

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

HENRIQUE CAPELATTO JORDÃO

**MyDog: Um aplicativo mobile para
gerenciamento e cuidado de cachorros**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof^a. Dr^a. Renata Galante

Porto Alegre
2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^a. Patricia Helena Lucas Pranke

Pró-Reitor de Graduação: Prof^a. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Rodrigo Machado

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço a minha família. Meus pais Mário e Érika por terem me dado a vida que eu tenho hoje e sempre estarem ao meu lado. Minha esposa Marianna, por ter me dado forças, amor e carinho durante essa trajetória. Aos meus irmãos Guilherme e Natália que sempre estão ao meu lado independente de tudo. As famílias Jordão e Capelatto em especial meu vô Osvaldo e minha vó Geiza, apesar de não estarmos na mesma cidade, o amor e o apoio de vocês sempre me dão forças. A todos os meus amigos que fizeram parte dessa grande jornada, em especial a tropa da Zombie Rage. Ao corpo docente da UFRGS por me darem tamanho conhecimento que levarei para o resto da minha vida. À minha orientadora Profa. Renata Galante, por sempre ser atenciosa e solícita em responder e me apoiar nesse percurso do TCC.

Muito obrigado.

RESUMO

Dentro do Brasil existem inúmeros donos de cachorros de estimação que passam dificuldade por não conseguir gerenciar seus afazeres assim como cuidar das necessidades de seu cão. A saúde de um animal canino é tão importante quanto a de um ser humano, e dentro dos dias corridos esses compromissos com o animal podem passar despercebidos. O objetivo desse trabalho é dar uma solução para usuários através de uma aplicação para dispositivos móveis iOS para plataforma Apple. A aplicação ajudará o usuário final a conseguir gerenciar os eventos dos seus cães com facilidade de uso, com uma organização sequencial por calendário e notificações de lembrete para futuros compromissos, assim tentando melhorar a experiência do usuário com relação a outros aplicativos existentes no mercado. Experimentos realizados com usuários demonstraram que há um potencial para dentro do mercado para aplicativos de organização de afazeres dos cachorros de estimação, dado que acima de 90% dos usuários indicaram que nunca haviam utilizado um aplicativo que tem como objetivo principal facilitar a vida do usuário em relação a estas atividades.

Palavras-chave: Saúde canina. Gerenciamento. iOS.

MyDog: A mobile app for managing dog's needs

ABSTRACT

On Brazil there are various dog owners which go through difficulties due to the fact of not being able to address tasks as being able to take care of its pet necessities. The health of a canine animal is as important as that of a human being, and during fast times these commitments to the animal can go unnoticed. The purpose of this project is to provide a solution to users through an mobile iOS application for the Apple platform. The system would help the end user to manage their dog's events with ease of use, with a sequential organization by calendar and reminder notifications for upcoming appointments, thus improving the user experience compared to other existing applications on the market. Experiments carried out with users have shown that there is a potential into the market for apps for organizing the tasks of pet dogs, given that over 90% of users indicated that they had never used an app that has as its main objective to make life easier for the user. in relation to these activities.

Keywords: Canine health. user self-management. iOS..

LISTA DE FIGURAS

Figura 2.1 IDE - Xcode.....	16
Figura 3.1 O aplicativo MeuPet	24
Figura 3.2 O aplicativo Things3.....	25
Figura 3.3 O aplicativo PetNotes	26
Figura 3.4 O aplicativo MyTrackPet.....	27
Figura 4.1 Elementos do sistema	30
Figura 4.2 Conceito original do MyDog.....	31
Figura 5.1 Versionamento do código via GitKraken.....	35
Figura 5.2 Organização de tarefas via Notion.....	36
Figura 5.3 Alterações da modelagem.....	38
Figura 6.1 Passos 1 e 2 do onboarding.....	40
Figura 6.2 Passos 3 e 4 do onboarding.....	40
Figura 6.3 Passos 5 e 6 do onboarding.....	41
Figura 6.4 Tela de Home	42
Figura 6.5 Tela de meus cachorros.....	43
Figura 6.6 Botão adicionar novo cachorro	43
Figura 6.7 Tela de detalhe de cachorro	44
Figura 6.8 Interação gráfico e listagem de eventos	45
Figura 6.9 Tela de eventos.....	46
Figura 6.10 Tela de detalhe de evento (medicamento).....	47
Figura 6.11 Notificação de evento.....	48
Figura 7.1 Resultado (Escolaridade e Faixa etária)	50
Figura 7.2 Resultado (número de cachorros de estimação)	51
Figura 7.3 Resultado (Experiência em dispositivos móveis e internet)	52
Figura 7.4 Resultado (Experiência em aplicativos de organização e cuidado de animais).....	53
Figura 7.5 Resultado (Usabilidade cadastro, evento, expectativas organização).....	54
Figura 7.6 Resultado (Usabilidade cadastro, evento, expectativas)	55

LISTA DE TABELAS

Tabela 3.1 Plataformas onde as aplicações existem.....	27
Tabela 3.2 Funcionalidades por plataforma	28

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
GUI	<i>Graphical User Interface</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IPA	<i>iOS App Store Package</i>
UI	<i>User Interface</i>
UX	<i>User Experience</i>
XML	<i>Extensible Markup Language</i>
WWDC	<i>Apple Worldwide Developers Conference</i>

SUMÁRIO

1 INTRODUÇÃO	11
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 <i>Saúde/Bem-estar Canino</i>	13
2.2 <i>A plataforma iOS</i>	14
2.3 <i>Ferramentas</i>	15
2.3.1 XCode	15
2.3.2 Git	16
2.3.3 Notion	17
2.4 <i>Arquitetura de Software</i>	17
2.4.1 The Composable Architecture	17
2.5 <i>Padrões de projeto de software</i>	18
2.5.1 Factory	18
2.5.2 Singleton	18
2.6 <i>Metodologia de Desenvolvimento</i>	18
2.7 <i>Frameworks</i>	19
2.7.1 Frameworks nativos	19
2.7.1.1 Combine	19
2.7.1.2 Foundation	20
2.7.1.3 SwiftUI	20
2.7.2 Frameworks terceiros	21
2.7.2.1 Keychain Swift	21
2.7.2.2 The Composable Architecture	21
2.8 <i>Armazenamento de Dados</i>	22
2.9 <i>Considerações finais</i>	22
3 TRABALHOS RELACIONADOS	23
3.1 <i>Descrição de Trabalhos</i>	23
3.1.1 MeuPet	23
3.1.2 Things3	24
3.1.3 PetNotes	25
3.1.4 MyTrackPet	26
3.2 <i>Análise comparativa</i>	27
4 PROJETO MYDOG	29
4.1 <i>Elementos do sistema</i>	29
4.2 <i>Modelagem</i>	30
4.3 <i>Histórias de usuário</i>	32
5 IMPLEMENTAÇÃO	34
5.1 <i>Ambiente de desenvolvimento</i>	34
5.1.1 Hardware	34
5.2 <i>Organização de tarefas</i>	35
5.3 <i>Frameworks utilizados</i>	36
5.4 <i>Alterações na modelagem</i>	37
6 APLICAÇÃO	39
6.1 <i>Onboarding</i>	39
6.2 <i>Home</i>	41
6.3 <i>Meus cachorros</i>	42
6.4 <i>Detalhe de cachorro</i>	44
6.5 <i>Eventos</i>	45
6.6 <i>Detalhe de evento</i>	46

6.7 Notificação de evento	47
7 EXPERIMENTO DE USABILIDADE	49
7.1 Protocolo	49
7.2 Participantes.....	50
7.3 Resultados.....	50
7.4 Considerações Finais	55
8 CONCLUSÃO	56
9 APÊNDICE A - FORMULÁRIO	57
REFERÊNCIAS.....	61

1 INTRODUÇÃO

Animais de estimação fazem parte da vida da população mundial desde 13,000-15,000 anos atrás através da primeira domesticação histórica descoberta de lobos no fim do período glacial (BENECKE, 1987). Desde então, a relação com os animais se tornou cada vez mais importante dentro da vida do ser humano. Dentre esses animais, o cachorro, se tornou um grande fator social/familiar. Atualmente, existem cerca de 471 milhões de cachorros de estimação (STATISTA, 2018).

O Brasil é o terceiro maior país em população total de animais de estimação, tendo cerca 54,2 milhões de cachorro, 23,9 milhões de gatos, 19,1 milhões de peixes, 39,8 milhões de aves e mais 2,3 milhões de outros animais . No total tendo 139,3 milhões de cachorros, o que demonstra a importância desses animais dentro da vida do brasileiro (ABINPET, 2018). Levando em consideração a demografia, os cachorros, que são o tópico principal desse trabalho de conclusão de curso, estão presentes em 46,1% das residências domiciliares brasileiras (IBGE, 2019).

A fragilidade de vida de um animal canino é tão parecida quanto a de um ser humano. Existem várias áreas de atenção para manter o animal saudável e bem cuidado. Há a necessidade de cuidar de sua saúde com exames, ida a médicos, vacinas, controles parasitários, alergias, banhos, entre outros. A vida de um dono de cachorro pode ser bem conturbada com seus próprios afazeres, o que pode levar a esquecimento ou despriorização dos compromissos com os bichos.

Este trabalho de conclusão de curso tem o propósito de alavancar uma solução dentro da área de tecnologia de dispositivo móvel iOS na plataforma Apple com uma usabilidade melhor do que a competição no mercado já existente, trazendo uma melhor experiência para o usuário. O aplicativo auxilia o dono a cuidar das necessidades de seu animal canino, através de ferramentas como calendários, organizadores e notificações de todos os eventos de seus bichos de estimação. Este projeto foi denominado como MyDog. O público alvo dessa aplicação são donos de cachorro que passam por complexidade em gerenciamento do seu tempo e dificuldade de lembrar seus compromissos com seus cachorros. Experimentos com usuários realizados comprovam que os mesmos veem necessidade e utilidade em aplicativos como o apresentado neste trabalho.

A motivação para esta proposta em específico se deu pelo autor ter uma aproximação muito forte com bichos de estimação e por ter sentido as dificuldades que foram descritas acima como dono de um animal canino. Essa motivação levou a criação desta

ferramenta que poderá auxiliar futuros usuários. O conceito do projeto já havia sido discutido e idealizado com grupos dentro Apple Developer Academy entretanto o projeto nunca foi desenvolvido.

O texto está estruturado de maneira a guiar o leitor através da jornada percorrida pelo autor no desenvolvimento deste trabalho. O Capítulo 2 apresenta os conceitos teóricos que fundamentaram a criação do protótipo e as tomadas de decisão necessárias para a definição das funcionalidades a serem implementadas. Em seguida, no Capítulo 3, são apresentadas ferramentas com objetivos similares já existentes no mercado. Estas soluções são então comparadas com a versão da ferramenta obtida com este trabalho.

Em seguida, o Capítulo 4 dá um panorama sobre o contexto em que o aplicativo iOS desenvolvido neste trabalho se encontra dentro do projeto MyDog, descrevendo os elementos existentes e os esforços necessários para a conclusão deste trabalho. Com este plano de fundo, o Capítulo 5 descreve os detalhes da implementação do MyDog, e em seguida, no Capítulo 6, é apresentado a ferramenta resultante.

Então, no Capítulo 7, são descritos os experimentos feitos com usuários a fim de avaliar a usabilidade do MyDog. Os resultados dos experimentos são então analisados, e em seguida, no Capítulo 8, são apresentados as conclusões às quais se chegou no término deste trabalho e apontamentos para trabalhos futuros

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as tecnologias (técnicas e ferramentas) utilizadas durante o desenvolvimento do trabalho. No início, apresenta-se um panorama sobre o tema de saúde/bem-estar canina. Após isso, são explicados conceitos teóricos de computação em conjunto aos do tema para desenvolvimento do aplicativo MyDog. Além disso, o capítulo fala sobre ferramentas que auxiliaram ou foram integradas na solução final. O conteúdo desta parte se encerra com algumas considerações finais.

2.1 Saúde/Bem-estar Canino

A saúde canina é tão frágil quanto a humana. Nesta seção menciona-se alguns aspectos da vida canina que são importantes para manter o cachorro saudável, isto dá introdução ao trabalho na questão de indicadores do que é importante ou não para um dono ter em seu aplicativo de lembrete.

- *Vacinas*: A vacinação é uma chave importante para a convivência harmônica entre humanos e cachorros, elas são divididas entre essenciais ou não essenciais. A determinação se um vacina é vital para aplicação ocorre através de risco de exposição, severidade da doença e transmissibilidade a seres humanos. Dentre as vacinas mais importantes se encontram parvovírus canino, cinomose, hepatite canina e raiva. É portanto, de suma importância manter e seguir as datas de vacinação do animal de estimação (ASPCA, 2021).
- *Controle Parasítico*: Animais são suscetíveis a parasitas, isso ocorre devido ao comportamento do animal, área de onde vive, condições higiênicas do cachorro e contato com outros animais. Os parasitas se dividem em dois grupos: internos e externos. Os internos são parasitas que vivem dentro do corpo do animal, tais quais vermes pulmonares, do coração, redondos, em formato de ganchos ou de chicotes. O outro grupo de parasitas são os externos são relacionados a pele do animal tais como pulgas, carrapatos, sarna demodécica, sarna sarcóptica, entre outros. Os tratamentos de vermes são através de medicamentos e normalmente é necessário aplicações em intervalos de tempo (LOPES, 2021).
- *Consultas preventivas*: Consultas preventivas são significativas ferramentas de identificação de doenças e tratamentos da forma mais rápida possível. Entretanto,

donos de cachorros não possuem o hábito de realizar idas aos veterinários. Ainda há cerca de 34% dos donos de cachorros que levam seus animais menos de uma vez ao ano ou sem frequência (SANT'ANA, 2020).

- *Higiene*: Banhos, ambientes limpos, tosa, higiene bucal são importantes para manter o animal de estimação saudável. Além disso, manter o animal higienizado previne o bicho de carregar microorganismos que podem provocar doenças em seres humanos e outros animais. Devido a isso, manter uma rotina de higienização do cachorro é importante para uma qualidade de vida melhor ao animal e a todos ao redor (PAIVA, 2018).
- *Temperatura*: A temperatura natural de um cachorro é entorno de 38° - 39° graus célcus independente da raça e tamanho do cachorro, qualquer oscilação fora do padrão pode ser indicador de algum problema de saúde do animal (PETZ, 2019).
- *Peso*: Obesidade canina é um grande problema que vem ocorrendo nos últimos anos. O sobrepeso pode ser o início de outras doenças ou condições se desenvolvendo no animal. Um estudo feito pela Associação para a Prevenção da Obesidade Pets estima que 44% dos cachorros americanos estavam em sobrepeso ou obesos, não existe um estudo atual que reflete isso para o contexto brasileiro, entretanto podemos afirmar com esse estudo que é uma preocupação que tem chances de ocorrência de forma mais frequente. A observação do peso pode facilitar ao dono a tomada de decisão de iniciar um controle de peso mais rígido ao seu pet (HOFMEISTER, 2019).

Resumidamente, é visível que uma plataforma que permite o usuário a ter facilidade no controle das necessidades de seu animal canino auxiliaria na qualidade de vida de ambos dono e bicho de estimação. Para resolver este problema será utilizado um aplicativo iOS.

2.2 A plataforma iOS

A plataforma iOS (APPLE, 2021h) é o sistema operacional desenvolvido pela empresa Apple. Ela foi criada para dispositivos móveis do tipo iPhone. Originou-se em 2007 com o nome de iPhoneOS por Steve Jobs na conferência denominada Macworld. Desde então, todo ano ocorre atualizações do software, atualmente a versão mais nova é o iOS 15 no último anuncio da WWDC dia 7 de Julho de 2021 (APPLE, 2021d).

iOS é o nome dado ao sistema operacional desenvolvido pela Apple exclusivamente para seus aparelhos móveis (iPhone, iPod Touch e iPad). Foi lançado em 2007, juntamente com o lançamento do primeiro iPhone, sob o nome de iPhone OS. Após isso, novas versões do sistema operacional foram lançadas juntamente com o lançamento anual e modelos de iPhone. A partir da versão 4, o sistema deixou de se chamar iPhone OS, sendo abreviado para iOS. Atualmente, a versão atual é a 13, lançada em 3 de junho de 2019, na conferência anual de desenvolvedores da Apple (*Apple Worldwide Developers Conference*, ou WWDC).

O desenvolvimento de sistema Apple para iOS se dá por hardwares que possuem o sistema operacional da Apple, no caso o macOS (APPLE, 2021i). Não é possível compilar a aplicação em outros tipos de sistema. A Apple recomenda a utilização da IDE Xcode que possui todas as ferramentas para início e finalização da geração do aplicativo. Esse software pode ser desenvolvido sobre 2 linguagens de programação:

- **Swift:** Uma linguagem moderna com uma definição de linguagem de programação orientada a protocolos, lançada em 2014, seu crescimento e popularidade tem crescido progressivamente. (APPLE, 2021a)
- **Objective-C:** Essa linguagem de programação orientada a objetos tem sua origem em no início 1980. Se tornou mais conhecida quando a NeXT comprou a licença da StepStone. Após esses acontecimentos a Apple comprou a licença em 1996 e declarou a mesma como linguagem oficial de seus produtos. (APPLE, 2014)

2.3 Ferramentas

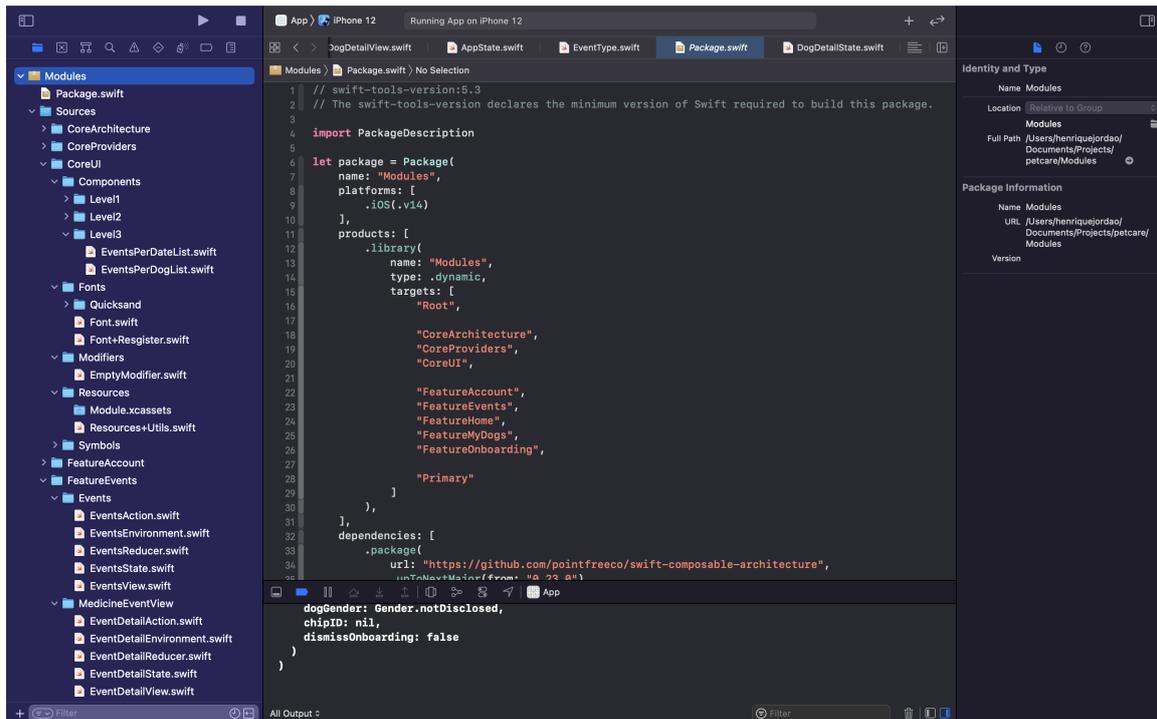
Nesta seção são apresentadas as ferramentas utilizadas para desenvolvimento da aplicação MyDog.

2.3.1 XCode

O Xcode (APPLE, 2021i) é uma ferramenta disponibilizada pela Apple para desenvolvimento de aplicações de seus sistemas operacionais (macOS, iOS, iPadOS, watchOS e tvOS). Essa ferramenta é denominada de IDE. Essa plataforma prove todos os instrumentos para desenvolvimento, e isso vai da parte inicial de estruturação de código, visualização das telas através de previews, compilação de código e debug após produção.

A figura 2.1, apresenta a interface gráfica da IDE para compilação do código que gerou a aplicação MyDog. Além de ferramentas para programação, também é disponibilizado a parte de distribuição, que se resume a geração de uma arquivo IPA, que é o tipo de arquivo para envio para loja de aplicativos da Apple, conhecido como App Store (APPLE, 2021e).

Figura 2.1 – IDE - Xcode



2.3.2 Git

A criação de um software se da em sua grande maioria através de times de desenvolvimento, como o programa nada mais é do que arquivos de texto que passa através de um compilador para gerar um binário, as alterações em mesmos arquivos geram dificuldades para a equipe envolvida. Para solucionar o problema é utilizado o versionamento de código, que gerencia versões de código através de commits que são estados do código em um momento de tempo. Essa abordagem permite ao grupo a ter retroatividade de código, uma visão da evolução do mesmo e também como mencionado anteriormente auxiliar na resolução de possíveis conflitos durante o processo de desenvolvimento.

Para esse trabalho foi decidido utilizar o git que é atualmente o sistema de versionamento de código mais utilizado globalmente passando em 2014 o Apache Subversion, segundo pesquisa descrita em (SKERRETT, 2014). Seu uso é gratuito e possui código

aberto. A ferramenta utilizada para ajudar nessas operações foi o GitKraken (AXOSOFT, 2016) que é uma interface gráfica de usuário GUI.

2.3.3 Notion

O Notion (LABS, 2018) é uma aplicação que prove componentes como notas, banco de dados, calendários, lembretes, entre outras funcionalidades. A junção dessas funcionalidades menores permite ao usuário a criar sistemas funcionais para gerenciamento de suas necessidades. O autor desse projeto utilizou essa plataforma para fazer o gerenciamento visual de seu projeto através de um estilo Kanban (ESPINHA, 2019) em questões de suas tarefas de desenvolvimento devido prévia experiência com a aplicação.

2.4 Arquitetura de Software

A arquitetura de software utilizada nesse projeto é feita através de uma biblioteca de terceiros destinada a desenvolvimento de aplicativos de celular na plataforma iOS chamada The Composable Architecture (POINTFREE, 2020). A base teórica da estrutura do projeto em iOS é uma arquitetura em camadas onde cada módulo tem suas responsabilidades e expõe uma determinada API para os módulos que são utilizados.

2.4.1 The Composable Architecture

The Composable Architecture (POINTFREE, 2020) é uma arquitetura voltada para implementação de aplicativos iOS. Essa arquitetura é construída em cima das bibliotecas nativas Combine e SwiftUI que serão abordadas na seção de frameworks. A base segue os seguintes princípios:

- Gerenciamento de estados: A arquitetura permite uma declaração explícita de como o sistema funciona em cada etapa com as ações do usuário e com chamadas externas.
- Componentização: Quebra de funcionalidades em pequenos componentes isolados e reaproveitáveis.

- Efeitos Colaterais: Como certas partes do sistema interagem com as suas chamadas externas.

2.5 Padrões de projeto de software

Os padrões de projeto foram propostos por Erich Gamma (GAMMA et al., 1995) tem como finalidade tornar mais fácil a reutilização de projetos e arquiteturas. Dentre os vários tipos de padrões existentes, os seguintes foram utilizados nesse trabalho.

2.5.1 Factory

O padrão de projeto Factory (também conhecido por *Virtual Constructor*) é designado por encapsular a criação de objeto. O padrão *Factory* fornece uma interface para a criação de famílias de objetos correlatos ou dependentes sem a necessidade de especificar a classe concreta destes objetos. Assim, permitindo fácil substituição de classes a medida que o código evolui sem necessidade de reescrever código de outros objetos que utilizam o mesmo.

2.5.2 Singleton

O padrão de projeto *Singleton* tem o propósito de garantir que só exista uma instância e que o acesso da mesma só pode ser feito apenas por um local. Assim, garantindo que este objeto único fique disponível para toda a aplicação e que não seja possível instanciá-lo novamente. Este tipo de padrão é útil pois dá ao sistema controle em acesso compartilhados a recursos que serão usados em diferentes módulos da aplicação.

2.6 Metodologia de Desenvolvimento

A metodologia utilizada nesse projeto foi baseada no modelo Cascata (PRESSMAN, 1995), pois aborda o problema de forma sistemática e sequencial com flexibilidade para o desenvolvimento de software e isso se enquadra bem quando temos requisitos bem definidos e estáveis. Esse modelo é separado em 5 etapas:

- *Requisitos e Análise*: Definição de funcionalidades, regras de negócio, restrições e principal objetivo do sistema são definidos através de consultas com o usuário do sistema.
- *Projeto*: Definição de padrões de software, arquitetura do sistema, bibliotecas externas, regras de API que serão seguidas e identificação e especificação dos componentes e como interagem entre si.
- *Implementação*: Programação dos subsistemas em prol do projeto completo. Definição de pronto para verificar se o conjunto de subsistemas está atendendo suas especificações, a saída dessa etapa ocorre quando os componentes estão finalizados.
- *Integração e testes*: Integração de todos os subsistemas para validação do sistema. Garantir que os requisitos da especificação estão cumpridos. O sistema após esta etapa é entregue ao cliente final, no caso enviado a App Store.
- *Manutenção*: Correção de erros e evolução do projeto a medida que novos requisitos são identificados.

2.7 Frameworks

Framework traduz para bibliotecas que são utilizadas dentro de um sistema que dão ao programador ferramentas através de classes e funções que podem ser utilizadas dentro de seu ambiente. Essas bibliotecas, no contexto de desenvolvimento iOS, podem ser nativas que são frameworks desenvolvidos pela equipe da Apple quanto externas que são que são desenvolvidos por grupos fora desse ecossistema. O objetivo da utilização dessas ferramentas é a aceleração de codificação do programa além de criar um sistema padronizado aos costumes do mundo de desenvolvimento mobile iOS. A seguir, estão descritos os frameworks que foram utilizados no projeto;

2.7.1 Frameworks nativos

2.7.1.1 Combine

O Combine (APPLE, 2021f) é uma biblioteca nativa voltada a programação reativa inspirada no sucesso que foi RxSwift (X, 2015). Ela prove uma API declarativa para processar valores ao longo do tempo, isso permite declarar eventos de UX do usuário,

respostas de camada de rede, eventos em geral que são assíncronos através de observadores sobre os fluxos de dados. Esse paradigma de programação possui em vantagens nos seguintes pontos:

- Evitar inúmeros *callbacks* dentro do código, evitando código caótico.
- Facilitar tratamento de código assíncrono.
- Bibliotecas provem operadores que simplificam o trabalho.
- Fácil criação de fluxos de dados.

RxSwift veio de origem das extensões de ReactiveX (REACTIVEX, 2011) originadas pela Microsoft em 2011. Esse paradigma se tornou tão popular que se estendeu para várias bases de códigos com suas próprias bibliotecas tais como: RxJS, RxPy, RxJava, entre outros.

2.7.1.2 Foundation

O Foundation (APPLE, 2021g) é o framework primário de desenvolvimento mobile iOS. Através dele é possível acessar uma série de funções base e acesso ao sistema operacional que são necessárias para início da codificação da aplicação.

2.7.1.3 SwiftUI

O SwiftUI (APPLE, 2021j) é uma nova biblioteca nativa lançada em 2019 para facilitar a criação de interfaces de usuário de forma declarativa. Este formato de desenvolvimento facilita a separação de componentes visuais que podem ser reaproveitados no código, devido a redução de código para criação de um elemento visual comparado a antiga forma imperativa. Essa biblioteca atua muito bem em conjunto com o framework de programação reativa Combine. A definição dos dois tipos de formas (MUEMA, 2021):

- Forma Declarativa: Padrão que permite os desenvolvedores a projetar interfaces de usuário baseado nos dados recebidos. Foca no *o que* a interface deve mostrar para o cliente ao contrário de como é construída. Um exemplo comparativo pode ser a forma de declaração em websites no formato HTML.
- Forma Imperativa: Padrão que foca no como a interface de usuário deve se comportar. Este é o padrão mais comum de desenvolvimento de UI devido a forma declarativa ter se tornado mais popular nos últimos anos. Um exemplo comparativo pode ser a forma de declaração em websites no formato XML.

A biblioteca também fornece em conjunto a IDE ferramentas de visualização de teste (APPLE, 2020) que permite a prototipação da UI de forma muito mais acelerada, elas podem ser acessadas através da criação de previews sobre as janelas de SwiftUI.

2.7.2 Frameworks terceiros

2.7.2.1 Keychain Swift

O Keychain Swift (EVGENII, 2015) é uma biblioteca de terceiros open source que prove ao usuário funções utilitárias que facilitam o acesso ao sistema de Keychain da Apple (APPLE, 2015). Esse sistema prove uma forma segura e criptografada de salvar dados de usuário dentro do banco de dados do sistema operacional que a aplicação está rodando.

2.7.2.2 The Composable Architecture

O TCA é uma biblioteca de terceiros criada pela organização Point-Free em 2020 (POINTFREE, 2018) que evoluiu para um projeto em open-source. Ela prove uma série de ferramentas fundamentais que permitem a construção de uma arquitetura focada em reaproveitamento de componentes e com facilidade de testes. Esse framework atua através de 5 elementos base:

- Estado: Um tipo que descreve os dados que o subsistema precisa para executar sua lógica e exibir sua interface de usuário.
- Ação: Um tipo que define todos os tipos de ação que podem ocorrer dentro daquele subsistema, tais como: ações de usuário, notificações, chamadas assíncronas, entre outros.
- Ambiente: Um tipo que possui todas as dependências que o subsistema precisa, tais como: serviços de rede, serviços de armazenamento de dados, serviços de manipulação de dados internos compartilhados como manipulação de dados de usuário.
- Redutor: Uma função que descreve como evoluir do estado atual para o próximo estado dado que ocorreu uma ação. Essa função também é responsável por tratar e retornar qualquer efeito colateral que deva ser executado, tais como: chamadas de api, atualizar dados de usuário interno, entre outros.

- Loja: Um tipo que recebe todas as ações do usuário e permite o envio para a função de redução. Esse tipo atua como uma ponte entre a UI aplicação e o redutor.

2.8 Armazenamento de Dados

O sistema tem necessidade de armazenar dados para execução de suas regras de negócio com o intuito de cumprir os requisitos da aplicação. O aplicativo MyPet não possui necessidade de armazenar os dados em banco de dados ou em uma estrutura de webservice dado que o objetivo é uma manipulação local das informações, por isso os dados são persistidos através do Keychain das plataformas da Apple para carregar e alterar os dados do cliente.

As informações salvas se relacionam ao ambiente de cachorros de estimação como datas, fotos, informações sobre eventos do animal, informações sobre os animais, informações do usuário, horários assim como informações sobre veterinários.

2.9 Considerações finais

Neste capítulo do trabalho foram mostrados os conceitos base para construção do MyPet. Essas ferramentas e termos apresentados serão utilizados para o desenvolvimento do sistema final em prol de atingir o objetivo em que o público alvo consiga suprir as necessidades e compromissos de seus cachorros de estimação de forma organizada e assim melhorando a qualidade de vida dos mesmos.

Esse objetivo acima será cumprido através da aplicação mobile com arquitetura TCA, programação reativa com Combine, interface de usuário declarativa com SwiftUI e com persistência de dados através do Keychain das plataformas Apple.

3 TRABALHOS RELACIONADOS

A vida cotidiana de um adulto que trabalha de forma integral 5 dias por semana, todas as semanas já é bem corrida, aplicações que permitem pessoas a otimizar seu tempo já foram criadas. Neste capítulo, primeiramente, é dada uma visão sobre outras aplicações, relacionadas ao caráter de organização de afazeres não estreitamente ligado a organização de cachorro de estimação. Posteriormente, faremos uma análise comparativa sobre as aplicações abordadas e a solução criada neste trabalho.

3.1 Descrição de Trabalhos

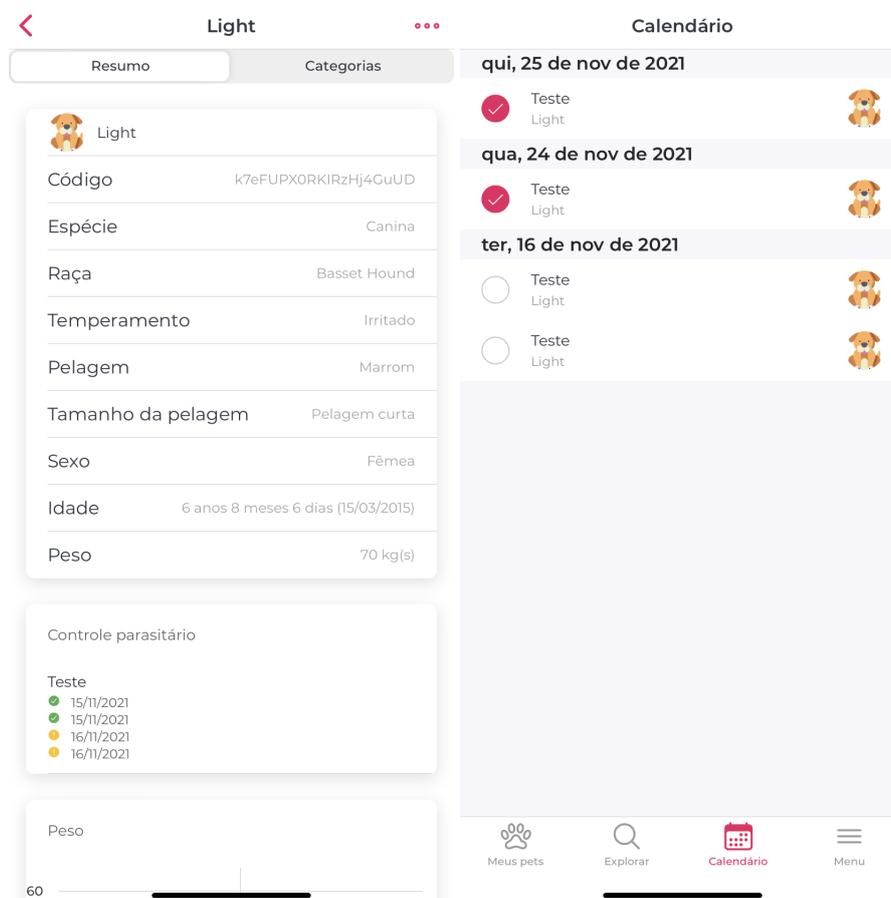
Nesta seção serão discutidos as seguintes aplicações: MyTrackPet, MeuPet, Pet Notes, Things3.

3.1.1 MeuPet

A MeuPet (PETS, 2021) é uma aplicação nova para plataformas iOS (Figura 3.1) e Android (APPLE, 2021b). Nessa aplicação o usuário pode: criar uma conta, registrar cachorros de estimação, adicionar múltiplos tipos de eventos de cuidados de saúde, acompanhar eventos pelo calendário, ter um acompanhamento do peso e temperatura do seu animal.

Essa plataforma surgiu em 2021 na plataforma iOS e já possui múltiplas atualizações de melhorias na performance no app. Comparado com as outras aplicações no mercado em relação a cuidados de animais de estimação é a que está mais a frente em questão de experiência de usuário com uma avaliação de 4.5 de 5 estrelas dentro da AppStore.

Figura 3.1 – O aplicativo MeuPet



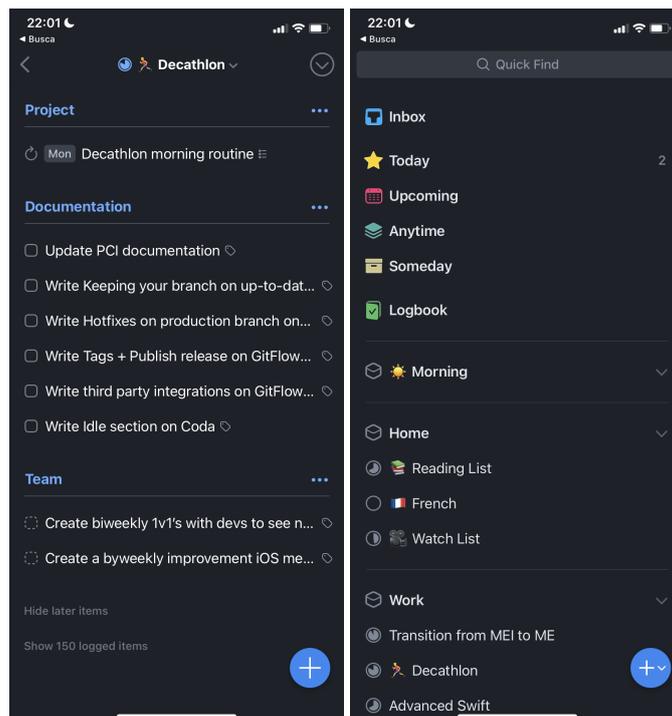
3.1.2 Things3

Things3 (APPLE, 2021c) é uma aplicação desenvolvida para plataformas da Apple, tendo suporte os seguintes sistemas operacionais: iOS (Figura 3.2), iPadOS, macOS e watchOS. Não é uma aplicação voltada a especialidade da área de cuidado animal, entretanto é um organizador de afazeres de agenda que é relevante para uma comparação em relação ao sistema desenvolvido nesse trabalho. Similar a aplicações como todoist (APPLE, 2007) e OmniFocus3 (APPLE, 2008), o Things permite o usuário separar seus afazeres em projetos e tarefas simples de resolução com um acompanhamento de notificações e calendário.

A plataforma foi criada pelo grupo Cultured Code em 2012 (CODE, 2012) e vem em constante desenvolvimento com seu foco primário aos sistemas operacionais da Apple. Possui uma alta nota de satisfação com o cliente com uma avaliação de 4.8 de 5 estrelas

dentro da AppStore.

Figura 3.2 – O aplicativo Things3

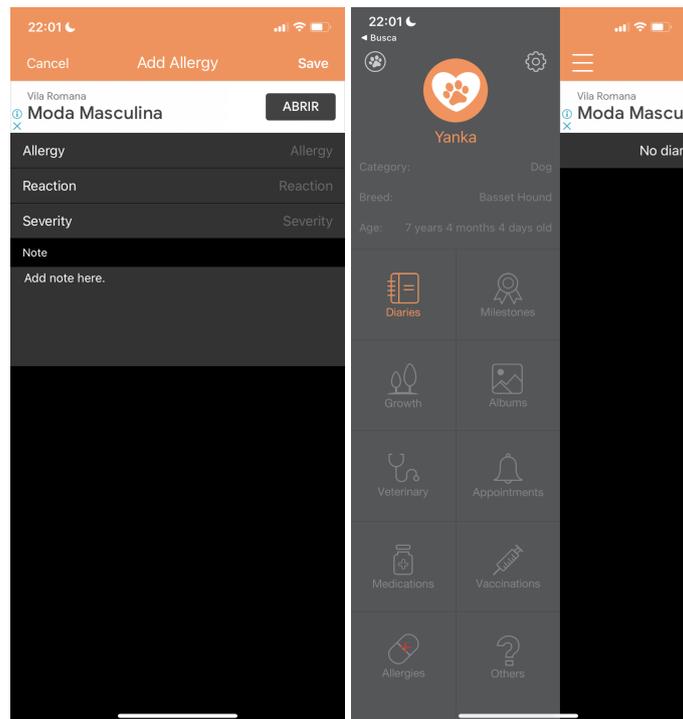


3.1.3 PetNotes

Pet Notes é uma aplicação desenvolvida para plataforma iOS (Figura 3.3). Nessa aplicação o usuário tem acesso as seguintes funcionalidades: registrar cachorros, um diário de acompanhamento do seu animal, acompanhamento de estatísticas de peso e temperatura, um espaço para armazenamento de fotos do seu cachorro, registro de múltiplos tipos de eventos de cuidados a saúde, assim como um acompanhamento desses eventos por um calendário e um sistema de exportação de dados através da geração de um arquivo de pdf.

Esta plataforma foi criada em 2017 pelo grupo The Link Links com 5 versões de atualização, o projeto apesar de disponível na AppStore não possui um indício claro de satisfação do cliente devido ao fato de não possuir um número suficiente de avaliações pelo cliente.

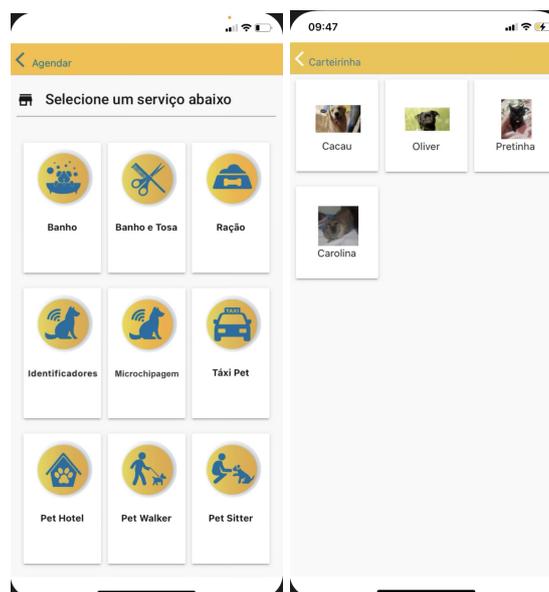
Figura 3.3 – O aplicativo PetNotes



3.1.4 MyTrackPet

My Track Pet (APPLE, 2018) é uma plataforma que provem serviços de compra de produtos relacionados a cachorros, assim como, uma área ao cliente que gostaria de organizar a saúde de seu cachorro, as interfaces existentes para esse produto são iOS (Figura 3.4), Android e web (PET, 2017). Dentre os serviços oferecidos ao cuidados do animal são: cadastro de animal, registro de vacinas aplicadas, registro de vermífugos utilizados e registro de ocorrências em caso de perda ou roubo. Essa plataforma veio para o iOS em 2018 e até hoje recebeu apenas 3 atualizações com as últimas 2 sendo as mais recentes em Setembro de 2021. De acordo com a empresa a plataforma web está em crescimento e é esperado que essa melhoria se transfira para as outras plataformas.

Figura 3.4 – O aplicativo MyTrackPet



3.2 Análise comparativa

Nesta seção, é feita uma análise comparativa das versões iOS de todas as aplicações mencionadas anteriormente em conjunto com o sistema final criado pelo autor. Esta comparação será feita através de tabelas com os aspectos de cada aplicação sendo registrado no momento de escrita desse trabalho (Novembro, 2021).

A tabela 3.1 a seguir apresenta a área de atuação de cada uma das plataformas. Todas são utilizadas dentro de iOS, entretanto, algumas são mais distribuídas em outras regiões enquanto outras são especializadas para apenas um público alvo. A aplicação desse trabalho apenas possui um sistema dentro da área de iOS.

Tabela 3.1 – Plataformas onde as aplicações existem

Plataforma	MeuPet	Things3	PetNotes	MyTrackPet	MyDog
Web	Não	Não	Não	Sim	Não
iOS	Sim	Sim	Sim	Sim	Sim
Android	Sim	Não	Não	Sim	Não
Desktop	Não	Sim	Não	Não	Não

A seguir a tabela 3.2 mostra, em questões de funcionalidade, quais aplicações possuem ou não certos aspectos na área de organização, alertas e cuidados de saúde de animais. É visível que, tirando a aplicação Things que é para organizações em gerais, todas as aplicações apresentadas possuem funcionalidades em comum como registro de

cachorros, registro de controle parasitário, medicamento e vacina. No entanto, tanto os sistemas PetNotes e MyTrackPet não oferecem tanta especificação em seus eventos que são registrados. É visível que a aplicação MeuPet possui muitas similaridades em questão funcional ao aplicativo MyDog.

Tabela 3.2 – Funcionalidades por plataforma

	MeuPet	Things3	PetNotes	MyTrackPet	MyDog
<i>Cadastrar múltiplos cachorros</i>	Sim	Não	Sim	Sim	Não
<i>Organizar agenda por data</i>	Sim	Sim	Não	Não	Sim
<i>Organizar agenda por cachorro</i>	Não	Não	Não	Não	Sim
<i>Resumo de eventos por cachorro</i>	Sim	Não	Sim	Sim	Sim
<i>Gráficos temperatura/peso do cachorro</i>	Sim	Não	Não	Sim	Sim
<i>Adicionar alergias ao cachorro</i>	Sim	Não	Sim	Não	Sim
<i>Adicionar cirurgia</i>	Sim	Não	Não	Não	Sim
<i>Adicionar controle parasitário</i>	Sim	Não	Sim	Sim	Sim
<i>Adicionar exame</i>	Sim	Não	Sim	Não	Sim
<i>Adicionar evento de higiene</i>	Sim	Não	Não	Sim	Sim
<i>Adicionar lembrete</i>	Sim	Sim	Não	Não	Sim
<i>Adicionar medicamento</i>	Sim	Não	Sim	Sim	Sim
<i>Adicionar vacina</i>	Sim	Não	Sim	Sim	Sim
<i>Notificar próximo eventos</i>	Sim	Sim	Não	Não	Sim
<i>Conversar com estabelecimentos de cachorro</i>	Sim	Não	Sim	Não	Não
<i>Registrar animal perdido ou roubado</i>	Não	Não	Não	Sim	Não

4 PROJETO MYDOG

A aplicação MyDog é um projeto com a ideia originada dentro do Apple Developer Academy (ELDORADO, 2010). Inicialmente, a proposta era um sistema que atuasse sobre todos os tipos mais comuns de animais de estimação dentro da residência de uma família brasileira. Devido ao tamanho do escopo do projeto, foi decidido pelo autor em apenas atuar em uma parte especializada que são cachorros. O núcleo da aplicação é fornecer ao usuário ferramentas de organização, rastreamento de afazeres relacionado ao seu cachorro de estimação assim como funcionalidades específicas a cachorros de estimação. Esta seção relata um escopo geral sobre a conceitualização do projeto antes da parte de implementação da aplicação, portanto, escolhas/ferramentas apresentadas aqui podem ter sido alteradas durante o processo de desenvolvimento.

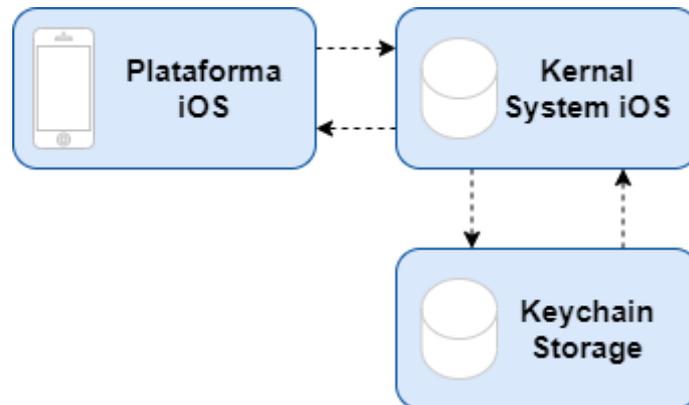
4.1 Elementos do sistema

Construído dentro de 2 grandes tópicos. Eles juntos fazem a composição da aplicação como um todo

- Aplicativo iOS: Aplicação para dispositivos móveis com sistema iOS da plataforma Apple. Desenvolvimento se deu dentro desse trabalho.
- Persistência de dados: Estrutura de dados salvo internamente dentro do celular do usuário. Construção dos modelos e gerenciamento das estruturas feito dentro deste trabalho. Apenas visível através da interface pelas injeções de dependências do projeto.

A aplicação MyDog faz pedidos ao kernel do sistema iOS para carregar/salvar/deletar as estruturas que são utilizadas. Esses dados ficam salvos dentro de um banco de dados criptografado através de uma biblioteca chamada Keychain do dispositivo móvel e não são sensíveis a desinstalação da aplicação por parte do usuário, caso a aplicação seja excluída, esses dados ficarão salvos dentro do dispositivo do mesmo até os dados se tornarem inválidos pelo sistema operacional ou a aplicação ser reinstalada. A figura 4.1 demonstra a interação entre os elementos do sistema.

Figura 4.1 – Elementos do sistema



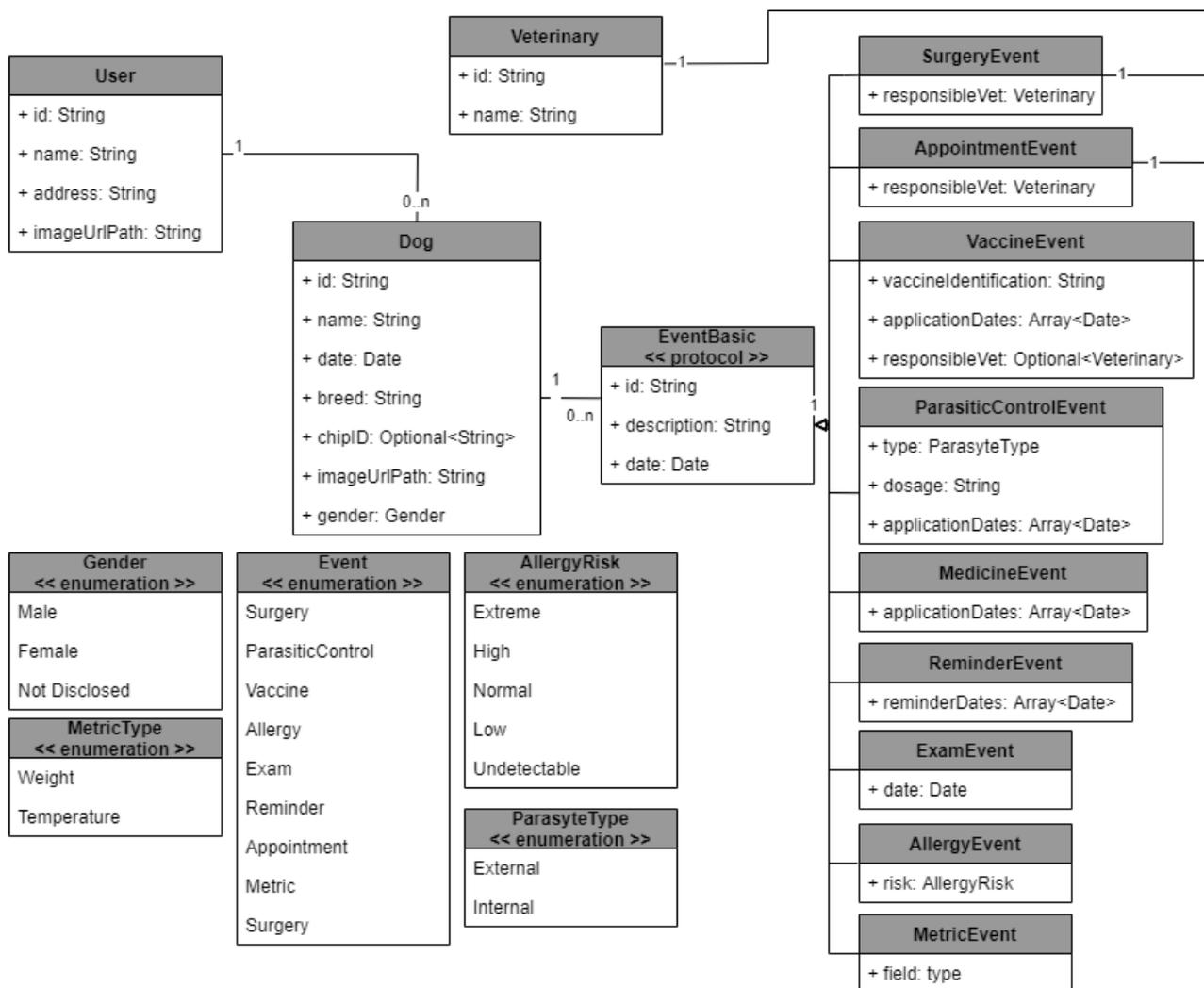
4.2 Modelagem

Nesta seção, é mostrada a concepção da abordagem de modelagem de objetos dentro do projeto MyDog. Essa concepção é volátil a mudanças durante o processo de implementação. A Figura 4.2 mostra os modelos utilizados, que são os seguintes:

- **Dog:** Faz a associação de cada cachorro a uma estrutura de dados com as informações básicas do cachorro junto com um identificador único.
- **Event:** Faz a generalização de todos os eventos (cirurgia, controle parasitário, exame, lembrete, medicamento, vacina) sobre um mesmo tipo. Essa estrutura de dados foi projetada para ser uma enumeração com um tipo associado, onde esse tipo é um dos eventos mencionados anteriormente.
- **SurgeryEvent:** Faz a associação de um evento cirúrgico a um cachorro.
- **ParasticControlEvent:** Faz a associação de um evento de controle parasitário a um cachorro.
- **ExamEvent:** Faz a associação de um evento de exame a um cachorro.
- **ReminderEvent:** Faz a associação de um evento de lembrete a um cachorro.
- **MedicineEvent:** Faz a associação de um evento de medicamento a um cachorro.
- **VaccineEvent:** Faz a associação de um evento vacina a um cachorro.
- **AppointmentEvent:** Faz a associação de um evento de compromisso a um cachorro. (Este modelo foi descontinuado, devido as similaridades a um evento de lembrete.)
- **MetricEvent:** Faz a associação de um evento de medição de temperatura e/ou peso a um cachorro.

- AllergyEvent: Faz a associação de um evento de alergia a um cachorro. (Este modelo foi descontinuado, devido ao fato de que alergias devem ser registradas ao cadastro do cachorro, não estreitamente relacionado a um evento.)
- Gender: Faz a associação de um cachorro a um tipo de gênero biológico.
- Veterinary: Faz associação de um evento que possui necessidade de acompanhamento veterinário a esse evento.
- User: Faz a associação de um usuário a múltiplos cachorros. (Este modelo foi descontinuado, devido o autor ter tomado a decisão de uma implementação local da aplicação sem necessidade de uma sessão de usuário.)

Figura 4.2 – Conceito original do MyDog



4.3 Histórias de usuário

Histórias de usuário é uma metodologia no processo de desenvolvimento ágil (BECK; ANDRES, 2004) onde o objetivo é focar nos objetivos do usuário e como o sistema pode alcançar esses objetivos. Essas histórias de usuário dividem os requisitos do sistema para que seja mais fácil a estimativa do mesmo. As seguintes histórias de usuário foram criadas para este projeto:

- User Story 1: Como usuário eu quero cadastrar meu perfil.
- User Story 2: Como usuário eu quero que o perfil do meu cachorro tenha nome, idade, gênero, endereço, alergias e a raça.
- User Story 3: Como usuário eu quero cadastrar 1 ou mais cachorros no meu aplicativo.
- User Story 4: Como usuário eu quero deletar cachorros previamente cadastrados.
- User Story 5: Como usuário eu quero visualizar os próximos eventos por data.
- User Story 6: Como usuário eu quero visualizar os próximos eventos por cachorro.
- User Story 7: Como usuário eu quero cadastrar um evento do tipo cirurgia.
- User Story 8: Como usuário eu quero cadastrar um evento do tipo controle parasítico.
- User Story 9: Como usuário eu quero cadastrar um evento do tipo exame.
- User Story 10: Como usuário eu quero cadastrar um evento do tipo lembrete.
- User Story 11: Como usuário eu quero cadastrar um evento do tipo medicamento.
- User Story 12: Como usuário eu quero cadastrar um evento do tipo vacina.
- User Story 13: Como usuário eu quero cadastrar um evento do tipo higiênico (tosa, banho, bucal, ...)
- User Story 14: Como usuário eu quero controlar o peso do meu cachorro visualmente.
- User Story 15: Como usuário eu quero controlar a temperatura do meu cachorro visualmente.
- User Story 16: Como usuário eu quero receber notificações de eventos próximos.
- User Story 17: Como usuário eu quero visualizar meus próximos eventos estilo calendário.
- User Story 18: Como usuário eu quero que o limite de eventos visualizados seja

dos próximos 6 meses.

5 IMPLEMENTAÇÃO

Neste segmento do trabalho será tratado os detalhes, ferramentas e mudanças necessárias que se desencadearam durante o decorrer do projeto. Assim como, informações sobre hardwares, frameworks e técnicas de desenvolvimento empregadas.

5.1 Ambiente de desenvolvimento

5.1.1 Hardware

Para desenvolvimento do projeto MyDog foram utilizados dois macbooks com as seguintes especificações:

- MacBook Pro (Retina, 13 polegadas) com processador Intel Core I5, 2015
- MacBook Pro (13 polegadas) com processador Apple M1, 2020

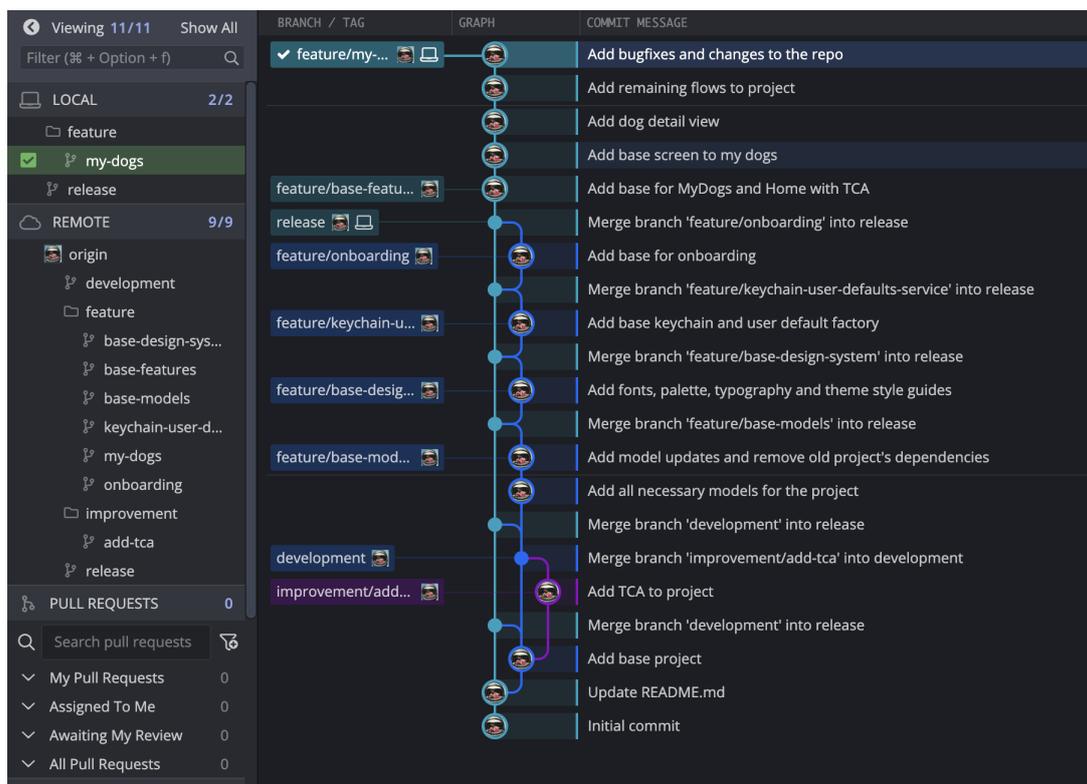
Em ambas as máquinas foi utilizado o sistema operacional macOS BigSur versão dentro da faixa da versão 11, sendo a última versão usada a 11.4.

Para a escrita e compilação do código foi utilizada a plataforma Xcode na versão 12.5.1. Todo o layout foi construído através de Code View sem a necessidade de utilização do Interface Builder da IDE junto com a utilização do sistema de previews que permite uma visualização rápida da UI sem necessidade de compilação da aplicação.

A linguagem escolhida para desenvolvimento foi Swift versão 5.4.2. A escolha do Swift se deu devido ao fato de ser uma linguagem de programação mais moderna e que está sendo atualizada com as novas funcionalidades e bibliotecas que estão sendo criadas para os sistema da Apple. O objective-C tem se tornado obsoleto nos últimos anos. Além disso a criação da interface gráfica foi feita através da biblioteca de SwiftUI onde só é possível utilizar com Swift.

Para versionamento do código foi utilizado o git com um repositório criado no GitHub sobre o nome de petcare. A utilização do git foi feito através da GUI do GitKraken a fim de facilitar e economizar tempo nas execuções de comandos do versionamento e manutenção do código dentro do repositório. A figura 5.1 demonstra a estrutura em uma fase de implementação do projeto.

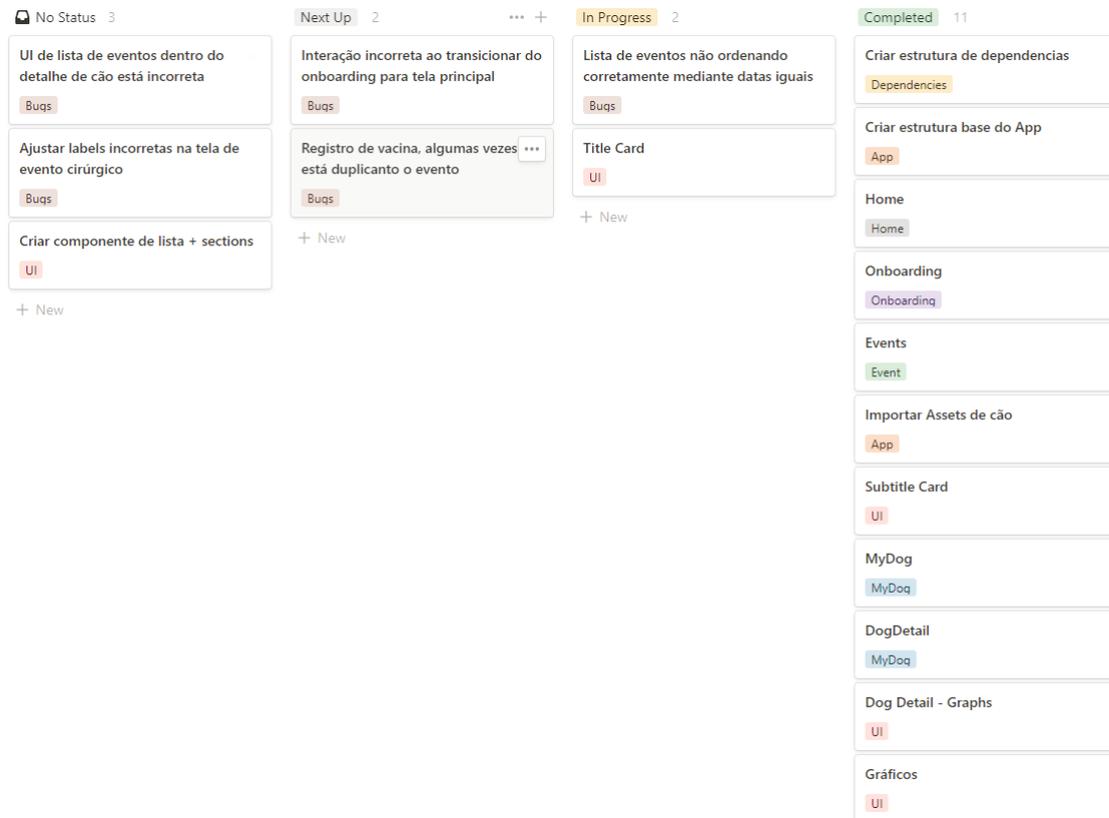
Figura 5.1 – Versionamento do código via GitKraken



5.2 Organização de tarefas

Em referência a organização das tasks, foi utilizado o Notion através de Kanban, onde cada task foi criada com o intuito de ser um incremental entregável, assim conseguindo observar a progressão do desenvolvimento da aplicação mesmo quem em pequenos passos de desenvolvimento. Novas alterações e bugs surgiram, foram rastreados e entraram direto no *board* em uma escala de prioridade de desenvolvimento. Durante o desenvolvimento do projeto, o Notion foi uma das peças fundamentais para garantir que o fluxo do sistema estava progredindo de forma correta. A escolha da plataforma Notion ao invés de outras plataformas conhecidas como Jira e Trello, foi devido ao uso prévio do autor com essa ferramenta. A figura 5.2 demonstra a utilização do *board* em uma das fases de desenvolvimento.

Figura 5.2 – Organização de tarefas via Notion



5.3 Frameworks utilizados

As seguintes bibliotecas foram utilizadas dentro do projeto MyDog:

- **Foundation:** Biblioteca nativa de iOS que prove funcionalidades básicas para aplicação, tais como tipos essenciais coleções, serviços de sistema operacional.
- **UIKit:** Biblioteca nativa de iOS de interface gráfica base, utilizado como auxiliar devido à algumas limitações da biblioteca de SwiftUI. (APPLE, 2021k)
- **SwiftUI:** Biblioteca nativa de iOS de construção de interface gráfica de forma declarativa.
- **Kingfisher:** Biblioteca terceira que permite download e cache de imagens da internet construído com uma camada de Swift. (WANG, 2015)
- **TheComposableArchitecture:** Biblioteca terceira que prove uma arquitetura destinada para aplicações em SwiftUI focada em componentização de módulos e interface gráfica.

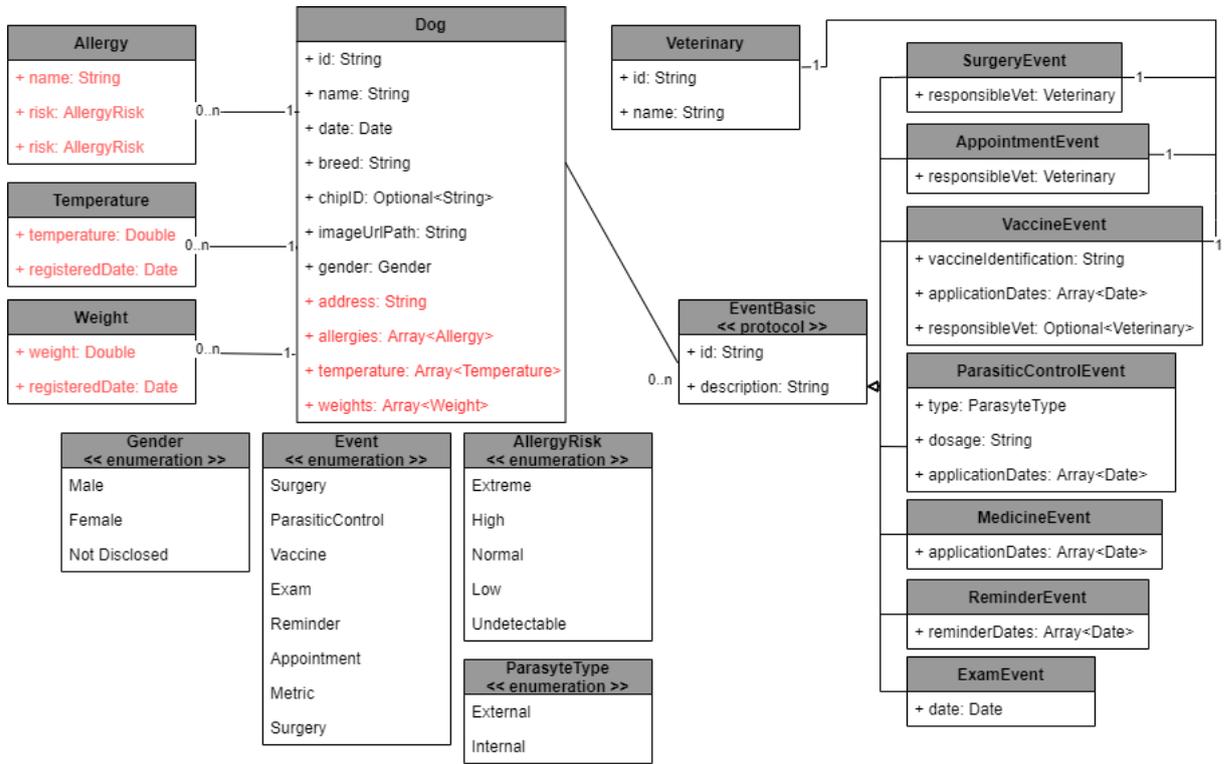
- **KeychainSwift:** Biblioteca terceira que prove funções de auxílio para salvar e carregar dados do keychain.
- **SwiftUICharts:** Biblioteca terceira que prove componentes visuais de auxílio a construções de gráficos em SwiftUI. (SAMU, 2019)
- **Introspect:** Biblioteca terceira que auxilia no acesso de componentes do UIKit que estão abaixo de componentes de SwiftUI. Necessário devido as limitações do SwiftUI. (SITE LINE, 2019)

5.4 Alterações na modelagem

Durante o processo de desenvolvimento houve necessidade de troca de escopo, alteração de modelos, demonstradas na figura 5.3, que não foram corretamente pensados para o propósito da aplicação. A seguir estão as mudanças que ocorreram:

- Removido a estrutura de dados de User. O aplicativo alterou de escopo para uma versão local sem necessidade de sessão do usuário.
- Removido a estrutura de dados AllergyEvent e criada uma nova estrutura Allergy que é utilizada dentro da estrutura de Dog. Alergias não são necessariamente eventos de um cachorro e sim uma característica do mesmo.
- Removido a estrutura de dados MetricEvent e MetricType e criada uma estrutura individual de temperatura e peso. Essas métricas não faziam sentido existirem pois elas não iriam nunca serem rastreadas dentro da listagem de eventos, portanto foram movidas para outro tipo de estrutura associada a um cachorro.
- Removido da enumeração de eventos o item Allergy e Metric, devido aos motivos mencionados nos 2 itens acima.
- Remoção do item de data dentro do protocolo EventBasic. As propriedades de data são utilizadas de forma específica dentro de alguns tipos de evento, tais como evento de vacina, controle parasítico, medicamento, entre outros.
- Adicionado a informação de endereço dentro da estrutura de cachorro. No início da concepção do projeto não se pensou em ser uma informação relevante, no entanto o autor do projeto decidiu inserir ela dentro da aplicação.

Figura 5.3 – Alterações da modelagem



6 APLICAÇÃO

Neste capítulo serão apresentados os resultados do projeto de desenvolvimento MyDog, passando pelas funcionalidades do sistema.

O usuário ao interagir com a aplicação pela primeira vez é levado ao Onboarding onde ocorre o processo de registro do seu primeiro animal. Por uma regra de negócio da aplicação o usuário necessita ter ao menos um cachorro ativo para funcionamento integral do aplicativo. Após realização do cadastro de seu primeiro cachorro de estimação, o usuário tem acesso a área principal da aplicação, que possui 3 fluxos diferentes: fluxo principal, fluxo de meus cachorros e fluxo de eventos. A seguir são explicados abordar cada passo das funcionalidades.

6.1 Onboarding

O onboarding dentro da aplicação é separado em seis passos principais obrigatórios para criação de um cachorro. Ao interagir com o aplicativo pela primeira vez, o usuário é apresentado com uma visualização paginada (demonstradas nas figuras 6.1, 6.2 e 6.3) desses seis passos:

- Inserir o nome do seu cachorro de estimação;
- Inserir o link de uma imagem de seu cachorro de estimação;
- Inserir a idade do seu cachorro de estimação;
- Inserir o sexo do seu cachorro de estimação;
- Inserir o endereço de residência do seu cachorro de estimação;
- Inserir a raça do seu cachorro de estimação.

Após realizar estes seis passos, o usuário pode confirmar e prosseguir para a área principal da aplicação.

Figura 6.1 – Passos 1 e 2 do onboarding

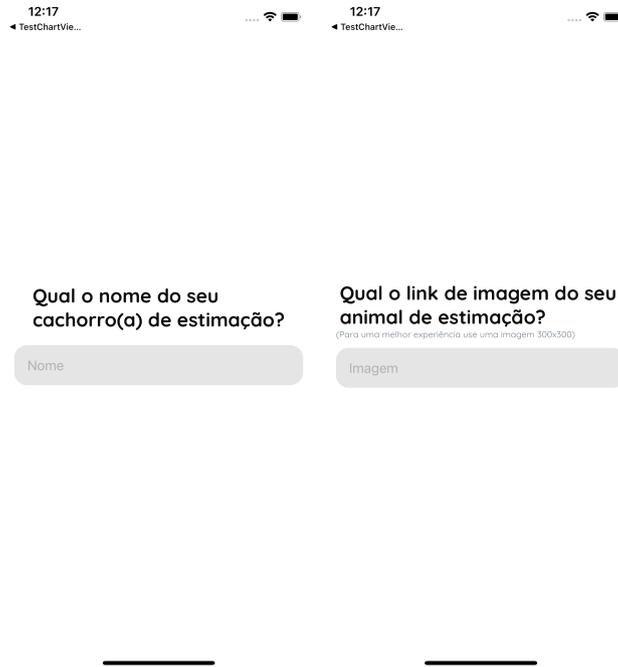


Figura 6.2 – Passos 3 e 4 do onboarding

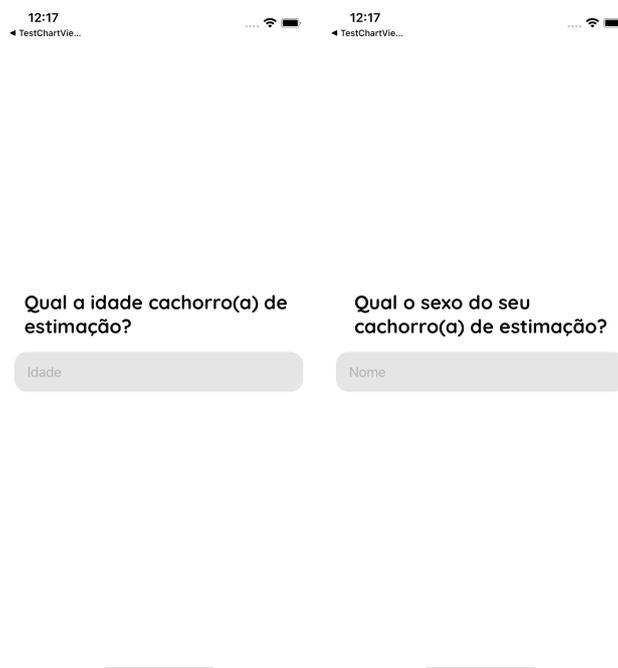
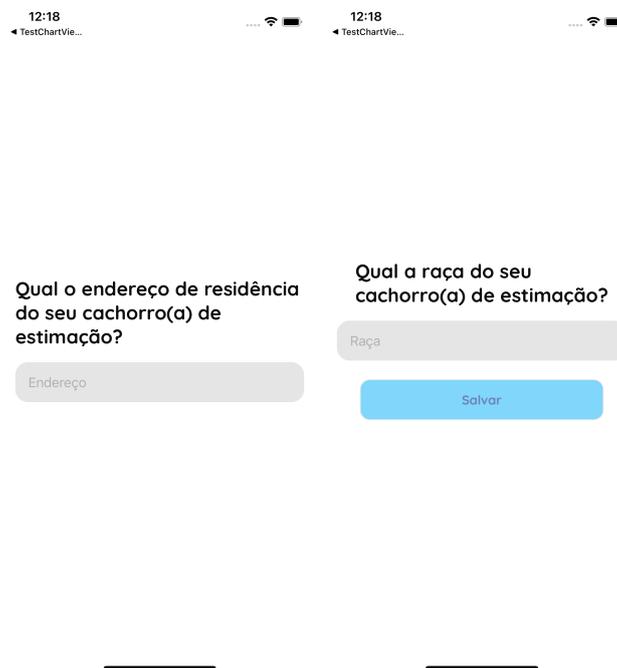


Figura 6.3 – Passos 5 e 6 do onboarding

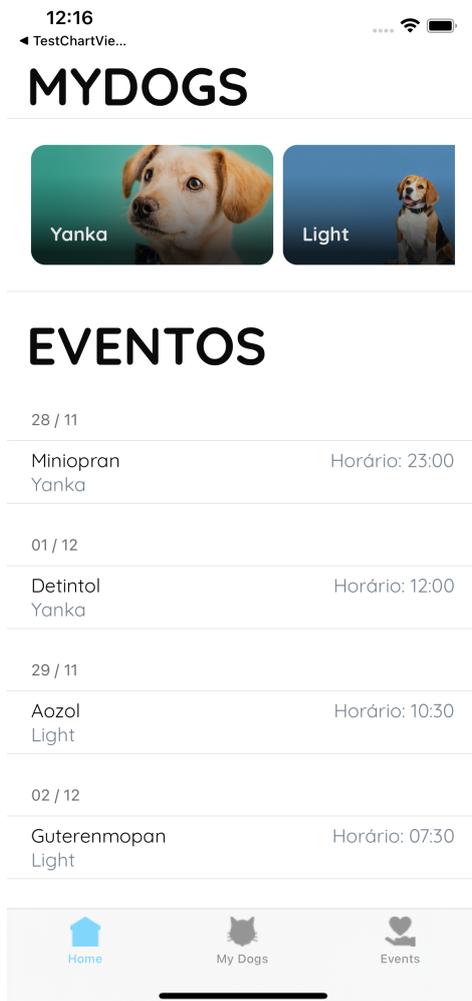


6.2 Home

O fluxo principal se inicia na tela da Home (Figura 6.4), onde se prove ao usuário acesso rápido a seus cachorros de estimação no topo da tela e logo abaixo uma lista de resumida dos eventos que seus cachorros tem se aproximando.

Ao interagir com um dos cachorros no topo da tela, o usuário é levado a tela de detalhe do cachorro selecionado.

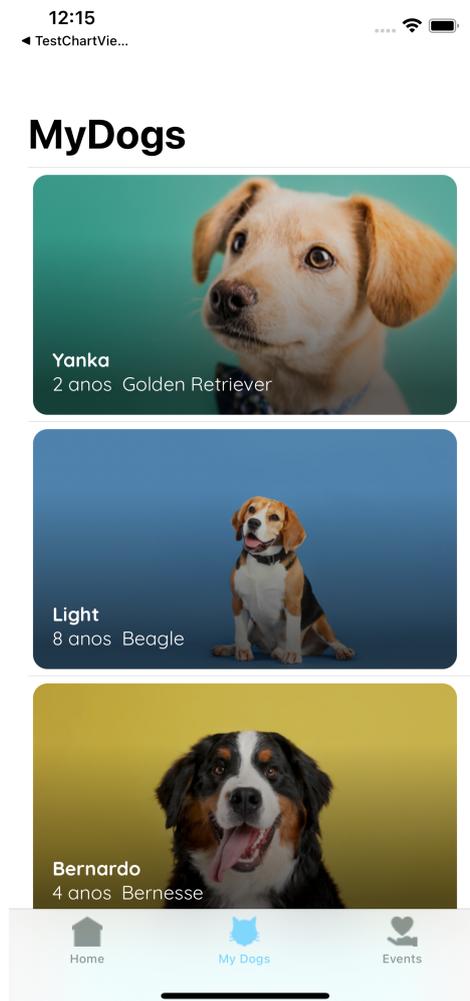
Figura 6.4 – Tela de Home



6.3 Meus cachorros

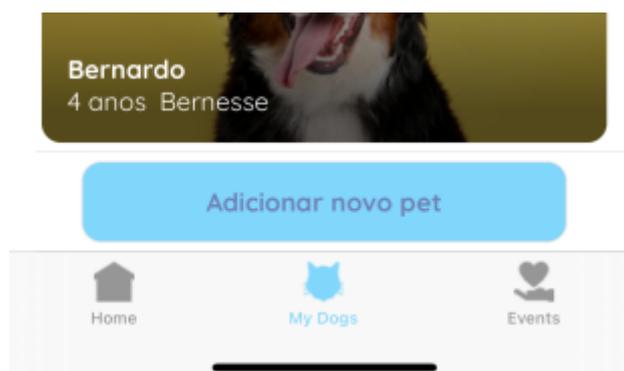
Ao interagir com o botão de MyDog na aba na parte debaixo do dispositivo, o usuário é levado para tela de listagem de seus cachorros (Figura 6.5) registrados na aplicação. Ao interagir com qualquer um dos cachorros, o usuário será levado a tela de detalhe de seu cachorro.

Figura 6.5 – Tela de meus cachorros



Na figura 6.6, há a listagem de cachorros onde existe um botão para adicionar novo cachorro. Ao interagir com esse botão, o usuário é levado para tela de cadastro de um cachorro que é o mesmo fluxo do Onboarding.

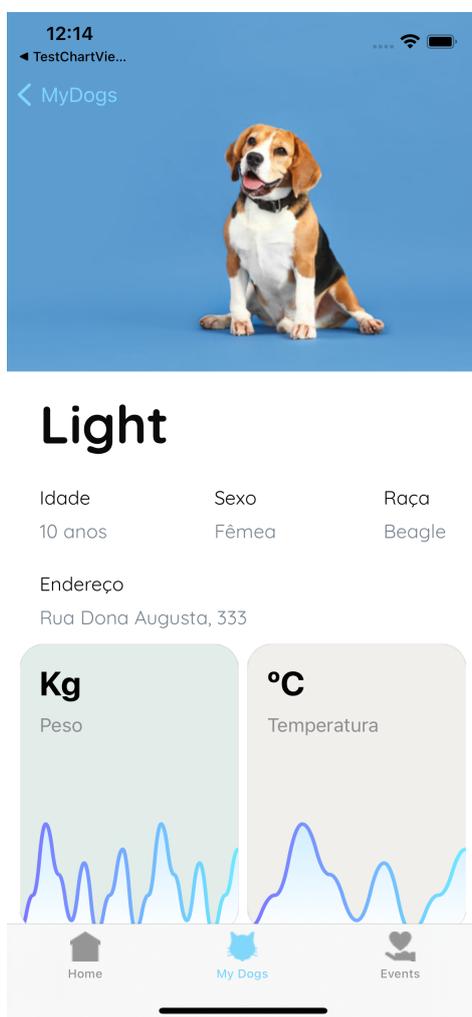
Figura 6.6 – Botão adicionar novo cachorro



6.4 Detalhe de cachorro

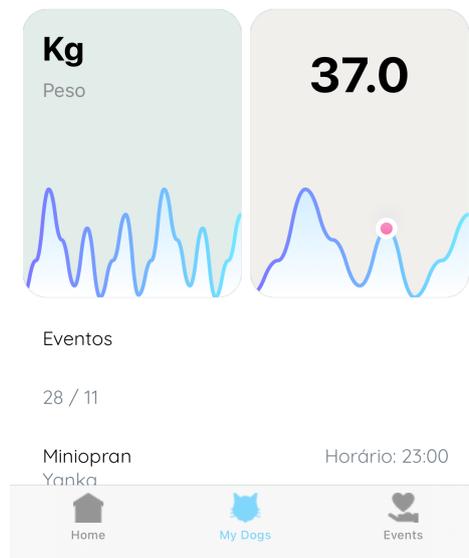
A figura 6.7 ilustra todos os detalhes sobre os cachorros cadastrados. Seu nome, idade, raça, endereço, alergias, estatística de temperatura, estatística de peso, lista de próximos eventos desse cachorro.

Figura 6.7 – Tela de detalhe de cachorro



Os gráfico dessa tela (Figura 6.7) são interativos para que o usuário consiga acompanhar a mudança de peso e temperatura do seu cachorro. A figura 6.8 ilustra como os gráficos são exibidos acima da listagem de eventos.

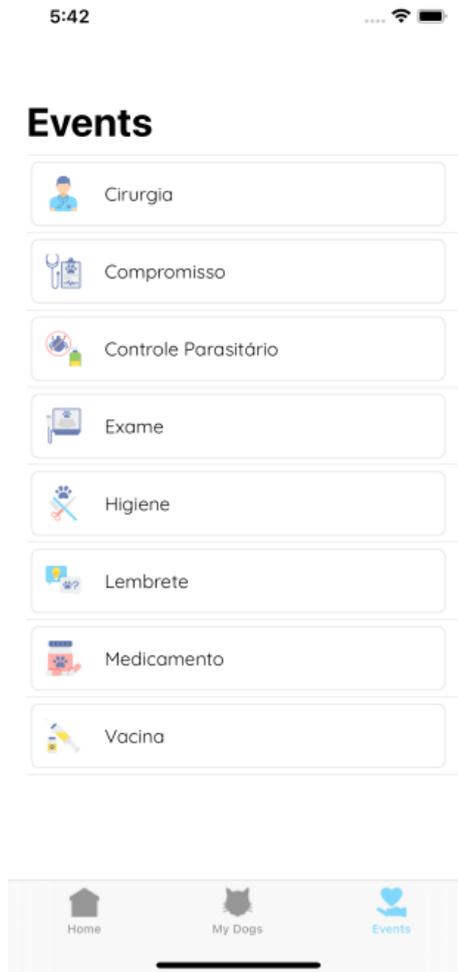
Figura 6.8 – Interação gráfico e listagem de eventos



6.5 Eventos

Ao interagir com o botão de eventos na aba na parte debaixo do dispositivo, o usuário é levado para tela de listagem (Figura 6.9) de seus possíveis eventos a serem registrados. Ao interagir com qualquer um dos eventos o usuário, é levado a uma tela base de cadastro de evento denominada de detalhe de evento.

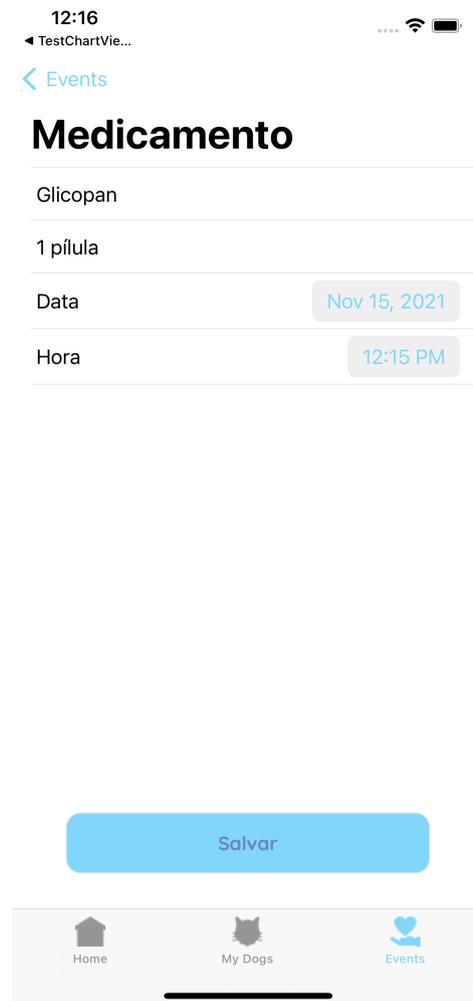
Figura 6.9 – Tela de eventos



6.6 Detalhe de evento

Todas as telas de evento seguem uma base de design ilustrado na figura 6.10, nesse momento, o usuário deve inserir os dados do evento registrado para um cachorro específico ou múltiplos a escolha do usuário. Após fazer essa inserção o usuário é direcionado para tela de eventos.

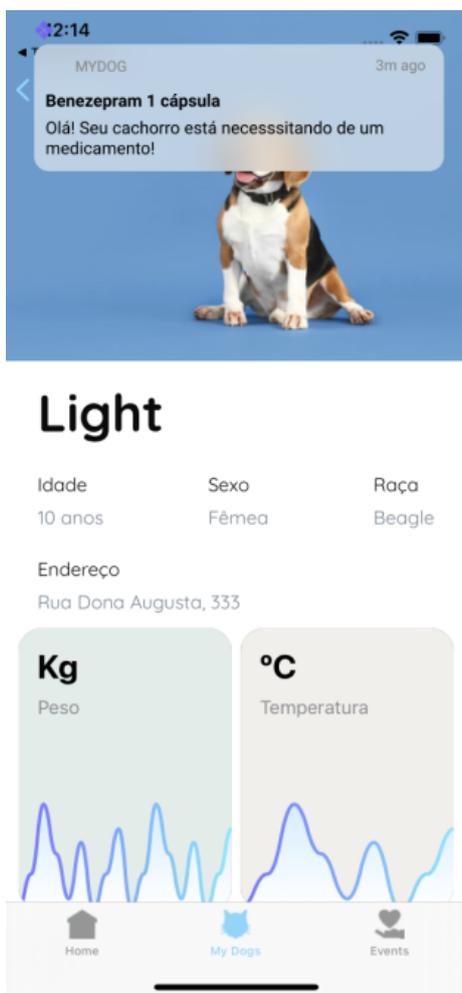
Figura 6.10 – Tela de detalhe de evento (medicamento)



6.7 Notificação de evento

Ao receber uma notificação, o usuário é apresentado com a seguinte interface visual (Figura 6.11) de alerta. ao interagir com a notificação o usuário será levado a aplicação.

Figura 6.11 – Notificação de evento



7 EXPERIMENTO DE USABILIDADE

Este capítulo relata o experimento de usabilidade que foi realizado com os usuários que aceitaram participar voluntariamente. A avaliação foi feita através de um teste de usabilidade com uma introdução básica da aplicação em seguida de uma lista de tarefas pré-determinadas onde o usuário necessitava realizar com a aplicação MyDog em mãos.

Após utilização da aplicação foi apresentado um formulário ao usuário na plataforma Google Forms onde o usuário precisava responder algumas perguntas para entendermos um pouco seu contexto sócio-cultural e depois algumas perguntas baseadas na escala Likert (LIKERT, 1987), onde o usuário deve escolher uma opção em uma escala de 1 (Discordo totalmente) até 5 (Concordo totalmente) em relação a usabilidade da aplicação.

7.1 Protocolo

No momento da realização do experimento o aplicativo ainda não estava disponível dentro da AppStore para *download*. Portanto, a realização desses testes foi junto a presença do autor. A lista de tarefas seguir foi apresentada aos participantes:

- Cadastrar 2 cachorros na aplicação;
- Registrar uma temperatura;
- Registrar um evento de medicamento com data de expiração para 2 minutos no futuro;
- Registrar X eventos diferentes dentro da aplicação a escolha do usuário;
- Verificar a listagem de eventos após o registros do item acima;
- Interagir com o gráfico na tela de detalhe de cachorro.

Após utilização do aplicativo, o usuário recebeu um link do formulário que se encontra anexado dentro do Apêndice A, o qual busca entender um pouco sobre as características do usuário assim como a perspectiva do mesmo sobre a usabilidade da aplicação.

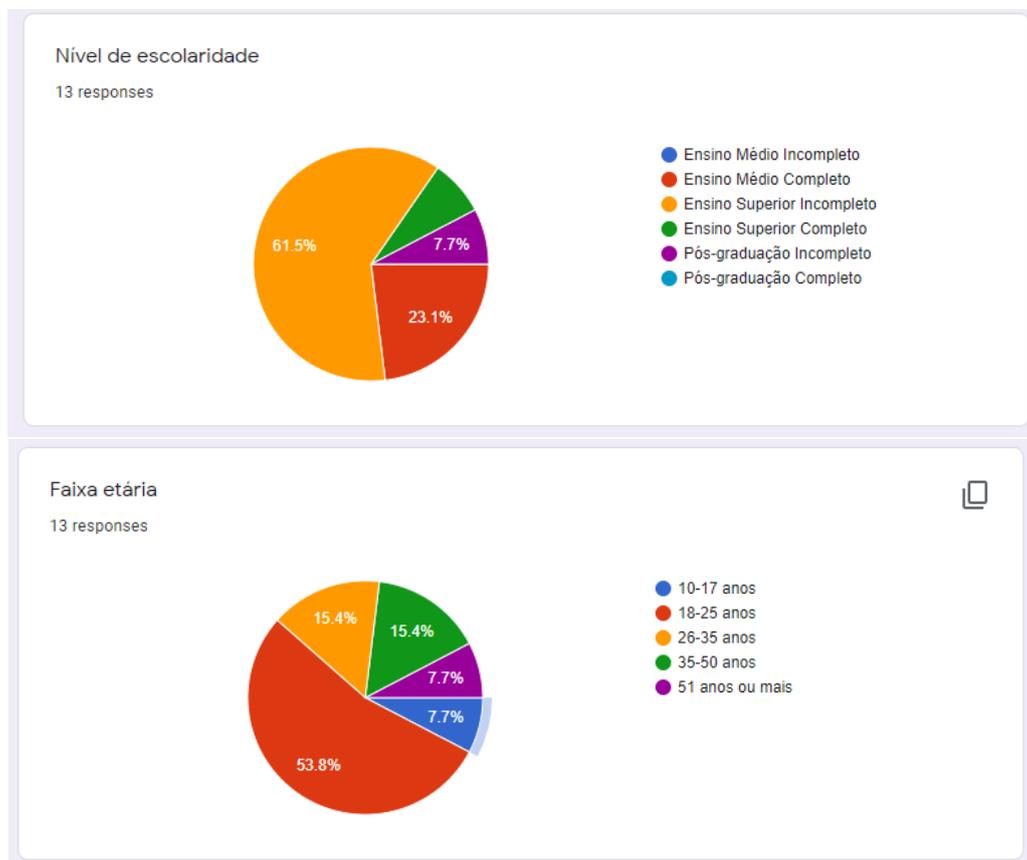
7.2 Participantes

Para o experimento, o aplicativo foi inserido na mão de 13 indivíduos que tem sob seus cuidados pelo menos 1 cachorro de estimação. Como o objetivo do MyDog é cuidados e organização sobre os afazeres dos cachorros domésticos, foi necessário fazer a busca por candidatos que já possuem uma experiência de vida com esses animais.

7.3 Resultados

Conforme pode ser observado na figura 7.1, os resultados do formulário apresentam que acima de 90% dos participantes são maiores de 18 anos, com apenas 1 indivíduo atendendo a faixa de 10-17 anos de idade. Quanto ao nível de escolaridade, acima de 75% dos indivíduos finalizaram ou estão atualmente cursando o ensino superior.

Figura 7.1 – Resultado (Escolaridade e Faixa etária)



Outra informação é que todos os participantes possuem pelo menos 1 cachorro de estimação que era um requerimento para fazer parte do experimento social. Conforme a

figura 7.2 é visível que grande maioria das pessoas possuem 1 cachorro com 61,5% dos casos.

Figura 7.2 – Resultado (número de cachorros de estimação)



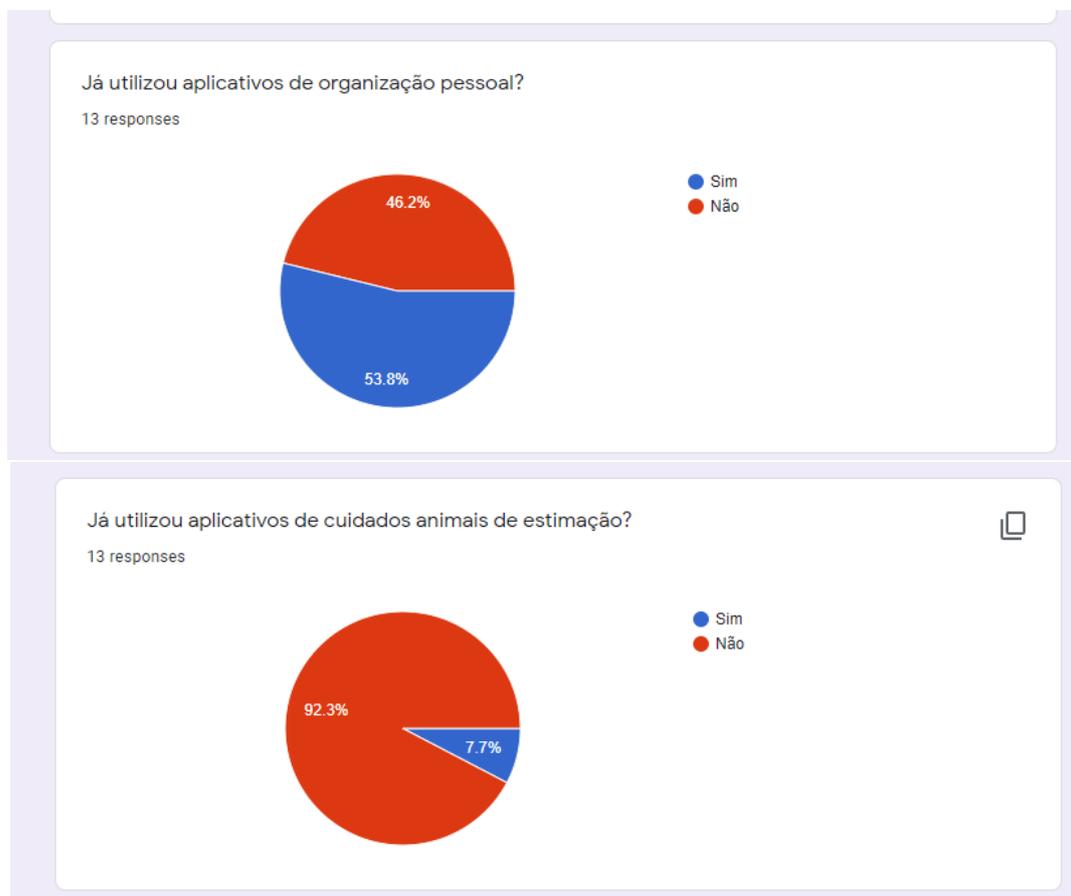
A figura 7.3 aponta que os participantes possuem um grau de conhecimento elevado com relação a smartphones e internet. Acima de 80% dos candidatos tem familiaridade ao mexer com aplicativos o que é um indicador positivo em relação aos feedbacks de utilização sobre o aplicação.

Figura 7.3 – Resultado (Experiência em dispositivos móveis e internet)



As perguntas sobre aplicativos de organização e cuidados de animais apontam de acordo com a figura 7.4 que apesar de termos metade dos candidatos com conhecimento no uso de sistemas de organização e afazeres, temos uma grande maioria que nunca sentiu necessidade de um software que os ajudassem a atender as necessidades de seus cachorros de estimação.

Figura 7.4 – Resultado (Experiência em aplicativos de organização e cuidado de animais)

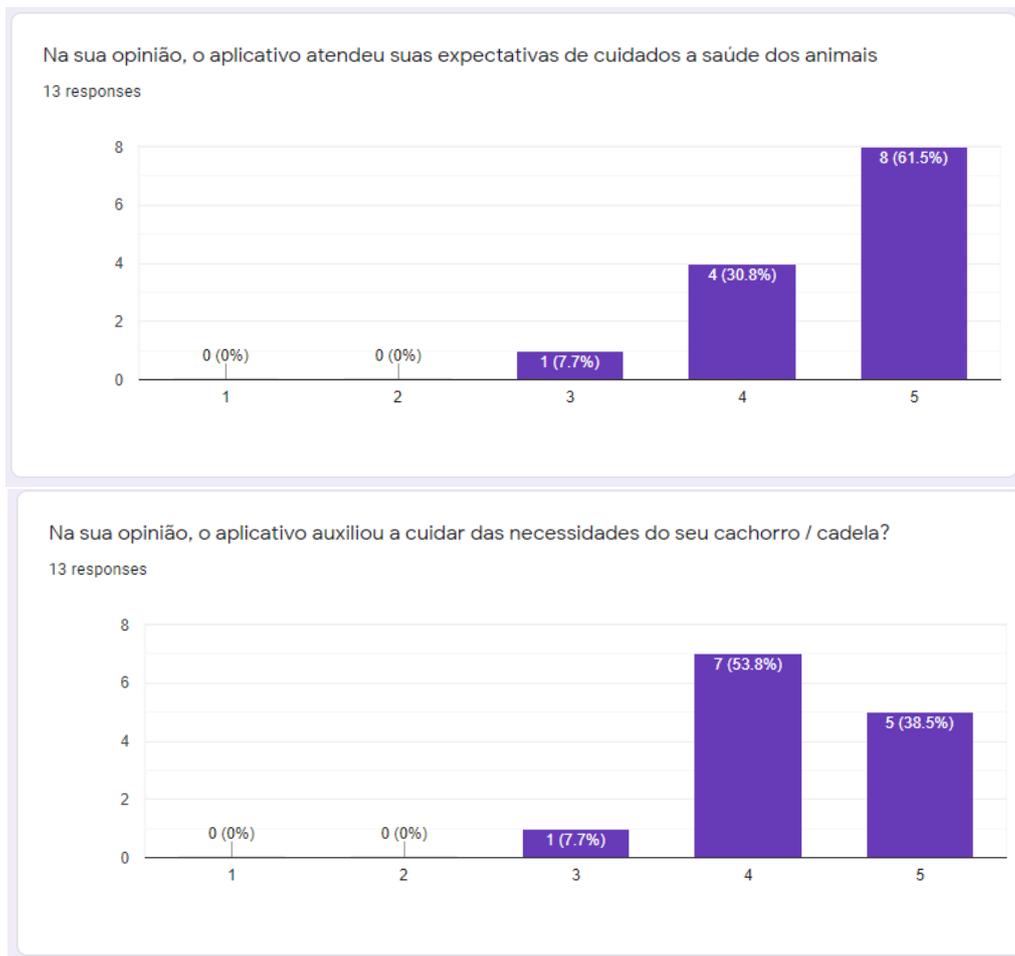


As perguntas abaixo, são focadas mais na usabilidade do MyDog. Conforme a figura 7.5 e 7.6, é visível que as avaliações estão tendendo ao lado positivo da balança, entretanto há uma distribuição sobre os pontos de 3 à 5 em multiplas questões apresentadas. Isso é um indicador de que apesar da experiência ser satisfatória, há espaço para melhorias.

Figura 7.5 – Resultado (Usabilidade cadastro, evento, expectativas organização)



Figura 7.6 – Resultado (Usabilidade cadastro, evento, expectativas)



7.4 Considerações Finais

Nesta seção foi visto pelos dados colhidos que a experiência de usabilidade da aplicação MyDog há grande espaço para melhorias dentro dos donos de cachorros. Especialmente dentro do cadastro do cachorro, um possível início para melhoria no processo de cadastro seria a realização de um novo fluxo de cadastro com uma nova pesquisa via formulário para observar as mudanças nas estatísticas. Os candidatos apesar de não terem experiência com aplicativos de gestão de cuidados a cachorros domésticos viram potencial na aplicação dado que acima de 80% dos usuários concordaram que o aplicativo após a sua introdução atendem suas expectativas do que esse tipo de aplicação deve trazer ao cliente final.

8 CONCLUSÃO

Este trabalho apresentou o projeto e desenvolvimento de uma aplicação destinada ao auxílio de donos de cachorros de estimação em questões de organização e cuidados relacionados ao animal. Esse objetivo é alcançado através de um aplicativo iOS onde o mesmo prove ao usuário funcionalidades de registro de eventos relacionados ao seu cachorro, notificações para lembrar os afazeres e uma agenda de acesso rápido para cada um de seus bichos de estimação. A concepção da aplicação foi iniciada em 2017 dentro do Apple Developer Academy e o desenvolvimento iniciou-se em 2021 com esse trabalho de conclusão de curso. A aplicação faz parte de um sistema local ao dispositivo móvel que não tem necessidade de acesso a internet para poder realizar suas ações como um todo. Os resultados vistos no experimento de usabilidade nos mostraram que há potencial de mercado, uma vez que, acima de 90% dos abordados indicaram que nunca tiveram experiência ao utilizar uma aplicação que tem como o intuito organizar a vida do animal de estimação assim facilitando o tempo do dono como um todo. No entanto, como apontado no mesmo experimento, há espaços de melhoria em questão de UX.

Para o futuro, seria interessante adicionar a funcionalidade de login a aplicação junto com a criação de um webservice que salva as informações, possibilitando o compartilhamento de cadastros de cachorros pelos seus múltiplos donos. Outra funcionalidade interessante seria possibilidade de gerar um histórico canino para apresentar ao veterinário assim mantendo toda a informação sobre seu cachorro centralizado em apenas um local salvo na web. Além disso, é necessário endereçar melhorias nos fluxos em questão de experiência do usuário. Em conjunto, seria interessante o desenvolvimento de um aplicativo para a plataforma Android com a visão de expandir o público alcançado.

9 APÊNDICE A - FORMULÁRIO

Avaliação de usabilidade - MyDog

Este questionário tem o intuito de avaliar a experiência de usabilidade do aplicativo MyDog para um trabalho de conclusão do curso de Ciência da Computação UFRGS. O objetivo é coletar opiniões dos usuários pós uso da aplicação que como forma de verificar a qualidade de aplicativos ed gerenciamento de tempo e cuidados de saúde de cachorros. O tempo estimado deste questionário é de 10 minutos e os resultados são de forma anônima.

*Obrigatório

Nível de escolaridade *

- Ensino Médio Incompleto
- Ensino Médio Completo
- Ensino Superior Incompleto
- Ensino Superior Completo
- Pós-graduação Incompleto
- Pós-graduação Completo
- Outro: _____

Faixa etária *

- 10-17 anos
- 18-25 anos
- 26-35 anos
- 35-50 anos
- 51 anos ou mais

Quantos cachorros você possui cuidados sobre? *

- 1
- 2
- 3 ou mais

Qual sua experiência com uso de internet? *

- 1 2 3 4 5
- Totalmente inexperiente Totalmente experiente

Qual sua experiência com uso de smartphones? *

- 1 2 3 4 5
- Totalmente inexperiente Totalmente experiente

Já utilizou aplicativos de organização pessoal? *

- Sim
- Não

Já utilizou aplicativos de cuidados animais de estimação? *

Sim

Não

Na sua opinião, qual foi o nível de cadastrar um cachorro no aplicativo *

Muito difícil 1 2 3 4 5 Muito fácil

Na sua opinião, qual foi o nível de cadastrar um evento no aplicativo *

Muito difícil 1 2 3 4 5 Muito fácil

Na sua opinião, o aplicativo atendeu suas expectativas de organização *

Totalmente discordo 1 2 3 4 5 Totalmente concordo

Na sua opinião, o aplicativo atendeu suas expectativas de cuidados a saúde dos animais *

1 2 3 4 5

Totalmente discordo Totalmente concordo

Na sua opinião, o aplicativo auxiliou a cuidar das necessidades do seu cachorro / cadela? *

1 2 3 4 5

Totalmente discordo Totalmente concordo

REFERÊNCIAS

- ABINPET. **A Indústria Pet e seus números**. 2018. Disponível em: <http://abinpet.org.br/infos_gerais/>.
- APPLE. **Aplicativo todoist**. 2007. Disponível em: <<https://apps.apple.com/us/app/todoist-to-do-list-tasks/id572688855>>.
- APPLE. **Aplicativo Omnifocus3**. 2008. Disponível em: <<https://apps.apple.com/us/app/omnifocus-3/id1346190318>>.
- APPLE. **Objective-C**. 2014. Disponível em: <<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>>.
- APPLE. **Keychain Services**. 2015. Disponível em: <https://developer.apple.com/documentation/security/keychain_services>.
- APPLE. **Aplicativo My Track Pet**. 2018. Disponível em: <<https://apps.apple.com/br/app/mytrackpet-cidadao/id1363963965>>.
- APPLE. **SwiftUI - Preview Provider**. 2020. Disponível em: <<https://developer.apple.com/documentation/swiftui/previewprovider/previews-swift.type.property>>.
- APPLE. **About Swift Language**. 2021. Disponível em: <<https://www.swift.org/about/>>.
- APPLE. **Aplicativo MeuPet**. 2021. Disponível em: <<https://apps.apple.com/br/app/meupet-cuidado-animal/id1342342113>>.
- APPLE. **Aplicativo Things3**. 2021. Disponível em: <<https://apps.apple.com/br/app/things-3/id904237743>>.
- APPLE. **Apple Worldwide Developers Conference 2021**. 2021. Disponível em: <<https://developer.apple.com/wwdc21/>>.
- APPLE. **The AppStore Platform**. 2021. Disponível em: <<https://www.apple.com/app-store/>>.
- APPLE. **The Combine Framework**. 2021. Disponível em: <<https://developer.apple.com/documentation/combine>>.
- APPLE. **The Foundation Framework**. 2021. Disponível em: <<https://developer.apple.com/documentation/foundation>>.
- APPLE. **The iOS Platform**. 2021. Disponível em: <<https://www.apple.com/br/ios/ios-15/>>.
- APPLE. **The macOS Platform**. 2021. Disponível em: <<https://developer.apple.com/macos/>>.
- APPLE. **The SwiftUI Framework**. 2021. Disponível em: <<https://developer.apple.com/documentation/swiftui>>.

APPLE. **The UIKit Framework**. 2021. Disponível em: <<https://developer.apple.com/documentation/uikit>>.

APPLE. **The Xcode IDE**. 2021. Disponível em: <<https://developer.apple.com/xcode/>>.

ASPCA. **Vaccinations for your pet**. 2021. Disponível em: <<https://www.aspc.org/pet-care/general-pet-care/vaccinations-your-pet>>.

AXOSOFT. **GitKraken GUI**. 2016. Disponível em: <<https://www.gitkraken.com/git-client>>.

BECK, K.; ANDRES, C. **Extreme Programming Explained: Embrace Change**. Pearson Education, 2004. (XP Series). ISBN 9780134051994. Disponível em: <<https://books.google.com.br/books?id=-DNcBAAAQBAJ>>.

BENECKE, N. Studies on early dog remains from northern europe. **Journal of Archaeological Science**, v. 14, n. 1, p. 31–49, 1987. ISSN 0305-4403. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305440387800043>>.

CODE, C. **Empresa Culture Code**. 2012. Disponível em: <<https://culturedcode.com/things/>>.

ELDORADO. **Apple Developer Academy Porto Alegre**. 2010. Disponível em: <<https://developeracademy.eldorado.org.br/poa/>>.

ESPINHA, R. G. **O que é kanban?** 2019. Disponível em: <<https://artia.com/kanban/>>.

EVGENII. **The Keychain Swift Framework**. 2015. Disponível em: <<https://github.com/evgenyneu/keychain-swift>>.

GAMMA, R. E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995. (Addison-Wesley professional computing series). ISBN 9783827330437. Disponível em: <<https://books.google.com.br/books?id=tmNNfSkfTlcC>>.

HOFMEISTER, N. **Obesidade em Cães: Controle de Peso e Prevenção**. 2019. Disponível em: <<https://saude.abril.com.br/vida-animal/como-combater-a-obesidade-de-caes-e-gatos/>>.

IBGE. **Tabela 4930 - Domicílios com algum cachorro, por situação do domicílio**. 2019. Disponível em: <<https://sidra.ibge.gov.br/tabela/4930>>.

LABS, N. **Notion Platform**. 2018. Disponível em: <<https://www.notion.so/>>.

LIKERT, R. **A Technique for the Measurement of Attitudes**. Archives of Psychology, Columbia University, 1987. (Archives of psychology). Disponível em: <<https://books.google.com.br/books?id=373RzQEACAAJ>>.

LOPES, V. **Os parasitas externos do cachorro**. 2021. Disponível em: <<https://www.peritoanimal.com.br/os-parasitas-externos-do-cachorro-21253.html>>.

MUEMA, L. **Declarative vs Imperative UI in Android**. 2021. Disponível em: <<https://www.section.io/engineering-education/declarative-vs-imperative-ui-android/>>.

PAIVA, R. **Você sabia que a higiene dos animais é tão importante quanto a dos seres humanos?** 2018. Disponível em: <<https://www.uov.com.br/cursos-online-pet/artigos/voce-sabia-que-a-higiene-dos-animais-e-tao-importante-quanto-a-dos-seres-humanos>>.

PET, M. T. **My Track Pet Inc.** 2017. Disponível em: <<https://mytrackpet.com/>>.

PETS, S. **Sistema de gestão para pet shops e clínicas veterinárias.** 2021. Disponível em: <<https://www.meupet.io/>>.

PETZ. **Temperatura de cachorro.** 2019. Disponível em: <<https://dicaspetz.com.br/cachorros/temperatura-de-cachorro-tire-suas-duvidas-sobre-o-tema/>>.

POINTFREE. **The Pointfree Org.** 2018. Disponível em: <<https://www.pointfree.co/blog>>.

POINTFREE. **The Composable Architecture.** 2020. Disponível em: <<https://github.com/pointfreeco/swift-composable-architecture>>.

PRESSMAN, R. **Engenharia de software.** Makron Books, 1995. ISBN 9788534602372. Disponível em: <<https://books.google.com.br/books?id=VvczPwAACAAJ>>.

REACTIVEX. **Reactive X Community.** 2011. Disponível em: <<http://reactivex.io/>>.

SAMU, A. **The SwiftUI Chart View Framework.** 2019. Disponível em: <<https://github.com/AppPear/ChartView>>.

SANT'ANA, L. **Dia Nacional dos Animais.** 2020. Disponível em: <<https://canaldopet.ig.com.br/cuidados/saude/2020-03-14/dia-nacional-dos-animais-medicina-veterinaria-preventiva-importancia-caes-gatos.html>>.

SITELINE. **The SwiftUI Introspect Framework.** 2019. Disponível em: <<https://github.com/siteline/SwiftUI-Introspect>>.

SKERRETT, I. **Eclipse Community Survey 2014 Results.** 2014. Disponível em: <<https://dzone.com/articles/eclipse-community-survey-2014>>.

STATISTA, E. B. **Global dog and cat pet population 2018.** 2018. Disponível em: <<https://www.statista.com/statistics/1044386/dog-and-cat-pet-population-worldwide/>>.

WANG, W. **The Kingfisher Framework.** 2015. Disponível em: <<https://github.com/onevc/Kingfisher>>.

X, R. **The RxSwift Framework.** 2015. Disponível em: <<https://github.com/ReactiveX/RxSwift>>.