**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído**

Projeto Integrado

Relatório Técnico

Sistema de Gestão de Vida Animal

Marcio Gadelha

Porto, Portugal

Junho, 2022.

# Projeto Integrado – Arquitetura de Software Distribuído

**Sumário**

[Projeto Integrado – Arquitetura de Software Distribuído 2](#_Toc116769209)

[1. Introdução 3](#_Toc116769210)

[2. Cronograma do Trabalho 6](#_Toc116769211)

[3. Especificação Arquitetural da solução 7](#_Toc116769212)

[3.1 Restrições Arquiteturais 7](#_Toc116769213)

[3.2 Requisitos Funcionais 8](#_Toc116769214)

[3.3 Requisitos Não Funcionais 9](#_Toc116769215)

[3.4 Mecanismos Arquiteturais 10](#_Toc116769216)

[4. Modelagem Arquitetural 11](#_Toc116769217)

[4.1 Diagrama de Contexto 11](#_Toc116769218)

[4.2 Diagrama de Container 12](#_Toc116769219)

[4.3 Diagrama de Componentes 13](#_Toc116769220)

[4.4 Projeto das Interfaces 14](#_Toc116769221)

[4.5 Código da Aplicação 17](#_Toc116769222)

[Referências 18](#_Toc116769223)

## Introdução

Nos últimos anos, há uma grande tendência na sociedade como um todo em adquirir animais de estimação. Segundo pesquisa do Insurance Information Institute, por exemplo, a porcentagem da população que possui um pet saltou de 56% em 1988 para 67% em 2020. Com isso, a relação entre os pets e seus donos está cada dia mais próxima, sendo que muitos donos os consideram como membros da família.

E, assim como as pessoas querem o melhor para seus filhos, esses pais de pets querem o melhor para seus animais. O resultado é uma abundância de produtos e serviços que até pouco tempo eram desconhecidos.

Consequentemente, ocorreu um aumento na procura produtos destinados a pets, como uma alimentação mais adequada, suprimentos, brinquedos, roupas, higiene e consultas veterinárias. Além disso, surgiram novos serviços, como hotéis e até mesmo spa para os animais.

Neste sentido, o mercado está demonstrando ser promissor, pois apresentou um grande crescimento nos últimos anos. Em 2006, os americanos gastaram US$ 38,5 bilhões com seus animais de estimação. Já em 2018, eles quase dobraram esse número, para US$ 72,6 bilhões. Ou seja, o mercado praticamente dobrou em pouco mais de uma década. A maior parcela de vendas do mercado de animais domésticos está relacionada à alimentação. Nos EUA em 2018 foram gastos US$ 30,3 bilhões com alimentos pet e estimou-se um aumento de 4,3% para até o final de 2019. No Brasil esta projeção também é observada, a Associação Brasileira da Indústria de Produtos para Animais de Estimação (Abinpet) indicou que a alimentação pet ocupa 70% das vendas totais nesta área. Inclusive, a venda de alimentos industrializados para pets aumentou 28,5% no último ano. Ou seja, mesmo com novas opções o mercado ainda está em alta. Vale mencionar, que uma alimentação adequada é um dos principais fatores que contribuem para saúde dos animais. Por isso, a busca dos donos por adquirir alimentos de qualidade e mais sofisticados, resulta em uma grande potencialidade expansão do mercado pet.

Atualmente, o mercado pet apresenta um grande potencial para investimentos. A busca dos donos por mais serviços e produtos para animais de estimação, proporciona novas oportunidades de negócio. A divisão do Departamento de Trabalho nos EUA projeta um crescimento na área de animais domésticos de 16% entre 2018 e 2028, o que é bastante significativo. Em alguns lares, inclusive, existem mais animais de estimação do que crianças. Por que isso acontece? Isso está relacionado a mudanças culturais, uma vez que atualmente as pessoas estão esperando mais tempo para terem filhos. Entretanto, há necessidade de uma companhia, por isso, adotam um animal de estimação. Além disso, são observadas mudanças também na forma de tratamento dos pets, devido à maior proximidade entre os animais e seus donos, demonstrando um aumento da humanização de animais de estimação. Com isso, donos aumentaram a procura por produtos de alta qualidade e acessórios mais caros e gourmetizados. Devido à demanda, o mercado mantém-se em alta e ao mesmo tempo saturado quando falamos das opções tradicionais, como pet shops e clínicas veterinárias. Porém, existe um espaço enorme para opções inovadoras, como: restaurantes para pets, roupas humanizadas, hotéis, escola e aplicativos para encontrar cuidadores. Além de uma atenção especial com a alimentação, que envolve desde a cautela na escolha de ingredientes que compõem o pet food até priorizar alimentos que atendem exigências específicas para gatos e cachorros. Um grande diferencial desse segmento é a constante demonstração de ser consistente e anti crise, com alta de 13,5% em ano de pandemia, o setor pet vislumbrava grande crescimento em 2021. Em 2020, ano em que a economia brasileira encolheu pelo menos 4% e diferentes atividades econômicas amargaram perdas expressivas, o setor pet teve um crescimento estimado de 13,5% em relação a 2019, com faturamento acima dos R$ 40 bilhões, de acordo o Instituto Pet.

Mesmo diante desse enorme mercado consumidor, existe uma escassez de ferramentas de qualidade para suportar a administração e controle, bem como prover produtos e serviços de forma prática e organizada.

Em nossa versão inicial (MVP), temos como missão prover uma plataforma que proporcione os seguintes recursos:

* Administração de informações das características dos animais como peso, tamanho, vacinas, bem como fotos e vídeos;
* Ser integrada com um mecanismo de autenticação aberto e popular para facilitar o acesso de novos usuários;
* De fácil entendimento e com excelente usabilidade para atingir os mais diversos perfis de usuários;
* Ser robusta, escalável e tolerante a falhas para suportar o crescimento da operação a nível mundial;

Portanto, neste documento serão apresentados os requisitos arquiteturais, funcionais, não funcionais e as diagramações da solução para o desenvolvimento da plataforma 4Pet que compreenda os objetivos citados acima.

## Cronograma do Trabalho

A seguir é apresentado o cronograma proposto para as etapas deste trabalho.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 01/06/2022 | 01/06/2022 | 1. Cronograma do Trabalho | Construção desta tabela |
| 02/06/2022 | 02/06/2022 | 1. Contextualização do trabalho | Construção da contextualização deste projeto |
| 03/06/2022 | 05/06/2022 | 1. Definição dos requisitos Arquiteturais | Lista dos requisitos Arquiteturais identificados |
| 03/06/2022 | 05/06/2022 | 1. Definição dos requisitos Funcionais | Lista dos requisitos funcionais identificados |
| 03/06/2022 | 05/06/2022 | 1. Definição dos requisitos Não-funcionais | Lista dos requisitos Não-funcionais identificados |
| 03/06/2022 | 05/06/2022 | 1. Definição dos Mecanismos Arquiteturais | Lista dos Mecanismos Arquiteturais identificados |
| 06/06/2022 | 08/06/2022 | 1. Construção dos Diagramas de Contextos – Modelo C4 | Diagrama de contexto criado no Draw.io e documentado |
| 08/06/2022 | 15/06/2022 | 1. Revisão da Etapa 1 | Documento Etapa 1 revisado |
| 16/06/2022 | 20/06/2022 | 1. Construção do vídeo de apresentação da Etapa 1 | Vídeo criado da Etapa 1 |
| 10/08/2022 | 10/08/2022 | 1. Apresentação em PPT da Etapa 1 | PPT |
| 10/08/2022 | 10/08/2022 | 1. Publicação no repositório Github Etapa 1 | Arquivos produzidos no Github disponíveis abertamente |
| 11/08/2022 | 12/08/2022 | 1. Construção dos Diagramas de Contêineres | Diagramas de contêineres |
| 13/08/2022 | 15/08/2022 | 1. Construção dos Diagramas de Componentes | Diagramas de componentes |
| 16/08/2022 | 18/08/2022 | 1. Desenho dos Wireframes da POC | Protótipos de telas de baixa fidelidade |
| 19/08/2022 | 30/09/2022 | 1. Código da aplicação | Aplicação com 3 requisitos implementados |
| 30/09/2022 | 30/09/2022 | 16. Publicação no repositório Github Etapa 2 | Arquivos produzidos no Github disponíveis abertamente |
| 01/10/2022 | 03/10/2022 | 17. Análise das abordagens arquiteturais | Seção do documento produzido |
| 04/10/2022 | 05/10/2022 | 18. Cenários | Seção do documento produzido |
| 06/10/2022 | 07/10/2022 | 19. Evidências da avaliação | Seção do documento produzido |
| 08/10/2022 | 09/10/2022 | 20. Resultados obtidos | Seção do documento produzido |
| 10/10/2022 | 11/10/2022 | 21. Avaliação crítica dos resultados | Seção do documento produzido |
| 12/10/2022 | 13/10/2022 | 22. Conclusão | Seção do documento produzido |
| 14/10/2022 | 19/10/2022 | 23. Construção do vídeo de apresentação da Etapa 3 | Vídeo da etapa 3 disponível |
| 20/10/2022 | 20/10/2022 | 24. Publicação no repositório Github Etapa 3 | Arquivos produzidos no Github disponíveis abertamente |

## Especificação Arquitetural da solução

Esta seção apresenta a especificação básica da arquitetura da solução a ser desenvolvida, incluindo diagramas, restrições e requisitos definidos pelo autor, tal que permitem visualizar a macroarquitetura da solução.

## Restrições Arquiteturais

Os Requisitos Arquiteturais são todos os requisitos, sejam eles Funcionais ou Não-Funcionais que têm **impacto direto** sobre a Arquitetura do Sistema. Dessa forma, o Arquiteto precisa analisar os requisitos do sistema identificando algumas propriedades e então “filtrando” os Requisitos Arquiteturais. A lista a seguir apresenta os requisitos arquiteturais que foram identificados para implementação inicial da plataforma.

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RA01 | Deve ser usado tecnologias abertas (Open source) para o desenvolvimento de toda a plataforma. |
| RA02 | Deve ser usado o serviço de nuvem da Amazon Web Services como provedora da infraestrutura necessária para a plataforma. |
| RA03 | Deve ser usado o serviço OAuth 2.0 do Google bem como a possibilidade de criação de conta diretamente na plataforma, para o gerenciamento de autenticação dos usuários. |
| RA04 | A solução móvel (Apps Mobile) deve suportar os sistemas operacionais móveis mais populares, Android e IOS. |
| RA05 | Deve se implementar uma API RESTful para prover todos os dados e comunicação com os clientes de forma agnóstica e desacoplada com o backend, facilitando uma possível mudança tecnológica no frontend. |

## Requisitos Funcionais

Os Requisitos Funcionais são todos aqueles que estão associados às funcionalidades que ditam **o que** sistema deve fazer. A lista a seguir apresenta os requisitos funcionais identificados para o desenvolvimento inicial da plataforma.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O aplicativo deve permitir o cadastro de usuário através de uma conta do Google. | B | A |
| RF02 | O aplicativo deve permitir o cadastro de usuário através de um e-mail e senha. | B | A |
| RF03 | O aplicativo deve permitir a atualização do cadastro do usuário, podendo adicionar: endereço, coordenadas e telefone. | M | A |
| RF04 | O aplicativo deve permitir o cadastro de um pet, informando: Foto do Perfil, Tipo, Porte, Raça, Data de nascimento, Disponibilidade para cruzamento, Mãe de leite, Disponibilidade para adoção. | A | A |
| RF05 | O aplicativo deve permitir o cadastro do peso do pet e sua data de referência. | M | A |
| RF06 | O aplicativo deve permitir o cadastro de medidas do pet e sua data de referência. | M | A |
| RF07 | O aplicativo deve permitir o cadastro de medicamentos em uso pelo pet e sua periodicidade. | M | A |
| RF08 | O aplicativo deve permitir o cadastro de vacinas tomadas pelo pet e sua data de referência. | M | A |
| RF09 | O aplicativo deve permitir o cadastro de fotos do pet, com legenda e data referência. | A | A |
| RF10 | O aplicativo deve permitir o cadastro de vídeos do pet, com legenda e data referência. | A | A |
| RF11 | O aplicativo deve permitir a atualização de um pet, informando: Peso, Medidas, Medicamentos, Vacinas, Fotos e Vídeos. | A | A |
| RF12 | O aplicativo deve permitir os prestadores a cadastrar e disponibilizar seus serviços. | M | B |
| RF13 | O aplicativo deve permitir os vendedores a cadastrar e disponibilizar seus produtos. | M | B |
| RF14 | O aplicativo deve permitir os donos a pesquisar e adquirir produtos ou serviços. | A | B |
| RF15 | O aplicativo deve permitir os prestadores a consultar um relatório com estatísticas sobre seus serviços. | M | B |
| RF16 | O aplicativo deve permitir os vendedores a consultar um relatório com estatísticas sobre seus produtos. | M | B |
| RF17 | O aplicativo deve permitir os donos configurarem quais notificações e frequência deseja receber. | M | B |
| RF18 | O aplicativo deve permitir o dono avaliar o produto ou serviço adquirido. | M | B |
| RF19 | O aplicativo deve permitir o prestador de serviço configurar as notificações que deseja em relação a seus serviços disponíveis. | M | B |
| RF20 | O aplicativo deve permitir o vendedor configurar as notificações que deseja em relação a seus produtos disponíveis. | M | B |
| RF21 | O sistema deve enviar informações referentes aos serviços para o prestador respeitando suas configurações. | M | B |
| RF22 | O sistema deve enviar informações referentes aos produtos para o vendedor respeitando suas configurações. | M | B |

\*B=Baixa, M=Média, A=Alta.

## Requisitos Não Funcionais

Os Requisitos Não Funcionais estão associados às restrições de funcionalidades que **ditam como** o sistema deve fazer. A lista a seguir apresenta os requisitos não funcionais identificados para o desenvolvimento inicial da plataforma.

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | A plataforma deve habilitar a autenticação baseado no modelo OAuth2 da Google e diretamente no sistema. | A |
| RNF02 | O sistema deve operar em tempo integral em 24h x 7d, tendo disponibilidade mínima de 90%, e para atender essa necessidade deverá usar a hospedagem AWS com mecanismo de escala automática para responder ao aumento de demanda. | A |
| RNF03 | A comunicação entre o sistema Backend e Mobile da plataforma deve ser implementado através do padrão de serviços REST. | A |
| RNF04 | O sistema deverá utilizar para persistência um banco de dados NoSQL para atender a necessidade de alta performance na resposta em alta demanda, aceitando em contra partida uma integridade eventual. | A |
| RNF05 | As notificações por e-mail ou push devem operar por meio de filas de mensagens por não haver a necessidade de serem em tempo real e atenderem a grande demanda. | A |
| RNF06 | A versão Web da plataforma deve suportar os navegadores modernos, prioritariamente Google Chrome e Mozila Firefox. | M |

## Mecanismos Arquiteturais

Os mecanismos arquiteturais representam conceitos técnicos fundamentais que serão padronizados por toda a solução. Eles são refinados durante o projeto em três estados, representados pelas três categorias de Mecanismos Arquiteturais:

* Mecanismo de Análise, que dá ao mecanismo um nome, uma descrição resumida e alguns atributos básicos derivados dos requisitos do projeto.
* Mecanismo de Design, que são mais concretos e assumem alguns detalhes do ambiente de implementação.
* Mecanismo de Implementação, que especifica a exata implementação de cada mecanismo.

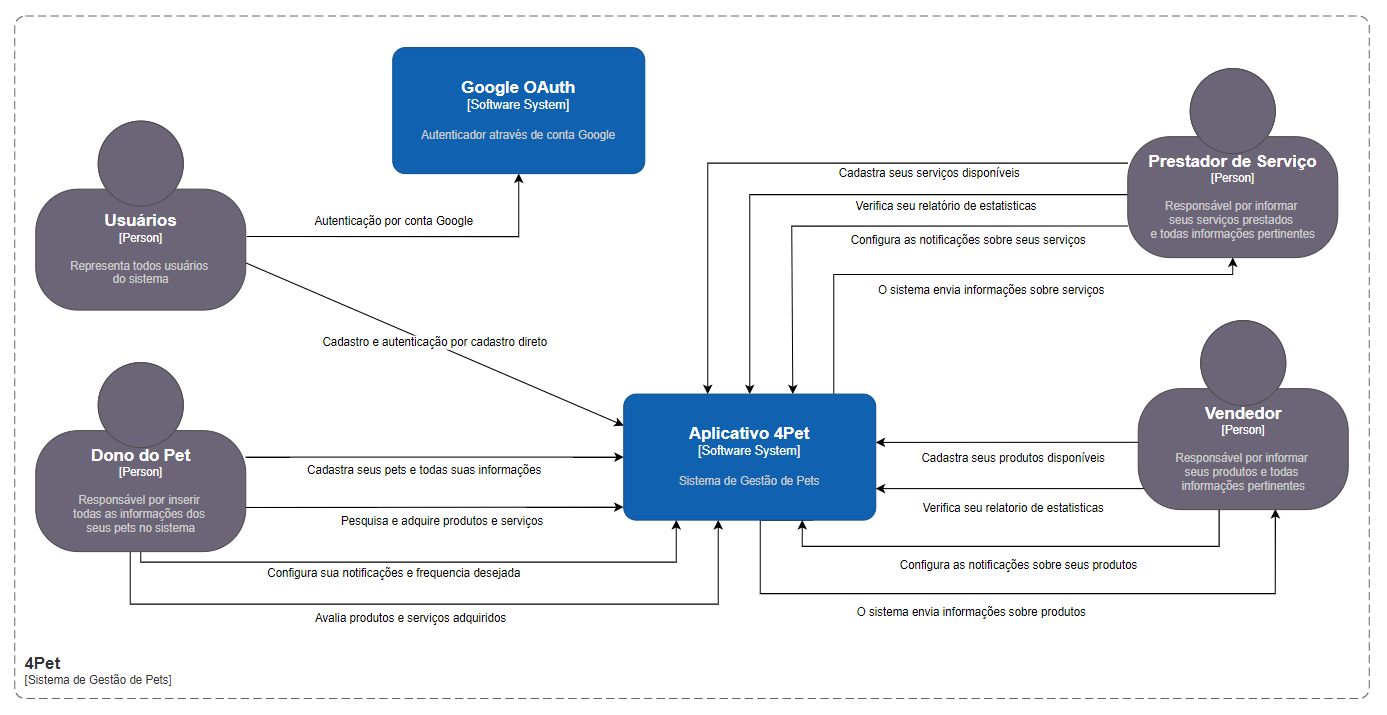
|  |  |  |
| --- | --- | --- |
| **Análise** | ***Design*** | **Implementação** |
| Persistência | ORM | JPA / Hibernate / Spring Data |
| Persistência | Banco de Dados NoSQL | Amazon DynamoDB |
| Front end | GraphQL | Spring GraphQL |
| Front end | Mobile | Google Flutter |
| Front end | Navegador Web | Mozilla Firefox / Google Chrome |
| Teste de Software | Testes Unitários | JUnit |
| Usabilidade | Notificação por Push | Amazon Pinpoint |
| Autenticação | OAuth2 | Google Open Authorization 2 |

## Modelagem Arquitetural

Desejamos criar uma solução focada no usuário que envolva mobilidade, visando ser uma ferramenta prática que esteja sempre nas mãos de donos de animais de estimação para ajudar dos cuidados diários com seus pets, bem como registrar seus momentos e adquirir produtos e serviços. Para isso através do diagrama de contexto abaixo demonstramos a criação de um componente de backend responsável por todos os serviços no servidor com regras de negócio e tendo um cliente leve apenas focado na apresentação baseado em uma solução hibrida utilizando o Ionic.

Para esta modelagem arquitetural optou-se por utilizar o modelo C4 para documentação de arquitetura de software. Mais informações a respeito podem ser encontradas aqui: <https://c4model.com/> e aqui: <https://www.infoq.com/br/articles/C4-architecture-model/>. Dos quatro níveis que compõem o modelo C4 três serão apresentados aqui e somente o Código será apresentado na próxima seção (5).

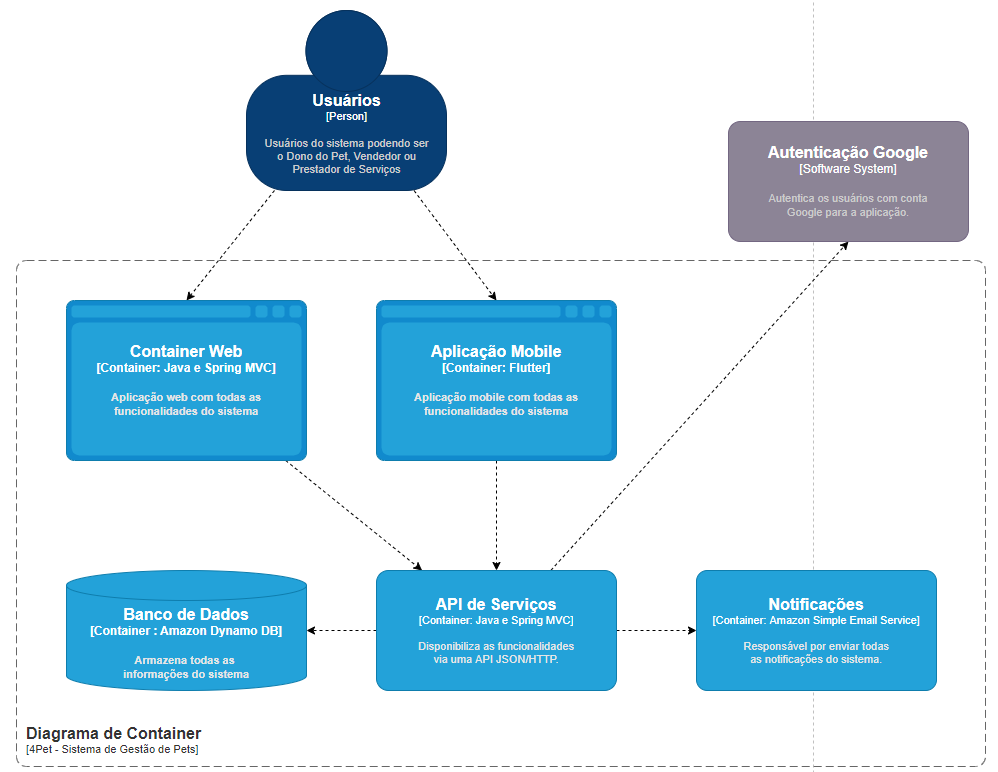
## 4.1 Diagrama de Contexto



**Figura 1 - Visão Geral da Solução 4Pet.**

A figura 1 mostra a especificação o diagrama geral da solução proposta, com todos seus principais sistemas e pessoas envolvidas nos processos de gestão de vida animal.

## 4.2 Diagrama de Container



**Figura 2 – Diagrama de Container da Solução 4Pet.**

A figura 2 mostra o diagrama de container no padrão C4 model para a solução proposta, com todos seus principais containers envolvidos.

## 4.3 Diagrama de Componentes

## 

**Figura 3 – Diagrama de Componentes da Solução 4Pet.**

A figura 1 mostra o diagrama de componentes da solução proposta, com todos seus principais componentes envolvidos nos processos de gestão de vida animal.

## 4.4 Projeto das Interfaces

Neste capítulo demonstramos as principais interfaces, através dos wireframe e o protótipo navegável da solução.

Wireframe: <https://bit.ly/3yISkcS>

Protótipo: <https://bit.ly/3Ep1O0e>.



**Figura 4 – Interface inicial para Login.**



**Figura 5 – Interface inicial para Cadastro de Usuário.**

**

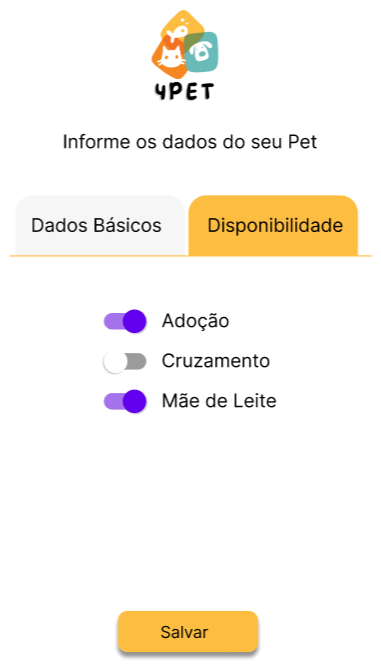
**Figura 6 – Interface inicial da aplicação após a autenticação (Home).**



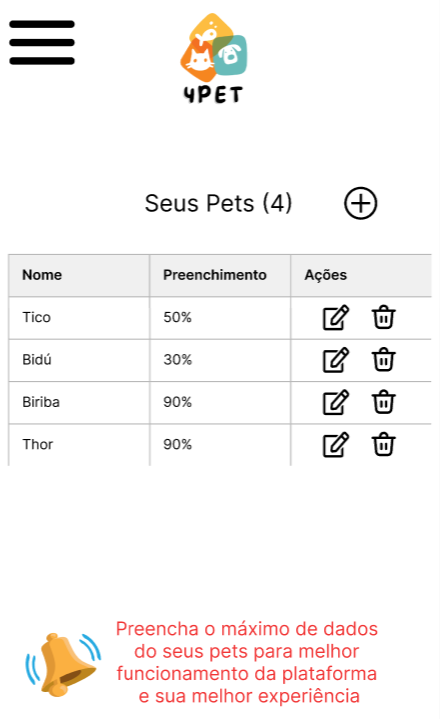
**Figura 7 – Interface do menu principal da aplicação.**



**Figura 8 – Interface do cadastro de Pet.**



**Figura 9 – Interface de configuração de disponibilidades do Pet.**



**Figura 10 – Interface inicial com Pets cadastrados (Home).**

## 4.5 Código da Aplicação

Neste capítulo demonstramos em detalhes como foi desenvolvido a aplicação.

Foram criados dois macro componentes arquiteturais, sendo um frontend e outro backend.

O frontend foi criado coma tecnologia híbrida baseada em Ionic e Angular, com isso, sendo necessário criar um único código fonte gerar os aplicativos para ambos os sistemas móveis.

O backend foi criado baseado em Java 8 e Spring Boot, com micro serviços, visado ter a capacidade de uma escala horizontal, redundância a alta disponibilidade.

**Repositórios:**

Frontend: <https://github.com/zagadelha/4-pet>

Branch: master

Backend: <https://github.com/zagadelha/tcc-mba-puc>

Branch: aws-dynamo-db

Diretório: tcc-mba-puc/code/4pet-backend/

## Referências

BRF INGREDIENTS. **Crescimento do mercado pet e seu papel na nutrição animal.**

Disponível em: <https://bit.ly/3OBxYIt>.

Acesso em: 24 de junho de 2022.

EXAME. **Mercado sem crise: com alta de 13,5% em ano de pandemia, o setor pet crescerá mais em 2021.**

Disponível em: <https://bit.ly/3u0Fsw4>.

Acesso em: 24 de junho de 2022.

CNN BRASIL. **Mercado pet deve ter crescimento de 14% em 2022, projeta instituto.**

Disponível em: <https://bit.ly/3ndS5jN>.

Acesso em: 24 de junho de 2022.

SEBRAE. **Mercado PET fatura quase 35 bi ao ano e tende a crescer.**

Disponível em: <https://bit.ly/3u3vhqO>.

Acesso em: 24 de junho de 2022.