

FATEC – SJC – Faculdade de Tecnologia de São José dos Campos – Prof. Jessen Vidal



Pet Sitter - Busca pelo cuidador de pets ideal

Relatório Entrega Final – Time 02

Bruna Larissa Clemente Gomes

Jennifer Dominique Dias Santos

Lucas Rafael Andrade Barroso

Prof. Claudio Lima

São José dos Campos – SP

2020

Índice

1.	<i>Introdução</i>	3
1.1	Objetivo Da Entrega.....	3
1.2.	Sprint Backlog.....	4
1.3.	Kanban Planner.....	6
2.	<i>Burndown Chart</i>	7
3.	<i>Arquitetura Básica</i>	8
4.	<i>Modelagem do Diagrama de Classes</i>	9
5.	<i>Desenvolvimento</i>	10
5.1	Mapeamento dos processos.....	10
5.2	Caso de Uso.....	11
5.3	Prototipação.....	12
5.4	Ferramentas Case.....	13
5.3.	Atividades Individuais Realizadas no Entrega:.....	14
6.	<i>Recomendações</i>	14
7.	<i>Referências</i>	15

1. Introdução

Time Scrum 02 (TS02) ficou responsável pelo planejamento e desenvolvimento do Software Pet Sitter, um software que ajudar os donos de pets encontrarem um *pet sitter* (babá de pets) para cuidar de seus pets.

Foi utilizado o método de priorização das tarefas **MoSCoW** (*Must-have, Should-have, Could-have, Will not have this time*), para o acompanhamento do desenvolvimento do projeto foi utilizado o modelo Burndown.

Burndown é uma representação gráfica do trabalho que resta fazer versus o tempo. O excelente trabalho (ou atraso) geralmente está no eixo vertical, com o tempo na horizontal. O Burndown é um gráfico de execução de trabalhos pendentes. É útil para prever quando todo o trabalho será concluído.

1.1 Objetivo Da Entrega

O principal problema da empresa, é a falta de um sistema que facilite a acessibilidade do cliente na busca de um cuidador para seus pets e no agendamento de serviços deles.

Observando o problema, foi concluído que seria necessário a criação de um sistema que permitisse o cadastro dos clientes e dos funcionários da empresa e a partir do cadastro, eles pudessem realizar, no caso do cliente, o cadastro de seus pets e o agendamento de data e hora em que utilizaria o serviço, no caso dos funcionários, o cadastramento de suas informações e localidades para melhor conhecimento do cliente sobre o cuidador que ficaria responsável pelo pet.

1.2. Sprint Backlog

Nota: Durante o desenvolvimento da Entrega, o TS02, o time teve que se reorganizar por conta da mudança da plataforma utilizada para a definição das atividades, por conta disso muitas atividades foram quebradas em outras pequenas tarefas para facilitar a visualização do andamento do projeto e o tempo que cada atividade estava levando para ser desenvolvida.

Para a realização das User Stories (US) foram utilizadas as informações fornecidas pela empresa e analisada as necessidades básicas necessárias para o funcionamento do software.

Tabela 01. User Story definida pelo TS03 através do MosCoW (40-1)

ID	MosCow	User Story	Crítérios de Aceitação
US101	40	Definição dos modelos a serem utilizados no programa	No início do processo, foi necessário realizar o levantamento dos dados para o desenvolvimento do projeto.
US102	40	Design Thinking e levantamento de Requisitos do projeto	Após realizar o levantamento dos dados, é necessário que haja uma documentação prévia do sistema para que todas as funcionalidades sejam desenvolvidas.
US103	38	Criação da Classe Pessoa e seus atributos e métodos	Classe que armazena todas as características comuns entre os clientes e os funcionários.
US104	38	Criação da Classe Cliente e seus atributos e métodos	Essa classe é responsável pelo armazenamento dos dados de cada cliente cadastrado.
US105	38	Criação da Classe Funcionário e seus atributos e métodos	Essa classe é responsável pelo armazenamento dos dados de cada funcionário cadastrado.
US106	38	Criação da Classe Pet e seus atributos e métodos	Classe responsável pela declaração das características dos pets e armazenamento dos mesmos
US107	37	Implementação do Cadastro do usuário na plataforma	Para que possa haver um agendamento do cuidado do pet, antes é necessário o cliente possuir um cadastro informando alguns dados pessoais e também sobre o seu pet.
US108	30	Implementação de um Ranking de Raças dos pets	O ranking pega todo os pets de todos os clientes da sua plataforma e informa quais são as raças mais comuns entre seus clientes.
US109	30	Implementação de Ranking de Gênero dos pets	Ranking que define qual gênero dos pets é mais comum entre seus clientes.

US110	30	Implementação de um histórico de atendimento dos funcionários da unidade	O histórico de atendimento do funcionário (<i>pet sitter</i>) tem o intuito de poder analisar quais os serviços ele já fez dentro da plataforma.
US111	30	Implementação do cadastramento de Pets durante o cadastro do cliente	De acordo com os requisitos passados pela empresa, cada cliente tem a possibilidade de cadastrar no máximo 5 pets.
US112	40	Implementação de cadastro dos funcionários	O cadastro de funcionários é necessário para saber as informações dos mesmos, e sem eles não é possível o cliente fazer o agendamento do cuidado do seu pet.

1.3. Kanban Planner

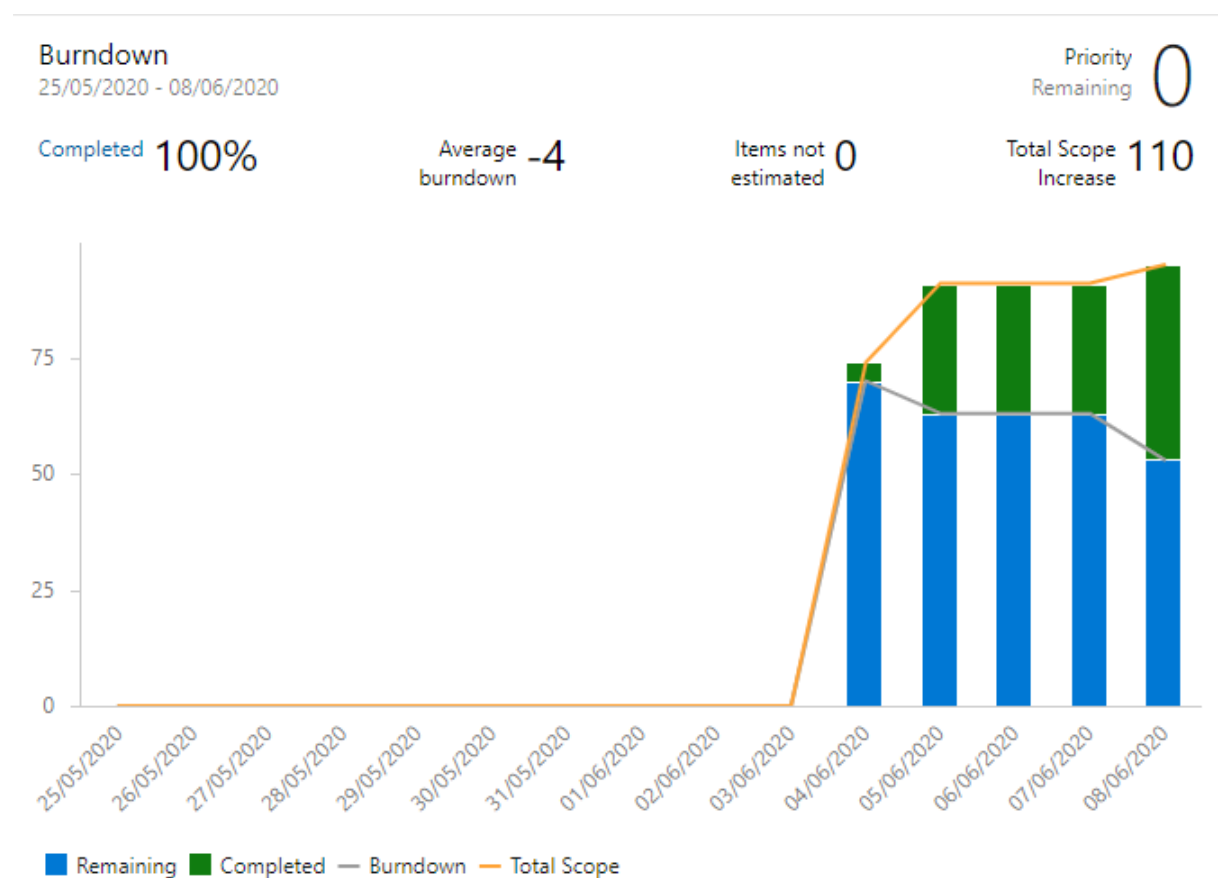
Para o gerenciamento das tarefas, foi utilizado inicialmente o Dev Azure onde foram alocadas as atividades e feitas as atribuições para cada membro do time, com a plataforma do Azure, foi possível ter uma melhor visualização do andamento das atividades que estavam sendo realizadas e que estavam pendentes ou atrasadas.



Figura_01: Kanban na plataforma Azure

2. Burndown Chart

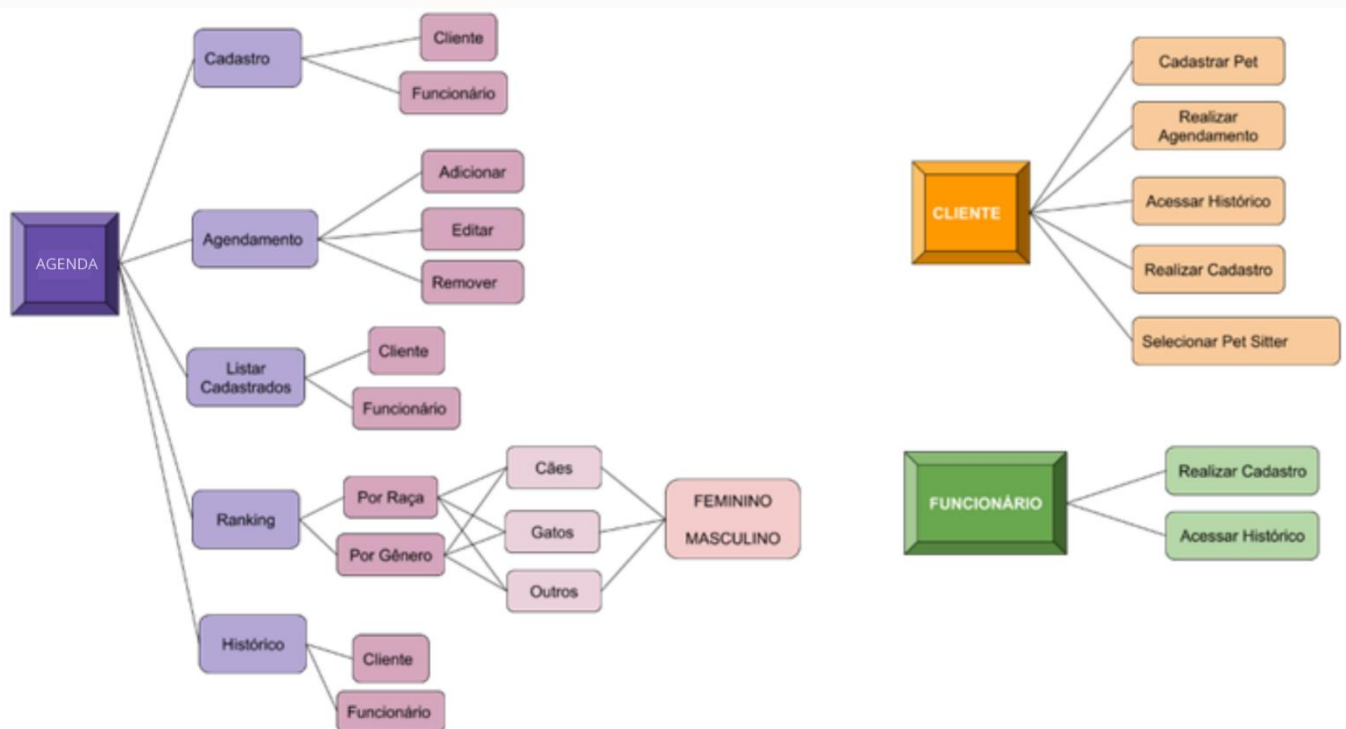
A figura a seguir apresenta o *Burndown Chart* do TS02 para a *Sprint 06*. É possível perceber claramente a formação de uma *barriga*, onde a velocidade real se distancia muito da velocidade ideal. Este comportamento demonstra uma antecipação do time no desenvolvimento do projeto, algumas atividades foram adiantadas para que o time pudesse dar andamento nas outras entregas, por conta da proximidade das datas.



Figura_02: Gráfico BurnDown

3. Arquitetura Básica

A modelagem das Classes Cliente, Funcionário e Agenda foram destacados na arquitetura pois possuem os métodos principais do programa, abaixo foram listadas cada funcionalidade do cliente e do funcionário dentro do sistema e a responsabilidade dos cadastramentos, acesso a históricos, sistema de ranqueamento e sistema de agendamentos, é a classe Agenda, quem tem ligação direta com a classe principal do sistema.



Figura_03:Arquitetura do Software Pet Sitter

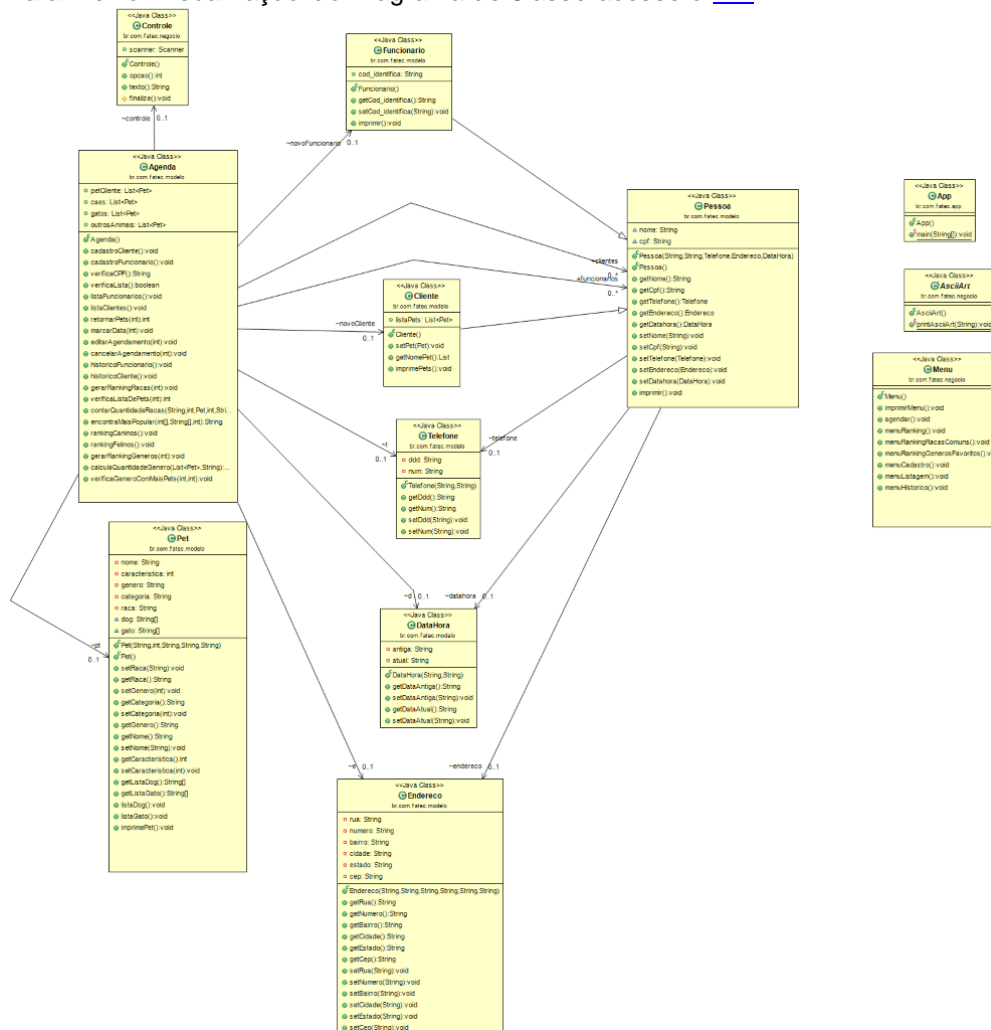
4. Modelagem do Diagrama de Classes

A modelagem do Diagrama de Classe foi realizada dentro do Editor de Texto Eclipse, com uma ferramenta adicional chamada ObjectAID, que permite gerar de forma automática um Diagrama de Classes.

As principais funções do programa se encontram na Classe Agenda, como foi mostrado na modelagem do sistema (figura_03), o sistema das classes utiliza o conceito do Polimorfismo, onde as funções das classes e seus atributos são reutilizados dentro de outras classes.

As classes Funcionário e Cliente utilizam o conceito da Herança, herdando atributos e métodos da super classe Pessoa.

Para melhor visualização do Diagrama de Classe acesse o [link](#).

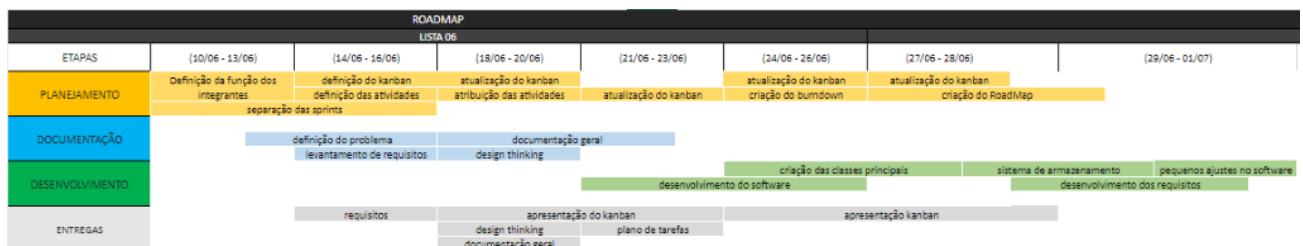


Figura_04: Diagrama de Classes

5. Desenvolvimento

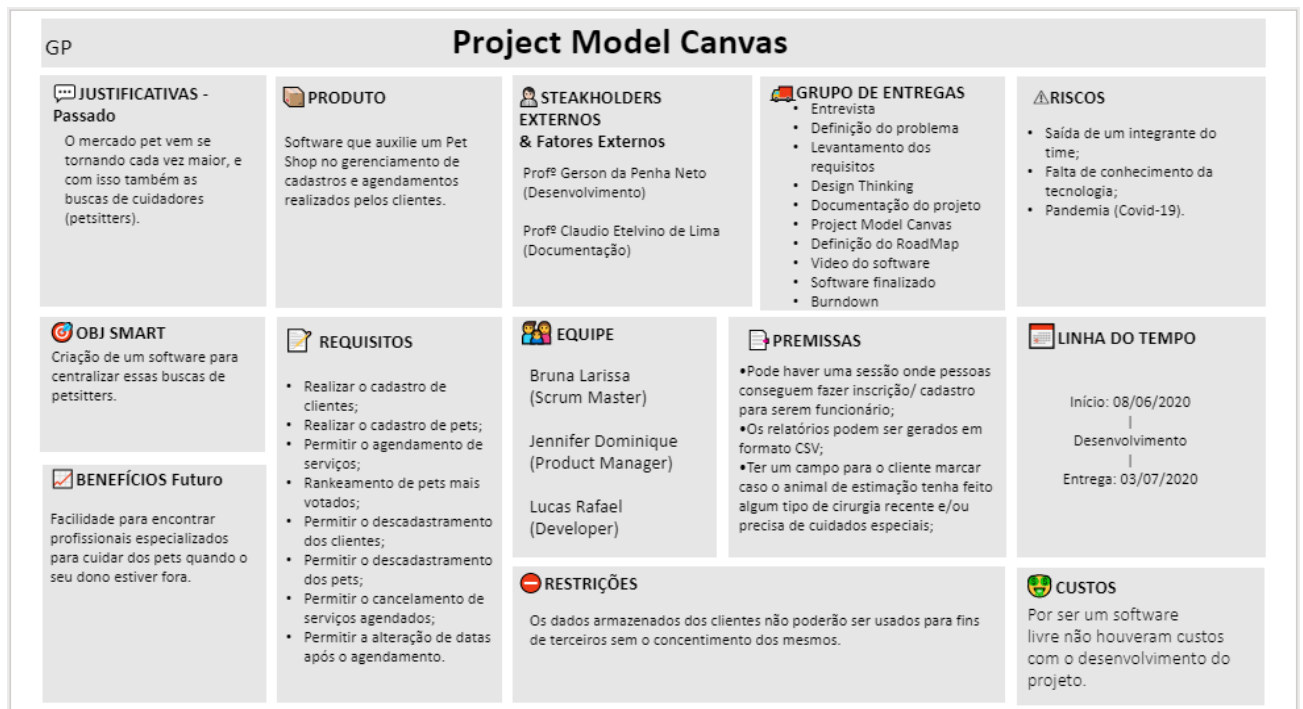
5.1 Mapeamento dos processos

Os requisitos levantados serviram como base para o desenvolvimento do Mapeamento, onde foi possível realizar a atribuição das atividades para cada membro do time, desde a documentação à implementação do projeto. O modelo desenvolvido para o acompanhamento das atividades foi o Kanban na plataforma Azure, com base nos resultados, foi desenvolvido um RoadMap, permitindo uma melhor visualização do andamento das Sprints e um Project Model Canvas, melhorando o planejamento do software para uma aplicação no mercado e um melhor planejamento futuro das atividades e implementações.



Figura_05: RoadMap do projeto

Para acessar o Project Model Canvas e ter uma melhor visualização acesse o [link](#).

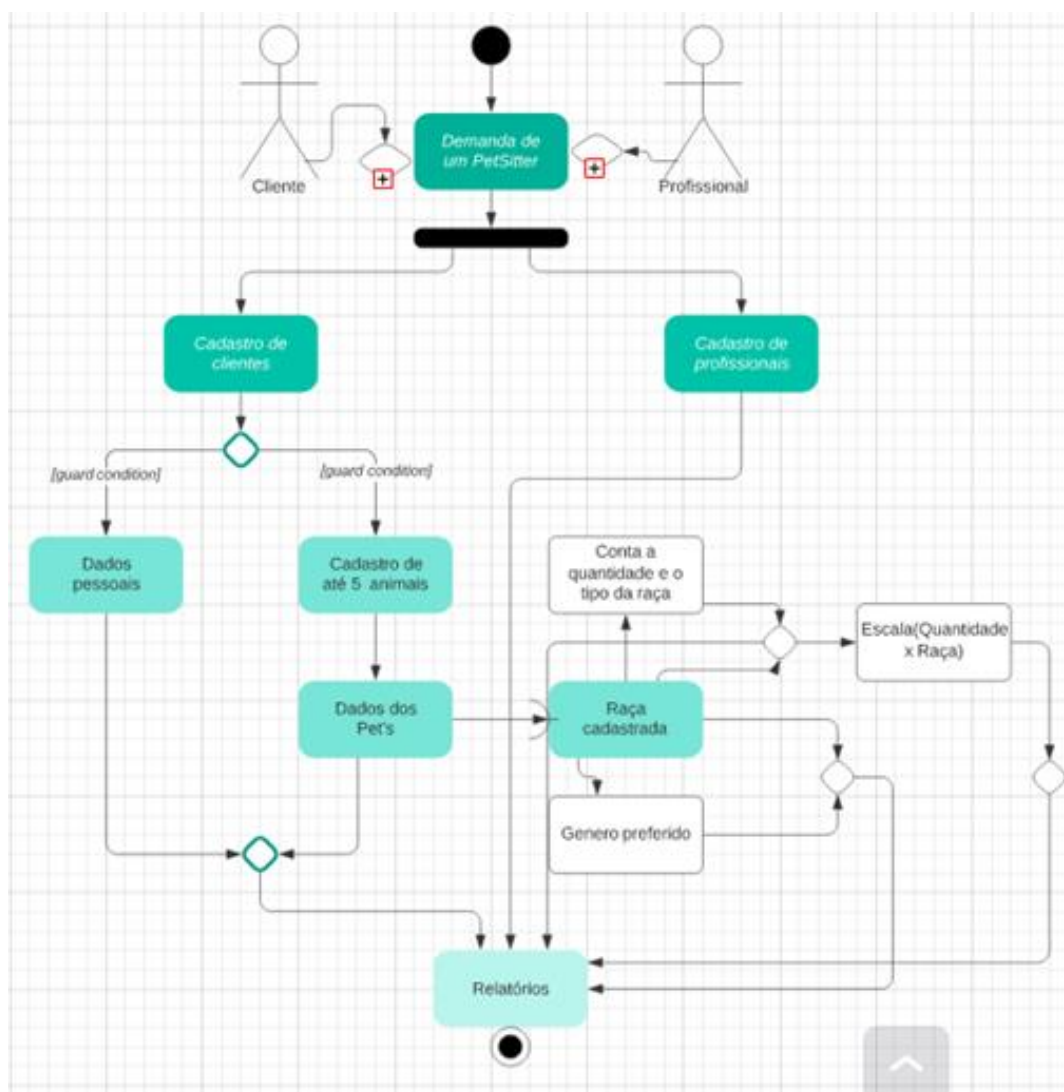


Figura_06: Project Model Canvas do Projeto

5.2 Caso de Uso

O diagrama de Caso de Uso do projeto, visa demonstrar as ações do cliente e do funcionário dentro da plataforma, com as possibilidades de:

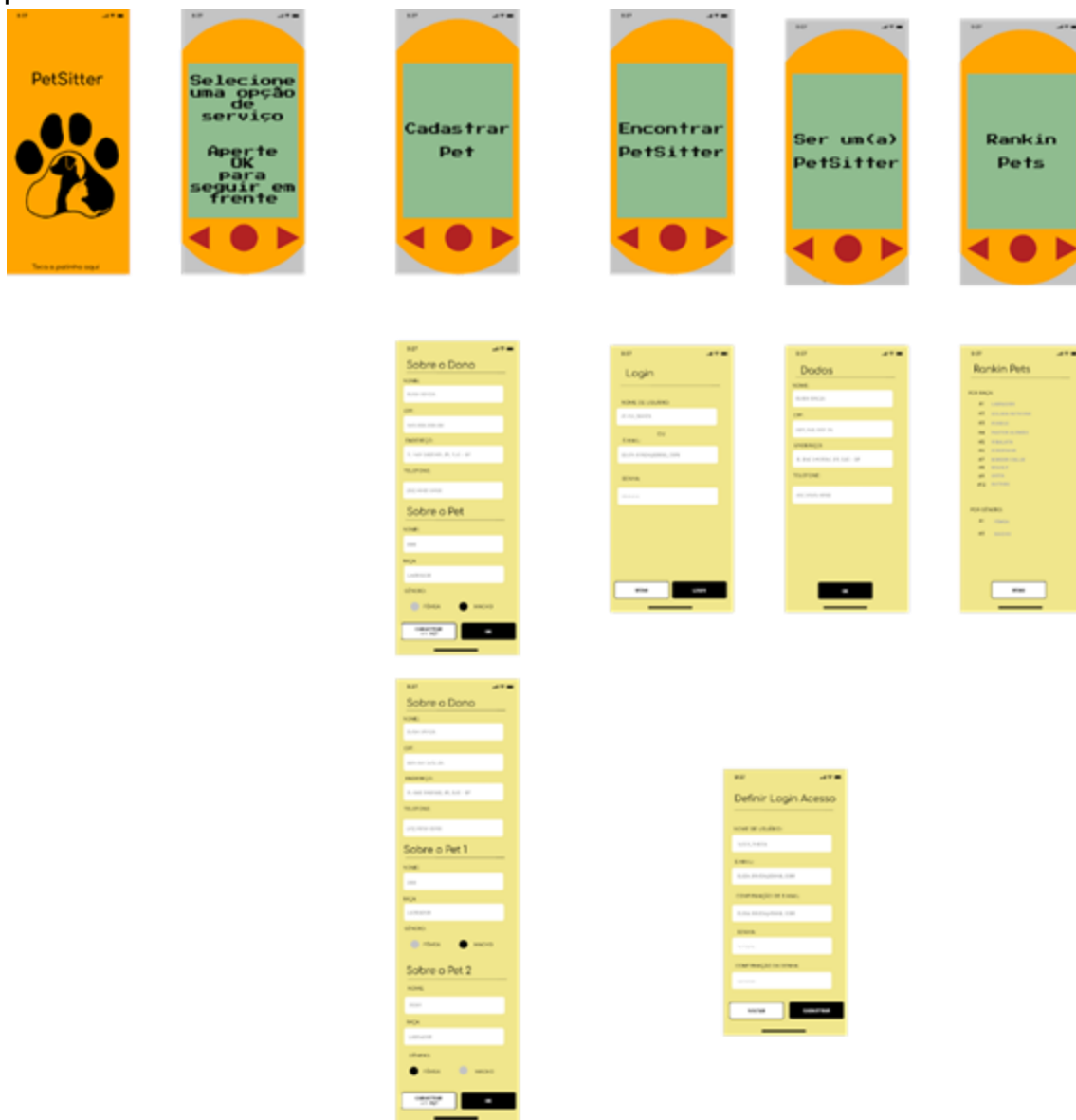
- Realização de Cadastro
- Acesso aos dados e histórico
- Cadastro de Pets
- Acesso ao Ranking de pets mais comuns no petshop
- Acesso a informações de quantos clientes foram atendidos
- Agendamento de serviços para os Pets



Figura_07: Diagrama de Caso de Uso Simples

5.3 Prototipação

A prototipação foi desenvolvida na plataforma do Figma, com um modelo de aplicação mobile, visando a mobilidade e acessibilidade do usuário ao software e aos serviços disponibilizados pelo PetSitter. O protótipo foi utilizado como base para visualizar as necessidades do cliente durante o uso, com ele foi possível planejar algumas telas adicionais para o sistema do PetSitter e trazer mais informação ao cliente sobre seus agendamentos e ao funcionário uma melhor noção de quantos cliente foram atendidos no dia.



Figura_08: Protótipo para facilitação da modelagem do sistema

Para ter interação com o protótipo acesse o [link](#).

5.4 Ferramentas Case

Abaixo estão listadas as ferramentas utilizadas no desenvolvimento do projeto, desde sua inicialização com as respectivas documentações, até a sua finalização com o software finalizado.



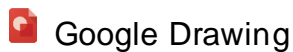
Editor de texto para a programação e o desenvolvimento em Java do projeto.



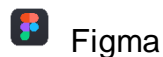
Utilizado no início do desenvolvimento do projeto como administrador de tarefas, durante a quinta Sprint as tarefas foram remanejadas para a plataforma do Azure.



Ferramenta para o acompanhamento da evolução do projeto.



Modelagem dos tópicos de Arquitetura Básica e o diagrama de Caso de Uso.



Utilizado para a criação do protótipo do projeto.

5.3. Atividades Individuais Realizadas no Entrega:

Bruna Larissa Clemente Gomes <i>ScrumMaster</i>	<ul style="list-style-type: none">✓ Atribuição dos membros em cada atividade da Sprint✓ Definição da prioridade das tarefas✓ Desenvolvimento do Relatório Sintético✓ Criação do Design Thinking✓ Desenvolvimento do Projeto
Jennifer Dominique Dias Santos <i>Product Manager</i>	<ul style="list-style-type: none">✓ Definição das atividades da Sprint✓ Desenvolvimento do Relatório Sintético✓ Desenvolvimento do software✓ Criação do Design Thinking✓ Prototipação do projeto
Lucas Rafael de Andrade Barroso <i>Developer</i>	<ul style="list-style-type: none">✓ Desenvolvimento do modelo do Design Thinking✓ Desenvolvimento do software✓ Modelagem do Caso de Uso✓ Desenvolvimento do Relatório Sintético

6. Recomendações

Durante o desenvolvimento do projeto o time teve um pouco de dificuldade na execução das tarefas, por conta do conteúdo de outras matérias que complementariam o conhecimento para o desenvolvimento. Em resposta a isso, o time priorizou as pequenas tarefas, principalmente as funções mais básicas do sistema e/ou que já possuíam conhecimento o suficiente para a efetuação do desenvolvimento.

7. Referências

- <https://www.ic.unicamp.br/~islene/mc514/prod-cons/prod-cons.pdf> , acessado em junho/2020
- <https://www.devmedia.com.br/explorando-a-classe-arraylist-no-java/24298> , acessado em junho/2020
- PRESSMAN, “Software Engineering: A Practitioner’s Approach”, 6a Ed. McGraw Hill, 2005 – Apresenta uma visão geral da Engenharia de Software voltada para o profissional da área; completo e atualizado
- PFLEEGER, “Engenharia de Software – Teoria e Prática”, Prentice Hall, 2004 – Equivalente ao anterior.
- SOMMERVILLE, “Engenharia de Software”, Addison Wesley, 2003 – Boa apresentação da Engenharia de Software, com os métodos orientados a objetos. Engenharia de Software II
- SCHACH, “Engenharia de Software: Os Paradigmas Clássico e Orientado a Objetos. São Paulo: McGraw Hill, 2009 – Apresentação excelente da ES.
- PAULA FILHO, “Engenharia de Software – Fundamentos, Métodos e Padrões”, LTC, 2001 – Apresentação da Engenharia de Software, com realce para a documentação e suas normas.
- LARMAN, “Utilizando UML e Padrões”, Bookman, 2000 – Apresentação sintética da metodologia orientada a objetos utilizando padrões.