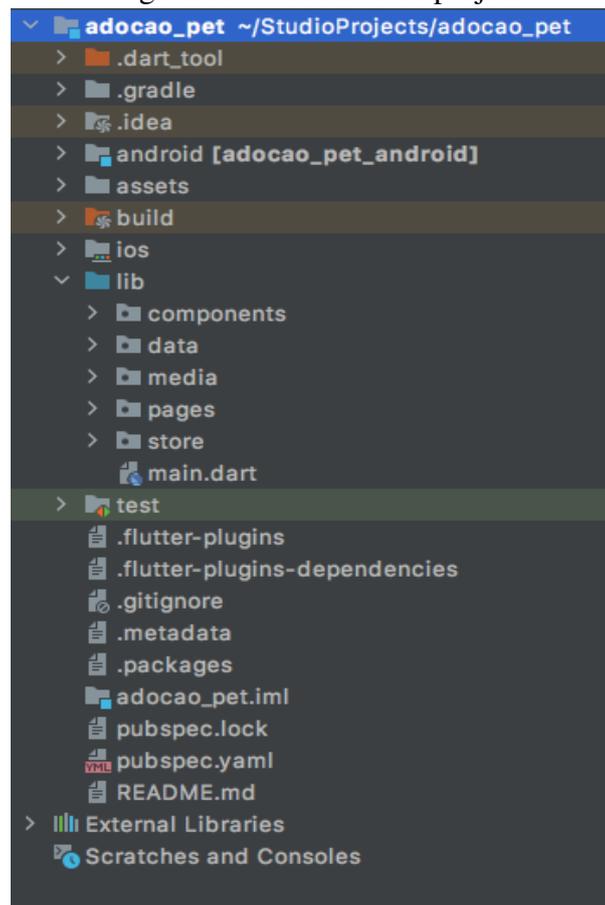
Fonte: Próprio autor (2021).

Algumas etapas foram entregues com atraso, já outras foram entregues antecipadamente como foi apresentado na figura 36. Mesmo com essas etapas que não seguiram o cronograma planejado o projeto acabou não sendo prejudicado.

A estrutura do projeto está seguindo o modelo padrão e na pasta *lib* está dividida da seguinte maneira com as pastas *components*, *data*, *media*, *pages* e *store*. Onde os componentes

ficam na pasta *components*, na pasta *data* tem a classe *petData* que possui o método para alimentar as variáveis com os dados do banco e o método que transforma os dados da variáveis de forma que o banco aceite. Na pasta *media* esta a classe *media config* que possui um método para fazer a escala de certos elementos utilizados pra diferentes tamanhos de tela. Na pasta *pages* estão as telas e na *store* estão as classe utilizadas com *mobx* para controle de estado. Como é mostrado na imagem abaixo.

Figura 37 – Estrutura do projeto.



Fonte: Próprio autor (2021).

A seguir é mostrado um trecho de código, onde é feito o controle de acesso a tela *Minha conta*, onde se o usuário não tiver *logado* não irá conseguir ter acesso a tela, pois será redirecionado para a tela *Home*.

Figura 38 – Código de controle de acesso a tela Minha conta.

```

@override
void didChangeDependencies() {
  super.didChangeDependencies();
  disposer = autorun((){
    if(!loginStore.loggedIn){
      pageStore.setPage(0);
      Navigator.of(context).push(
        MaterialPageRoute(
          builder: (_) => InicialScreen())); // MaterialPageRoute
    }
  });
}

```

Fonte: Próprio autor (2021).

No código da figura 38 tem o método *didChangeDependencies* que é chamado toda vez que uma dependência desse estado é alterado, dentro dele tem o método *autorun* que é executado assim que declarado e será chamado toda vez que os valores dentro de sua função forem alterados. Na condição *if* dentro do *autorun* é analisado se o usuário não está logado, se a condição for verdadeira a variável *page* é setada com 0 e o usuário é redirecionado para a tela Inicial, que é composta de quatro páginas sendo a 0 a página *Home*.

Na imagem abaixo será exibido o componente utilizado para mostrar as imagens do pet no cadastro e edição em tamanho maior e um botão onde pode deletar aquela imagem do cadastro ou edição.

Figura 39 – Código do componente *Image dialog*.

```

@override
Widget build(BuildContext context) {
  return Dialog(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: <Widget>[
        image is File
          ? Image.file(image)
          : Image.network(image),
        FlatButton(
          child: Text("Deletar"),
          textColor: Colors.red,
          onPressed: () {
            Navigator.of(context).pop();
            onDelete();
          },
        ), // FlatButton
      ], // <Widget>[]
    ), // Column
  ); // Dialog
}

```

Fonte: Próprio autor (2021).

No trecho de código da figura 39 é mostrado um *widget* que retorna um diálogo que possui um coluna que tem uma imagem e um botão, onde a imagem pode ser do tipo arquivo ou do tipo *URL* da internet, já o botão possui o método de deletar fazendo com que a imagem em questão seja deletada do processo de cadastro ou edição de um pet.

Na imagem seguir é apresentado o método *login*.

Figura 40 – Código do método *login*.

```
@action
Future<bool> LoginWithEmailAndPassword({
  @required String email, @required String password }) async {
  loading = true;
  try{
    FirebaseUser user;
    await FirebaseAuth.instance
      .signInWithEmailAndPassword(email: email, password: password).then((u){
        user = u.user;
        currentUser = Observable(user);
        loading = false;
        loggedIn = true;
        _dadosUser(currentUser.value.uid);
        petStore.petChange = true;
        petStore.getMyPets(currentUser.value.uid);
        return Future.value(true);
      });
    return Future.value(true);
  } on PlatformException catch(error){
    if(error.code == "ERROR_USER_NOT_FOUND"){
      setErrorLogin("Esse email não está cadastrado!");
    }
    if(error.code == "ERROR_WRONG_PASSWORD"){
      setErrorLogin("Senha Inválida!");
    }
    loading = false;
    loggedIn = false;
    return Future.value(false);
  }
}
```

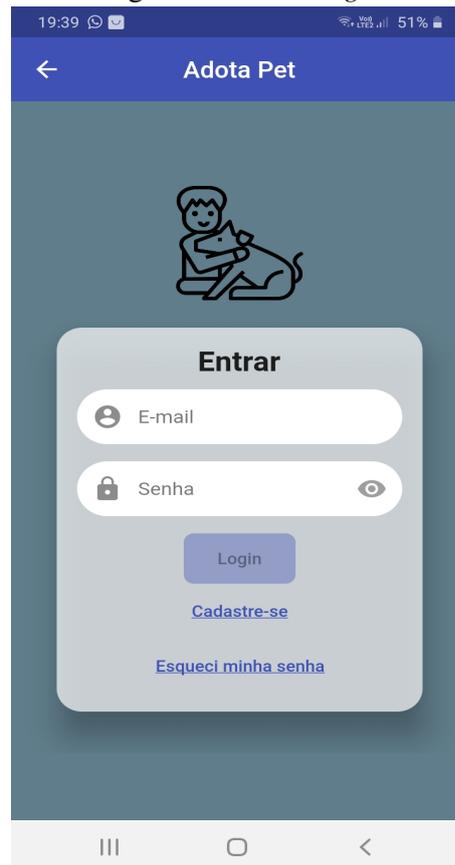
Fonte: Próprio autor (2021).

Na figura 40 é exibido o trecho de código do método *login* com *email* e senha que utiliza o *Firestore*, o método é do tipo futuro *bool* assíncrono e recebe o *email* e senha como parâmetros, depois utiliza o *try catch*, dentro do *try* é feita a conexão com o *Firestore* chamando o método *signInWithEmailAndPassword* do *auth* onde é feito o login se *email* e senha estiverem corretos, depois o usuário é carregado na variável *currentUser*, o método *dadosUser* é chamado para carregar os dados de telefone e data de aniversário do usuário e por último chama o método para carregar os pets desse usuário. Já se ocorrer algum erro o *catch* é chamado e é feito o

tratamento de erro, para poder mostrar ao usuário o que está acontecendo. A seguir será mostrado as telas desenvolvidas nesse trabalho.

A primeira tela a ser apresentada é a de *login* onde é mostrado a logo da aplicação e campos de *email* e senha, *links* para recuperar a senha e se cadastrar, e um botão para efetuar o *login* na aplicação como se ver na figura 41:

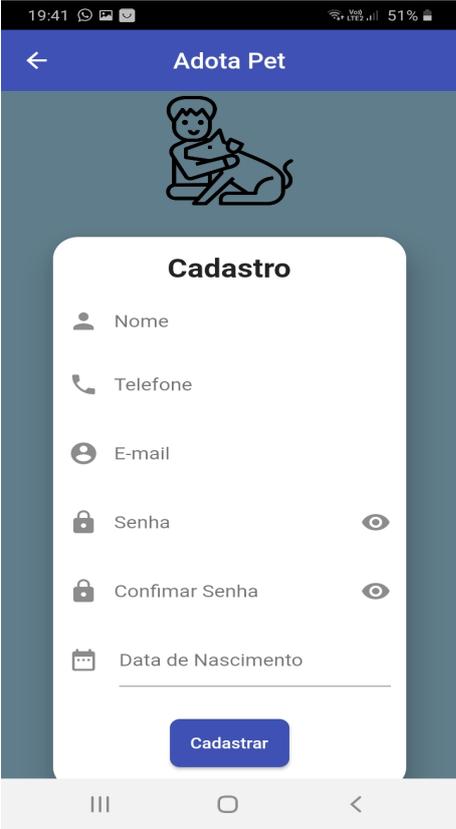
Figura 41 – Tela *login*.



Fonte: Próprio autor (2021).

Na figura 42 é mostrado a tela de cadastro de usuário que possui a logo da aplicação, formulário com os dados do usuário e um botão para realizar o cadastro ao ser realizado o cadastro o usuário será redirecionado para a tela inicial.

Figura 42 – Tela cadastro de usuário.



A imagem mostra a interface de usuário de um aplicativo móvel. No topo, há uma barra azul com o texto "Adota Pet" e um ícone de seta para trás. Abaixo, há um ícone de uma pessoa abraçando um cachorro. O formulário de cadastro, intitulado "Cadastro", contém os seguintes campos:

- Nome
- Telefone
- E-mail
- Senha (com ícone de olho para alternar visibilidade)
- Confirmar Senha (com ícone de olho para alternar visibilidade)
- Data de Nascimento

Um botão azul "Cadastrar" está localizado na base do formulário. A barra de navegação do sistema Android é visível na base da tela.

Fonte: Próprio autor (2021).

A seguir é exibida a tela inicial que possui blocos de imagem e dados clicáveis, que leva para a página de detalhe do pet, o menu no canto superior esquerdo e um botão para pesquisar no canto superior direito. Como é exibido na figura 43.

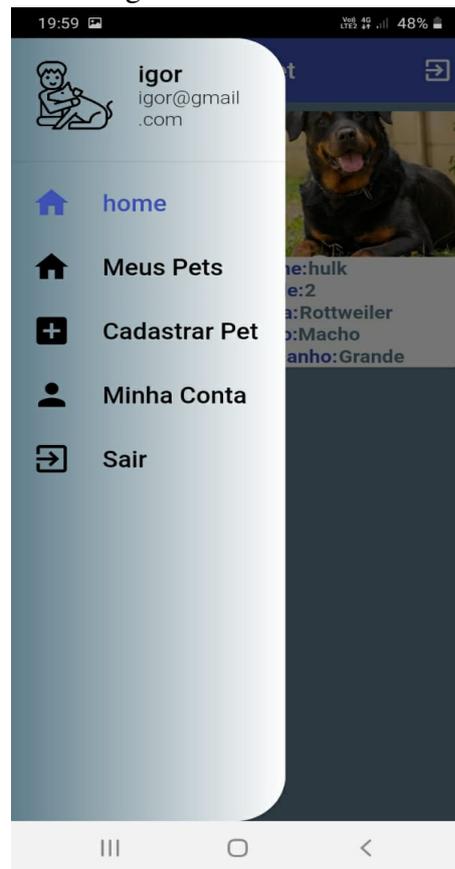
Figura 43 – Tela inicial.



Fonte: Próprio autor (2021).

A tela de menu é dividida em duas partes, onde na superior é apresentado a logo da aplicação e ao lado os dados do usuário se ele estiver *logado* e um *link* para fazer o *login* se ele não estiver *logado*. Na parte inferior tem os botões para o usuário navegar entre as telas da aplicação e o botão sair. A figura 44 apresenta essas funções.

Figura 44 – Tela menu.



Fonte: Próprio autor (2021).

A tela de detalhe pet que será mostrada a seguir apresenta um campo no formato carrossel, onde é exibida as imagens do animal de estimação, logo abaixo os campos com os dados sobre o pet e no canto inferior direito um botão que tem a função de levar o usuário para o *Whatsapp* abrindo uma conversa com o tutor do pet. A figura 45 mostrará essa tela.

Figura 45 – Tela detalhe pet visão adotar.



Fonte: Próprio autor (2021).

A tela meus pets é muito parecida com a tela inicial, as únicas diferenças são que nela apenas os animais de estimação do usuário irão ser apresentados e o campo de pesquisa não existe nessa tela, no lugar tem o botão de sair. Essa tela será exibida na figura 46.

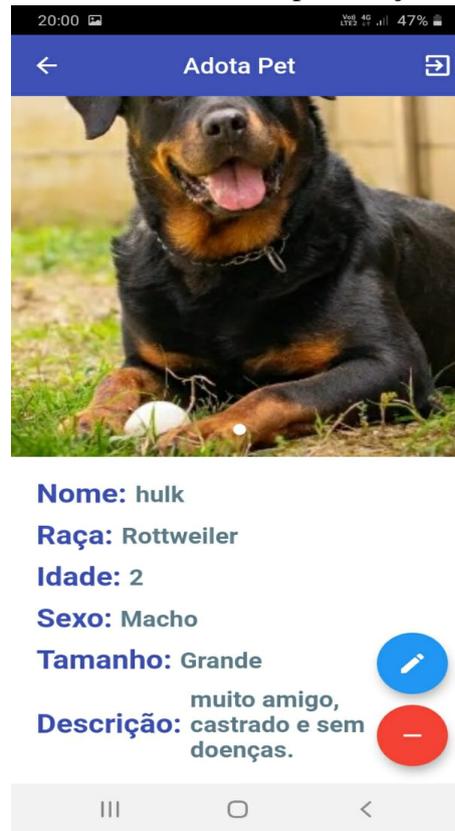
Figura 46 – Tela meus pets.



Fonte: Próprio autor (2021).

A tela detalhe do meu pet que será exibida a seguir é muito parecida com a figura 45, a única diferença são os botões de editar e remover que essa tela possui, onde no botão editar o usuário é direcionado para a tela de edição e no botão remover o usuário remove o pet. Essa tela é exibida na figura 47.

Figura 47 – Tela detalhe meu pets edição e remoção.



Fonte: Próprio autor (2021).

A tela de cadastro de pet possui, campo para adicionar imagens, formulário para cadastro do animal de estimação e um botão para cadastrar. A figura 31 mostra essa tela.

Figura 48 – Tela cadastro de pets.

19:45 50%

Adota Pet

Nome do pet

Raça

Idade

Tamanho

Pequeno Médio Grande

Sexo

Fêmea Macho

Descrição

Cadastrar

Fonte: Próprio autor (2021).

Na figura 30 é apresentada a tela minha conta que tem a logo da aplicação, os dados do usuário e um botão pra edição dos dados.

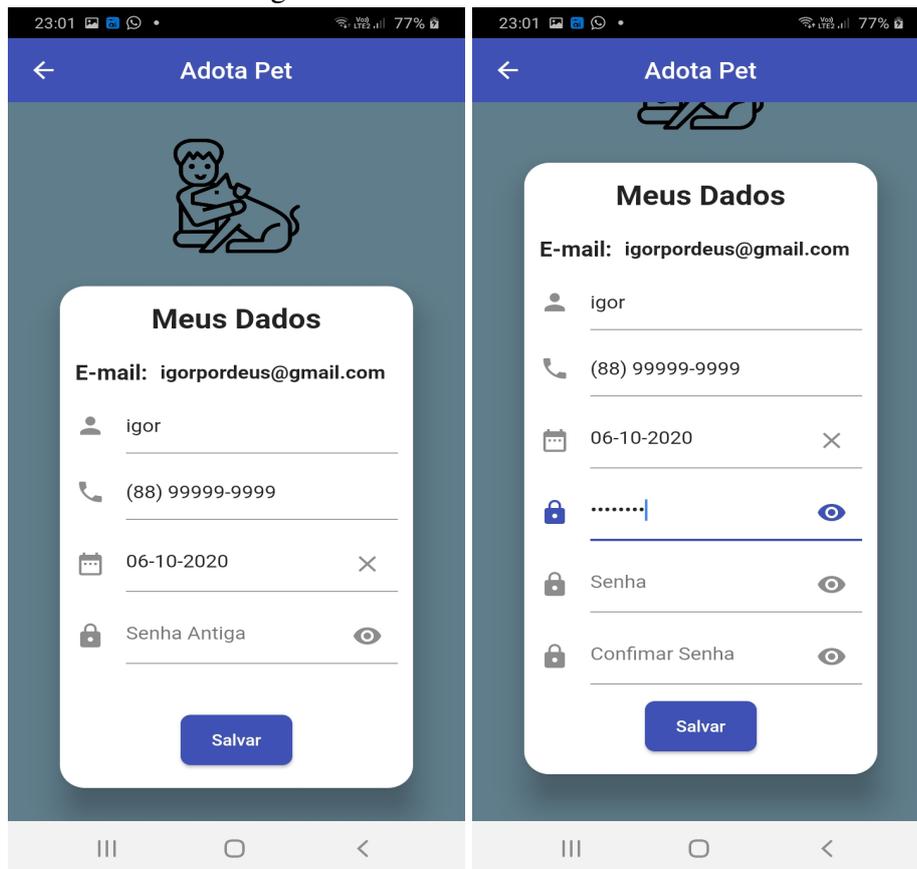
Figura 49 – Tela minha conta.



Fonte: Próprio autor (2021).

Na figura 50 é apresentada a tela de edição dos dados do usuário, a tela da esquerda mostra o *email* e os campos que podem ser modificados, se o usuário quiser modificar os dados sem mudar a senha ele pode realizar essa ação deixando o campo senha antiga em branco e apertando o botão salvar, se ele quiser modificar a senha, então quando a senha antiga for digitada aparecerá os campos para se colocar a nova senha e a confirmação da nova senha.

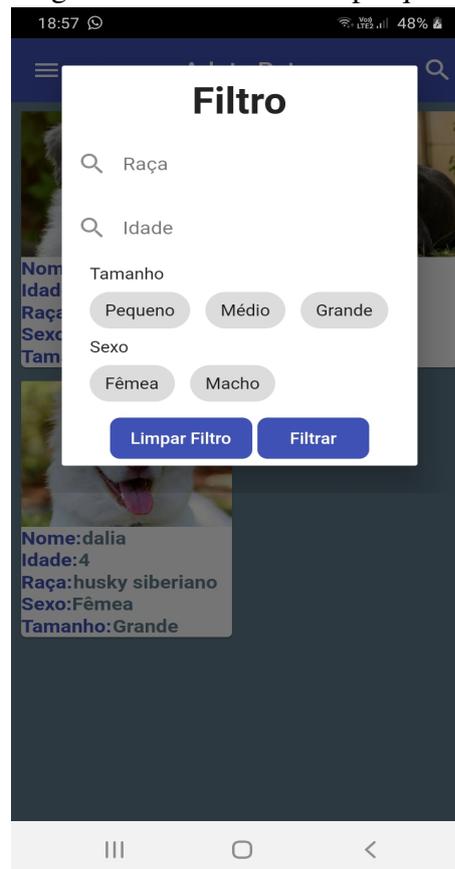
Figura 50 – Tela editar minha conta.



Fonte: Próprio autor (2021).

E por último é apresentado a tela de filtros onde o usuário pode colocar os filtros para a pesquisa e tem o botão de limpar os filtros. A figura 51 exhibe essa tela.

Figura 51 – Tela filtro de pesquisa.



Fonte: Próprio autor (2021).

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Nesse capítulo serão apresentadas as conclusões atingidas com o desenvolvimento desse projeto, também serão mostrados os trabalhos futuros que podem dar prosseguimento no desenvolvimento deste trabalho.

A motivação para realizar esse projeto vem da empatia que o autor tem ao ver um animal de rua sofrendo maus tratos, com fome, frio, sofrendo com doenças, entre outros.

Durante a pesquisa foi percebido que muitas pessoas abandonam seu pet por não possuírem as condições necessárias para criá-los, por mudança de lar, por que o pet faz dano, entre outras situações. E que existe um número crescente de pessoas que estão procurando animais para adotarem sendo essa uma das soluções para diminuir o número de animais abandonados.

A pesquisa também identificou que os usuários de internet estão cada vez mais utilizando os celulares para acessar a rede em vez de utilizar os computadores, gerando uma demanda por aplicativos cada vez maior.

Assim surgiu o primeiro objetivo que tinha como finalidade desenvolver um aplicativo que mostrasse os animais disponíveis para adoção e que ajudasse as pessoas a disponibilizar seu pet para adoção. Esse objetivo foi atingido com o desenvolvimento do aplicativo.

O desenvolvimento do aplicativo foi iniciado depois de ser feito as histórias do usuário, onde foram capturados os requisitos do aplicativo fazendo com que o segundo objetivo fosse alcançado.

O aplicativo foi desenvolvido utilizando o *framework Flutter*, pois é um dos *framework* mais utilizados atualmente e faz o desenvolvimento multi-plataforma, e o *cloud firestore*, *authentication*, *cloud storage* ferramentas do *Firebase*, que facilitaram muito o desenvolvimento, pois como o *Firebase* oferece o serviço como *back-end*, facilitou bastante nos métodos de *login*, cadastro, inserção e consulta no banco. Atingindo assim mais um objetivo.

A primeira versão do aplicativo foi disponibilizada apenas para um pequeno grupo de pessoas, pois assim seria possível mostrar como se utiliza o AdotaPet e ajudar elas a inserir os primeiros pets para adoção. Para que essas pudessem utilizar o aplicativo foi disponibilizado o *Android application pack (APK)* do aplicativo *android* para que assim elas instalassem em seus celulares. Alcançado assim mais dois objetivos.

Quando todos esses objetivos anteriores foram atingidos, foi então que o objetivo geral foi alcançado, pois o desenvolvimento de um aplicativo para dispositivo móvel multi-plataforma para adoção e doação de pets tinha sido obtido.

Nesse projeto também foi encontradas dificuldades, como configuração do *Firebase* com o aplicativo, utilização de dois computadores para depurar o aplicativo com acesso ao *Firebase*, pois era preciso compartilhar a chave *shal* dos computadores para o *Firebase*, algumas funções no desenvolvimento como atualização do dados do pet e mensagem de erro, e a maior dificuldade o tempo que foi muito curto para o desenvolvimento do trabalho.

Desde o começo do projeto foi surgindo funcionalidades novas para o projeto, mas como o tempo é curto e essas funcionalidades não fundamentais para o funcionamento do aplicativo, foram deixadas de lado, podendo ser realizadas em trabalhos futuros, essas funcionalidades são:

- login através de redes sociais como *Gmail*, *Facebook* e *Apple ID*.
- Criação de *chat* para que os usuários interajam dentro da própria plataforma.
- Compartilhamento dos animais para adoção em redes sociais com *Whatsapp* e *Instagram*.
- Criação de campos de localização para saber quais animais tem perto do usuário e mostrando esses pet no mapa.
- Colocar a possibilidade de vídeos no cadastro do pets.
- Melhorias de *layout*, para o aplicativo ficar mais bonito.
- Fazer um trabalho de interface humano computador para avaliar a usabilidade do aplicativo.
- Fazer questionários para avaliar o aplicativo.
- Criar uma funcionalidades para divulgar animais perdidos, com a possibilidade de compartilhar nas redes sociais.

## REFERÊNCIAS

- ALLIANCE, O. H. **What would it take to build a better mobile phone?** 2021. Disponível em: <<https://www.openhandsetalliance.com/>>. Acesso em: 25 de Setembro de 2021.
- ANDRADE, A. P. de. **O que é Firebase?** 2020. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-firebase>>. Acesso em: 10 de Outubro de 2021.
- BATSCHINSKI, G. **Backend as a Service: Prós e Contras.** 2016. Disponível em: <<https://www.infoq.com/br/news/2016/07/backend-pros-e-contras/>>. Acesso em: 10 de Outubro de 2021.
- BIØRN-HANSEN, A.; GRØNLI, T.-M.; GHINEA, G. A survey and taxonomy of core concepts and research challenges in cross-platform mobile development. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 51, n. 5, p. 1–34, 2018.
- BRASIL. **LEI Nº 9.605, DE 12 DE FEVEREIRO DE 1998: Dispõe sobre as sanções penais e administrativas derivadas de condutas e atividades lesivas ao meio ambiente, e dá outras providências.** 1998. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/19605.htm](http://www.planalto.gov.br/ccivil_03/leis/19605.htm)>. Acesso em: 19 de Setembro de 2021.
- BRASIL. **LEI Nº 14.064, DE 29 DE SETEMBRO DE 2020: Altera a Lei nº 9.605, de 12 de fevereiro de 1998, para aumentar as penas cominadas ao crime de maus-tratos aos animais quando se tratar de cão ou gato.** 2020. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_ato2019-2022/2020/lei/L14064.htm](http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2020/lei/L14064.htm)>. Acesso em: 19 de Setembro de 2021.
- BRASIL, C. **CONSTITUIÇÃO DA REPÚBLICA FEDERATIVA DO BRASIL.** 1988. Disponível em: <[https://normas.leg.br/?urn=urn:lex:br:federal:constituicao:1988-10-05;1988#/con1988\\_26.06.2019/art\\_225\\_.asp](https://normas.leg.br/?urn=urn:lex:br:federal:constituicao:1988-10-05;1988#/con1988_26.06.2019/art_225_.asp)>. Acesso em: 19 de Setembro de 2021.
- BRASIL, C. G. da Internet no. **TIC Domicílios 2019.** 2020. Disponível em: <[https://cetic.br/media/analises/tic\\_domicilios\\_2019\\_coletiva\\_imprensa.pdf](https://cetic.br/media/analises/tic_domicilios_2019_coletiva_imprensa.pdf)>. Acesso em: 18 de Novembro de 2021.
- BRASIL, I. P. **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: 06 de abril de 2021.
- BUENO, C. E. D. O. **DESENVOLVIMENTO DE UM APLICATIVO UTILIZANDO O FRAMEWORK FLUTTER E ARQUITETURA LIMPA.** Pontifícia Universidade Católica de Goiás, 2021.
- CAPELLÃO, C. **Histórias de Usuários.** 2020. Disponível em: <<https://cwi.com.br/blog/historias-de-usuarios/>>. Acesso em: 22 de Novembro de 2021.
- CISCATO, M.; SANTOS, A.; SUPRIANO, B.; REINA, G.; LIMA, L.; VINICIUS, M.; SIQUEIRA, L. A. V. **Me Leva Pra Casa–Adoção responsável de animais.** 2016. Disponível em: <<https://www.portalintercom.org.br/anais/sudeste2016/expocom/EX53-0516-1.pdf>>. Acesso em: 22 de Novembro de 2021.

COHN, M. **Advantages of the “As a user, I want” user story template.** 2008. Disponível em: <<https://www.mountangoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template/>>. Acesso em: 04 de Novembro de 2021.

COMPANHIA, C. A. de. **Apresentação Radar Pet 2020.** 2020. Disponível em: <<https://www.comacvet.org.br/mercado/>>. Acesso em: 30 de Março de 2021.

COODESH. **O que é Firebase?** 2021. Disponível em: <<https://coodesh.com/blog/dicionario/o-que-e-firebase/>>. Acesso em: 09 de Outubro de 2021.

CORAZZA, P. V. **Um aplicativo multiplataforma desenvolvido com flutter e NoSQL para o cálculo da probabilidade de apendicite.** 2018.

CUER, E. d. S. **Comparação de desempenho de bancos de dados sql e nosql.** 2015.

DANTAS, L. R. abordagem ao desenvolvimento móvel com flutter e dart. 2020.

EL-KASSAS, W. S.; ABDULLAH, B. A.; YOUSEF, A. H.; WAHBA, A. M. Taxonomy of cross-platform mobile applications development approaches. **Ain Shams Engineering Journal**, Elsevier, v. 8, n. 2, p. 163–190, 2017.

FIREBASE, D. **Documentação.** 2021. Disponível em: <<https://firebase.google.com/docs/>>. Acesso em: 11 de Outubro de 2021.

FIREBASE, D. **Documentação.** 2021. Disponível em: <<https://firebase.google.com/docs/firestore/>>. Acesso em: 11 de Outubro de 2021.

FIREBASE, D. **Documentação.** 2021. Disponível em: <<https://firebase.google.com/docs/auth/>>. Acesso em: 11 de Outubro de 2021.

FIREBASE, D. **Documentação.** 2021. Disponível em: <<https://firebase.google.com/docs/storage/>>. Acesso em: 11 de Outubro de 2021.

FLUTTER, D. **Documentação.** 2021. Disponível em: <<https://flutter.dev/>>. Acesso em: 05 de Outubro de 2021.

FLUTTER, D. **Documentação.** 2021. Disponível em: <<https://flutter.dev/docs/get-started/editor?tab=androidstudio>>. Acesso em: 03 de Novembro de 2021.

FLUTTER, D. **Flutter architectural overview.** 2021. Disponível em: <<https://flutter.dev/docs/resources/architectural-overview#what-is-flutter>>. Acesso em: 05 de Outubro de 2021.

FORTALEZA. **LEI Nº 11.074, DE 13 DE JANEIRO DE 2021: Modifica a Lei nº 10.214/2014, que Institui a Semana da Adoção Responsável de Cães e Gatos no Município de Fortaleza.** 2021. Disponível em: <<https://leismunicipais.com.br/a/ce/f/fortaleza/lei-ordinaria/2021/1108/11074/lei-ordinaria-n-11074-2021-modifica-a-lei-n-10214-2014-que-institui-a-semana-da-adocao-responsavel-de>>. Acesso em: 20 de Setembro de 2021.

FRANCO, M. **Além do Android e iOS: 5 sistemas diferentes para celulares.** 2021. Disponível em: <<https://www.techtudo.com.br/listas/2021/04/alem-do-android-e-ios-5-sistemas-diferentes-para-celulares.ghtml>>. Acesso em: 25 de Setembro de 2021.

- GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. [S.l.]: Plageder, 2009.
- HELM, R.; WILDT, D. **Histórias de Usuário por que e como escrever requisitos de forma ágil?** [S.l.]: Historiasdeusuario.com.br, 2014.
- JAYAKUMAR, P. **What do GitHub Stars say about “Flutter vs React Native”?** 2021. Disponível em: <<https://javascript.plainenglish.io/what-do-github-stars-say-about-flutter-vs-react-native-df4143094a82>>. Acesso em: 18 de Novembro de 2021.
- LEAL, A. M. d. M.; LEAL, A. P. *et al.* **Scholar: desenvolvimento de um aplicativo móvel genérico de apoio acadêmico a estudantes em universidades**. Florianópolis, SC, 2019.
- LIMA, W. D. Android e a influência do sistema operacional linux. **Tecnologias em Projeção**, v. 8, n. 1, p. 100–111, 2017.
- LONGO, H. E. R.; SILVA, M. P. A utilização de histórias de usuários no levantamento de requisitos ágeis para o desenvolvimento de software. **International Journal of Knowledge Engineering and Management (IJKEM)**, v. 3, n. 6, p. 1–30, 2014.
- LUETIC, M. **Cross-platform vs. native mobile development**. 2021. Disponível em: <<https://decode.agency/article/native-vs-cross-platform-mobile-apps/>>. Acesso em: 04 de Outubro de 2021.
- MADUREIRA, D. **Aplicativo nativo, web App ou aplicativo híbrido?** 2020. Disponível em: <<https://usemobile.com.br/aplicativo-nativo-web-hibrido/>>. Acesso em: 25 de Setembro de 2021.
- MAGALHÃES, Í. L. **Um estudo comparativo entre padrões arquiteturais para o desenvolvimento de aplicativos para a plataforma iOS**. Universidade Federal da Paraíba, 2018.
- MELO, R. de. **O que é Flutter?** 2019. Disponível em: <<https://www.flutterparainiciantes.com.br/o-que-e-flutter>>. Acesso em: 06 de Outubro de 2021.
- MINDS, A. **Comparison Between Hybrid Vs Native App**. 2021. Disponível em: <<https://www.angularminds.com/blog/article/comparison-between-hybrid-vs-native-app.html>>. Acesso em: 04 de Outubro de 2021.
- MOURA, A. L. T.; AMORIM, D. G. Big data: O impacto e sua funcionalidade na sociedade tecnológica. **Revista Opara**, v. 4, n. 1, 2014.
- MUNIZ, M. H.; PINTO, R. C. *et al.* Comparação de performance de processamento entre bases de dados relacionais e bases de dados nosql. **Seminários de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação**, v. 3, n. 1, 2018.
- OLIVEIRA, A. d.; LOURENÇÃO, C.; BELIZARIO, G. Índice estatístico de animais domésticos resgatados da rua vs adoção. **Revista Dimensão Acadêmica**, v. 1, n. 2, p. 5–18, 2016.
- ORLANDI, C. **Firestore: serviços, vantagens, quando utilizar e integrações**. 2018. Disponível em: <<https://blog.rocketseat.com.br/firebase/>>. Acesso em: 20 de Setembro de 2021.
- OURIQUES, J. R. Bem estar animal: Um abrigo para cães e gatos vítimas de maus-tratos e abandono em Florianópolis. **Arquitetura-Florianópolis**, 2018.

- PASSOS, P. da C. **Tô de Olho para IOS: Um Software Para Mapeamento Colaborativo de Desordens Sociais por Meio de Sistemas Georreferenciados**. 2019. Disponível em: <file:///Users/Igor/Downloads/2019\_PauloDaCunhaPassos\_tcc.pdf>. Acesso em: 25 de Setembro de 2021.
- PAULO, C. R. de Medicina Veterinária do Estado de S. **Animais de rua: um problema que demanda participação coletiva**. 2019. Disponível em: <https://www.crmvsp.gov.br/arquivo\_release/01.02.19\_Perigo\_Animais\_de\_rua.pdf>. Acesso em: 30 de Março de 2021.
- PIAZZA, L. **Prototipação: o que é, quais são os tipos e 10 Ferramentas**. 2021. Disponível em: <https://49educacao.com.br/startup/prototipacao/>. Acesso em: 04 de Novembro de 2021.
- PRADO, G. F. do; SILVA, A. F.; MEDEIROS, F. N. de; REZENDE, G. M. S. de; MARTINS, L. G. S.; OLIVEIRA, M. C. de; FERREIRA, N. de S.; VENANCIO, R. D. O. **Não abandone, adote: em defesa da adoção responsável**. 2015.
- REIS, A. C. S. d. **Um estudo comparativo entre modelos de desenvolvimento de aplicações móveis**. 2019.
- SADALAGE, P. J.; FOWLER, M. **NoSQL Essencial: Um guia conciso para o Mundo emergente da persistência poliglota**. [S.l.]: Novatec Editora, 2019.
- SCHERER, A.; LAUREANO, D. B.; ANDRADE, E. V. R. de; FERREIRA, J. de F.; BRAGHIROLI, N.; SILVA, S. G. da; MENDONÇA, R. C. *et al.* A importância da adoção de animais no brasil. **PUBVET, PUBVET**, v. 15, p. 208, 2021.
- SOUSA, J. M. **Levantamento de Requisitos – O ponto de partida do projeto de software**. 2018. Disponível em: <https://blog.cedrotech.com/levantamento-de-requisitos-o-ponto-de-partida-do-projeto-de-software>. Acesso em: 03 de Novembro de 2021.
- STRAZZI, A. **Maus tratos a animais e as leis que os protegem**. 2015. Disponível em: <https://alestrazzi.jusbrasil.com.br/artigos/252646607/maus-tratos-a-animais-e-as-leis-que-os-protegem>. Acesso em: 19 de Setembro de 2021.
- STUDIO, D. A. **Documentação**. 2021. Disponível em: <https://developer.android.com/studio/features>. Acesso em: 03 de Novembro de 2021.
- TAQTILE. **Híbrido vs Nativo**. Acesso em: 25 de Setembro de 2021.
- TRENDS, G. **Comparação Firebase e Back4app**. 2021. Acessado dia 18 de Novembro de 2021 url<https://trends.google.com.br/trends/explore?geo=BRq=firebase,back4app>.
- UNESP, R. **Adotar é um ato de amor e responsabilidade**. 2015. Disponível em: <http://reporterunesp.jor.br/2015/09/10/adotar-e-um-ato-de-amor-e-responsabilidade/>. Acesso em: 20 de Novembro de 2021.
- VALLE, D. X. **O que é um aplicativo Flutter?** 2020. Disponível em: <https://aldabra.com.br/artigo/o-que-e-um-aplicativo-flutter>. Acesso em: 20 de Novembro de 2021.