

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Edson dos Santos Junior

Leandro Henrique Marini

**Sistema de Gerenciamento de Animais de Estimação**

**Americana, SP**

**2022**

---

**Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Edson dos Santos Junior  
Leandro Henrique Marini

**Sistema de Gerenciamento de Animais de Estimação**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Esp. Antônio Alfredo Lacerda.

Área de atuação: Sistema de Informação.

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana -  
CEETEPS  
Dados Internacionais de Catalogação-na-fonte**

B876d SANTOS JUNIOR, Edson dos

Sistema de gerenciamento de animais de estimação. /  
Edson dos Santos Junior, Leandro Henrique Marini. – Americana,  
2022.

61f.

Monografia (Curso Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de  
Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientadores: Prof. Antonio Alfredo Lacerda

1 I MARINI, Leandro Henrique II. LACERDA, Antonio Alfredo  
III. Centro Estadual de Educação Tecnológica Paula Souza –  
Faculdade de Tecnologia de Americana

CDU: 681.3.05

**Edson dos Santos Junior**

**Leandro Henrique Marini**

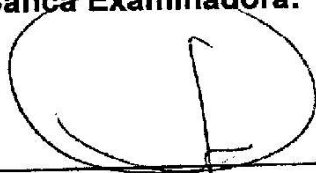
**Sistemas de Gerenciamento de Animais de  
Estimação**

Trabalho de Conclusão de Curso  
apresentado à Faculdade de Tecnologia  
de Americana como parte dos requisitos  
para obtenção do Título de Tecnólogo em  
Análise e Desenvolvimento de Sistemas  
pelo Centro Paula Souza.

Área de Atuação : Sistema de Informação

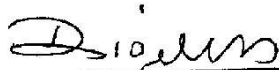
Americana, 22 de junho de 2022.

**Banca Examinadora:**



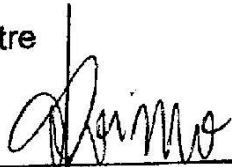
Antonio Alfredo Lacerda

Especialista



Diogenes de Oliveira

Mestre



Adriano Cilhos Doimo

Especialista

## RESUMO

O Projeto a seguir se trata do desenvolvimento de um sistema que visa o gerenciamento de gastos de animais de estimação, tendo como a redução e acompanhamento de tratamentos e custos. Desta forma, todas as etapas do desenvolvimento foram explanadas nesta documentação. A modelagem do sistema foi realizada usando a linguagem UML, e a codificação foi feita utilizando linguagem *Dart* junto de uma *WebApi* na tecnologia *Asp.Net Core*. O Projeto desenvolvido apresenta melhorias com os demais existentes no mercado.

**Palavras-Chave:** Sistema; Gerenciamento; Pets

## **ABSTRACT**

*The following project is about the development of a system that aims to manage pet expenses, with the reduction and monitoring of treatments and costs. Thus, all development stages were explained in this documentation. The system modeling was performed using the UML language, and the coding was done using Dart language along with a WebApi in Asp.Net Core technology. The developed Project presents improvements with the others existing in the market.*

**Keywords:** *System; Management; Pets.*

# SUMÁRIO

1	INTRODUÇÃO .....	8
1.1	OBJETIVOS ESPECÍFICOS.....	9
1.2	JUSTIFICATIVA.....	10
2.	CONCEITUANDO SISTEMAS E GERENCIAMENTO .....	11
2.1	SOFTWARES SIMILARES .....	12
3.	ENGENHARIA DE SOFTWARE .....	14
3.1	LEVANTAMENTO DE REQUISITOS.....	14
3.1.1	REQUISITOS FUNCIONAIS .....	14
3.1.2	REQUISITOS NÃO FUNCIONAIS .....	15
3.2	DIAGRAMA CASO DE USO.....	15
3.2.1	DOCUMENTAÇÃO DOS CASOS DE USO .....	17
3.3	BANCO DE DADOS .....	27
3.3.1	DIAGRAMA ENTIDADE RELACIONAMENTO .....	27
3.3.3	DICIONÁRIO DE DADOS .....	29
4.	PROJETO .....	36
4.1	TECNOLOGIA EMPREGADA .....	36
4.2	PRINCIPAIS INTERFACES DO PROJETO.....	37
4.3	RECURSOS E FERRAMENTAS .....	58
5.	CONCLUSÃO.....	59
6.	REFERÊNCIAS.....	60

## Lista de Figuras

Figura 1 - Estrutura básica dos sistemas de informação.....	11
Figura 2 - Diagrama de casos de uso .....	16
Figura 3 - Diagrama Entidade Relacionamento.....	28
Figura 4 - Representação do objeto Usuário.....	33
Figura 5 - Representação do objeto Animal .....	34
Figura 6 - Representação do objeto Vacina .....	34
Figura 7 - Representação do objeto Despesa .....	35
Figura 8 - Representação do objeto Compromisso .....	35
Figura 9 - Representação do objeto Indicador .....	35
Figura 10 - Arquitetura do sistema .....	36
Figura 11 - Interface Login .....	38
Figura 12 - Interface de criação do usuário .....	39
Figura 13 - Interface Homepage.....	40
Figura 14 - Interface de listagem de Pets.....	42
Figura 15 - Interface criação/edição de pet .....	44
Figura 16 - Interface de escolha de espécie .....	46
Figura 17 - Interface de apresentação do pet.....	47
Figura 18 - Modais de nova despesa, vacina e compromisso.....	49
Figura 19 - Resultado cadastro de despesa, vacina e compromisso .....	50
Figura 20 - Editar ou excluir despesa.....	51
Figura 21 - Interface evolução de peso e altura do pet .....	52
Figura 22 - Cadastro de peso e resultado .....	54
Figura 23 - Editar ou excluir de registro de indicador .....	55
Figura 24 - Interface de despesas.....	56
Figura 26 - Interface de todas as despesas .....	57



## Lista de Tabelas

Tabela 1 - Comparativo de funcionalidades da aplicação MyPets em relação aos concorrentes.....	13
Tabela 2 - Requisitos funcionais do projeto. ....	14
Tabela 3 - Requisitos não funcionais do projeto.....	15
Tabela 4 - Caso de uso “Fazer Login - UC01”.....	17
Tabela 5 – Caso de uso “Fazer Registro - UC02” . ....	18
Tabela 6 – Caso de uso “Manter usuário - UC03”.....	19
Tabela 7 – Caso de uso “Consultar pet shops - UC04”.....	20
Tabela 8 – Caso de uso “Manter Pets - UC05”. ....	21
Tabela 9 – Caso de uso “Manter vacinas - UC06”. ....	22
Tabela 10 – Caso de uso “Manter agenda - UC07”.....	23
Tabela 11 – Caso de uso “Manter despesas - UC08”. ....	24
Tabela 12 – Caso de uso “Manter indicadores - UC09”. ....	25
Tabela 13 – Caso de uso “Gerar PDF do pet - UC10”. ....	26
Tabela 14 - Dicionário de dados da entidade Usuário.....	30
Tabela 15 - Dicionário de dados da entidade Animal.....	30
Tabela 16 - Dicionário de dados da entidade Vacina.....	31
Tabela 17 - Dicionário de dados da entidade Despesa.....	31
Tabela 18 - Dicionário de dados da entidade Compromisso.....	32
Tabela 19 - Dicionário de dados da entidade Indicador. ....	32

# 1 INTRODUÇÃO

Com inúmeras aplicações sendo desenvolvidas para atender diversos segmentos do mercado, os Pets não poderiam permanecer sem uma aplicação que venha a contribuir no gerenciamento e acompanhamento dos custos, além de um acompanhamento clínico do animal.

A ideia é proporcionar um controle para as pessoas que possuem Pets de estimação em que seja possível registrar dados do Pet como, despesas, consultas médicas, fotos, permitindo relatórios e mostrando ao usuário a evolução do seu Pet ao longo da vida registrada no sistema, concentrando as informações em um só lugar e as tornando mais acessíveis aos donos, visto que os animais estão cada vez mais presentes na vida do homem.

Este Projeto está distribuído da seguinte forma: Capítulo 1 – Introdução, Capítulo 2 – Conceituando sistemas, Capítulo 3 – Engenharia de Software, Capítulo 4 – Projeto MyPets, Capítulo 5 – Conclusão e Capítulo 6 – Referências.

## **1.1 OBJETIVOS ESPECÍFICOS**

Identificar as dificuldades do homem no gerenciamento e registrar as informações geradas como despesas, cuidados médicos, imagens, estabelecendo um banco clínico do animal, na clínica, a fim de contribuir nos diagnósticos no transcorrer da vida dos seus Pets de estimação para idealizar, levantando requisitos e funcionalidades de sistemas concorrentes para desenvolver um sistema.

## 1.2 JUSTIFICATIVA

Os animais estão próximos do homem desde a pré-história, período que decorreu entre 3500 a.C. e 3000 d.C., quando eram retratados nas pinturas rupestres das cavernas e com o passar do tempo se tornaram cada vez mais presentes e influenciáveis na vida do homem.

A domesticação dos animais na cultura humana começou com a sua criação para o auxílio na produção de alimentos, para transporte de pessoas ou cargas e até mesmo para cuidados com os terrenos para agricultura, ainda na pré-história, há cerca de 2800 anos atrás.

Nos dias atuais, os animais domésticos deixaram de ser usados para fins de trabalho e passaram a fazer parte do cotidiano das famílias, sendo em muitos lares considerado como um membro como qualquer outro ser humano, e como consequência, esse animal precisa de cuidados como: gerenciamento de despesas e cuidados médicos.

Justifica-se, portanto, uma solução para o cuidado destes animais, o trabalho a seguir detalha o processo de desenvolvimento do sistema **MyPets**, para ajudar os donos de Pets a registrar as informações dos seus Pets.

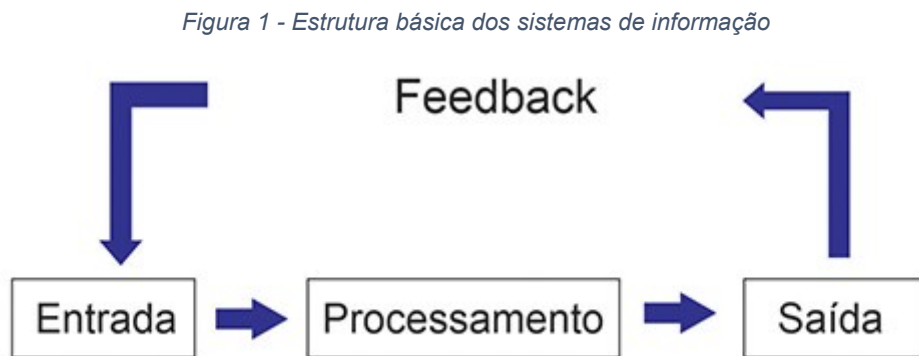
## 2. CONCEITUANDO SISTEMAS E GERENCIAMENTO

Um sistema pode ser definido como um conjunto de elementos que possuem certas responsabilidades, são interdependentes e comunicam entre si para juntos desempenhar um objetivo maior solucionando algum problema encontrado.

Assim, um sistema de informação pode ser representado por um conjunto de componentes pessoas, hardware, software, redes de comunicações e conjunto de dados que são imputados, processados, transformados em informações, armazenados e disponibilizados aos usuários.

“Sistemas de informação é um conjunto organizado de pessoas, hardware, software, rede de comunicação e dados, que são coletados e transformados em informações dentro de um ambiente organizacional”. (O'BRIEN, 2004).

A Figura 1 abaixo demonstra a estrutura básica do funcionamento de sistemas de informação:



*Fonte – Site Gestão de Segurança Privada.*

Desta forma, os sistemas são utilizados por usuários e empresas para que seja possível tomar uma decisão com base na interpretação da informação devolvida pelo sistema, servindo de apoio para o gerenciamento do negócio que visa planejar, organizar, dirigir e controlar os processos internos da organização.

## 2.1 SOFTWARES SIMILARES

A finalidade deste Projeto, não é concorrer com os aplicativos existente no mercado e sim apresentar módulos soluções que outros não comportam tornando um aplicativo completo no quesito do gerenciamento de despesas. Atualmente existem diversos aplicativos voltados para o gerenciamento de Pets, foram selecionadas as três aplicações mais populares e mais bem avaliadas pelos usuários da Play Store, são eles:

- **Pet Booking:** aplicativo que oferece agendamentos em Petshops, clínicas veterinárias, e até hotel, para os Pets com base na localização do usuário. Aplicação com nota 4,6 na Google Play (GOOGLE PLAY, 2022a).
- **11 Pets: Cuidado animal:** aplicativo que oferece recursos de lembretes para cuidados como banho, organização de dados médicos e acompanhamento de incidentes. Aplicação com nota 4,5 na Google Play (GOOGLE PLAY, 2022b).
- **Dog Health:** aplicativo que permite controlar os dados a respeito da saúde de um pet, memorizar seu histórico e lembrar medicação agendada. Aplicação com nota 4,0 na Google Play (GOOGLE PLAY, 2022c).

Levando estes aspectos em consideração, foi elaborada a Tabela 19 mostrando as principais diferenças do **MyPets**, em relação aos aplicativos citados anteriormente:

Tabela 1 - Comparativo de funcionalidades da aplicação MyPets em relação aos concorrentes.

Funcionalidades	11Pets: Cuidado animal	Dog Health	Pet Health	MyPets
Estabelecimentos próximos para cuidar do pet	X	X	X	X
Gerenciamento das despesas do pet	-	-	-	X
Gerenciamento das vacinas do pet	X	X	X	X
Gerenciamento de indicadores de peso e altura	X	-	X	X
Gerenciar dados do usuário	X	X	X	X
Exportar dados do pet em PDF - <i>Portable Document File</i>	-	-	-	X
Ser multiplataforma	X	X	-	X

Fonte: Elaborado pelo autor (2022).

### 3. ENGENHARIA DE SOFTWARE

A Engenharia de Software é definida como um processo de analisar os requisitos do usuário e, em seguida, projetar, construir e testar aplicativos de software que satisfaçam esses requisitos, documentando os processos para que seja possível desenvolver, operar e manter o sistema.

#### 3.1 LEVANTAMENTO DE REQUISITOS

A engenharia de requisitos (RE – *Requirements Engineering*) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

##### 3.1.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 2 apresenta os requisitos funcionais deste projeto.

*Tabela 2 - Requisitos funcionais do projeto.*

Identificação	Requisito Funcional	Prioridade
RF001	Manter usuário	Essencial
RF002	Manter Pets	Essencial
RF003	Acompanhar evolução de altura e peso de Pets	Essencial
RF004	Mostrar estabelecimentos como pet shops próximos da localização do usuário	Desejável
RF005	Manter vacinas dos Pets	Essencial
RF006	Manter gastos dos Pets	Desejável
RF007	Notificar um evento próximo como consulta ou vacina	Importante

*Fonte: Elaborado pelo autor (2022).*



### 3.1.2 REQUISITOS NÃO FUNCIONAIS

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 3 apresenta os requisitos não funcionais deste projeto.

*Tabela 3 - Requisitos não funcionais do projeto*

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Arquitetura MVC.	Padrões	Essencial
RNF002	Banco de dados NoSQL	Desempenho	Essencial
RNF003	Framework Flutter	Hardware e Software	Essencial
RNF004	Material Design	Padrões	Essencial
RNF005	Implementação LGPD	Segurança	Essencial
RNF006	Tecnologia ASP.Net Core.	Hardware e Software	Essencial
RNF007	Aplicativo disponível 24/7	Confiabilidade	Essencial

*Fonte: Elaborado pelo autor (2022).*

### 3.2 DIAGRAMA CASO DE USO

O diagrama de caso de uso descreve o cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do diagrama são retratados os atores (representado por bonecos), as funcionalidades (representadas por balões com a ação escrita por dentro) e as relações (representadas pelas linhas). (GUEDES, 2005).

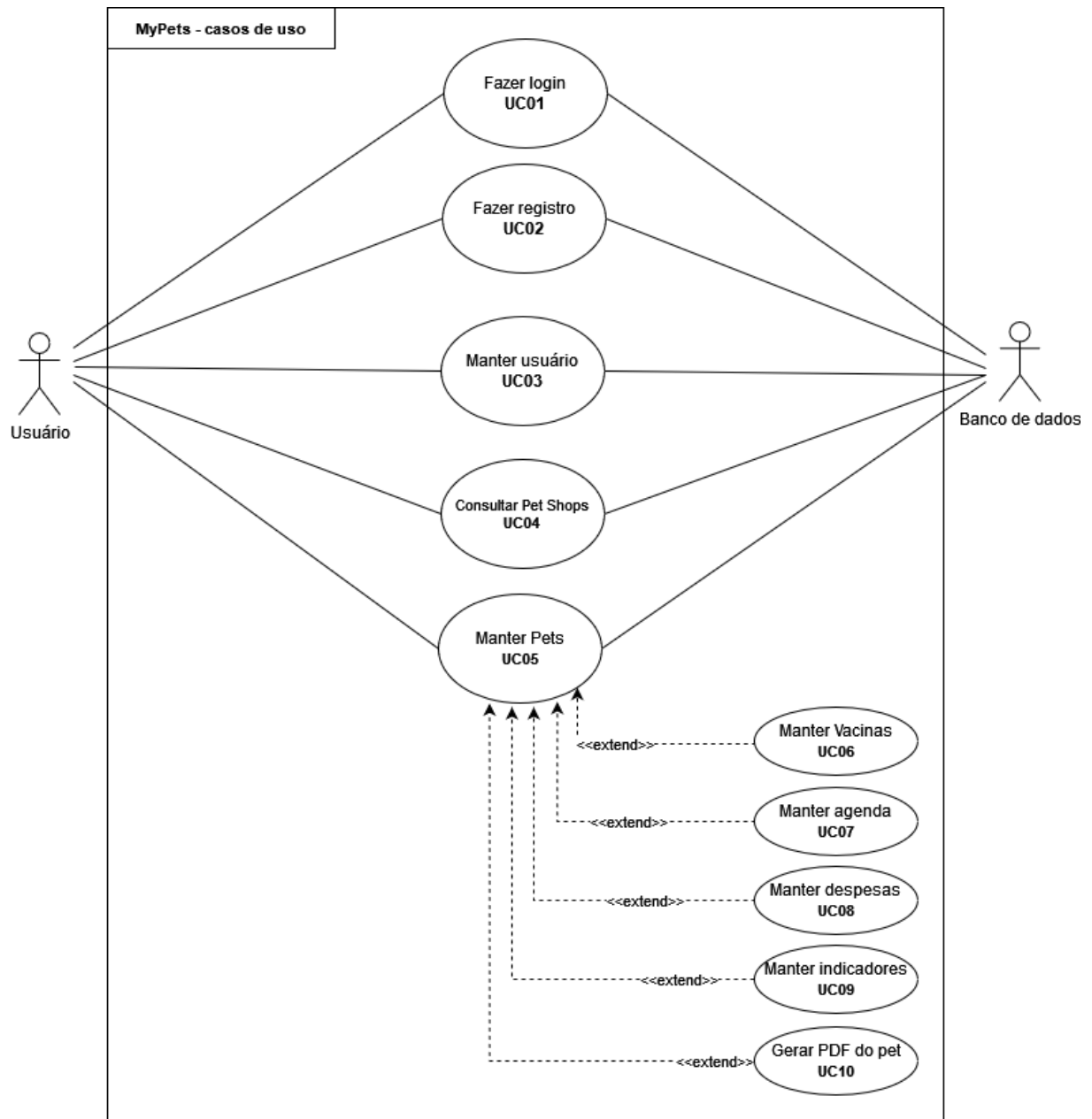
Os atores que interagem com o sistema são: o usuário e o *MongoDB*, Banco de Dados que o sistema utiliza. O sistema é um caso de uso explícito e se trata do sistema em si em que os casos de uso acontecem.

- **Usuário:** ator que representa os utilizadores deste aplicativo. Um ator pode, por exemplo, buscar Pets, editar, favoritar, entre outros.
- **Places Api:** serviço do Google que retorna informações de lugares usando requisições HTTP.

- **MongoDB:** Banco de Dados do sistema, onde o sistema armazena as informações dos usuários e dos medicamentos.

A seguir é exibida a Figura que retrata os casos de uso do sistema:

Figura 2 - Diagrama de casos de uso



Fonte: Elaborado pelo autor (2022).

O diagrama acima descreve todos os casos de uso presentes no sistema, iniciados por meio de ações do usuário (representado pelo boneco à direita) e que interagem com o banco de dados (representado à esquerda). Alguns dos

casos de uso descritos acima (UC06 à UC10) dependem previamente do caso de uso UC05 para ser executado, desta forma estes estendem o caso de uso UC05, ou seja, poderão somente ser executados após o caso de uso UC05 ser executado.

A seguir será apresentado a documentação dos casos de uso do Projeto.

### 3.2.1 DOCUMENTAÇÃO DOS CASOS DE USO

Cada funcionalidade do diagrama de caso de uso será descrita da Tabela 1 à Tabela 7.

*Tabela 4 - Caso de uso "Fazer Login - UC01".*

<b>Nome do caso de uso</b>	Fazer Login - UC01
<b>Atores envolvidos</b>	Usuário, sistema e banco de dados
<b>Objetivo</b>	Descrever o fluxo de login.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Inserir o <i>email</i> , senha e pressiona o botão de fazer o login	
	2. Verificar se existe uma conta com o e-mail inserido. e caso haja, comparar a senha inserida com a senha da conta encontrada.
	3. Caso a verificação tenha retornado sucesso, redireciona o usuário para a interface inicial, caso não, exibe uma mensagem com o motivo do erro.
<b>Validações</b>	Verifica se se há uma conta cadastrada para o e-mail inserido; verifica se a senha inserida corresponde a senha cadastrada.

*Fonte: Elaborado pelo autor (2022).*

Tabela 5 – Caso de uso “Fazer Registro - UC02”.

<b>Nome do caso de uso</b>	Fazer Registro - UC02
<b>Atores envolvidos</b>	Usuário, sistema e banco de dados
<b>Objetivo</b>	Descrever o fluxo de registro de uma nova conta.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Pressionar o botão de cadastrar nova conta.	
	2. Exibir a interface de novo usuário.
3. Preencher os campos de nome, <i>email</i> e senha.	
	3. Realizar a validação das informações.
	4. Com base no resultado da validação, criar ou não a nova conta.
	5. Retornar o resultado da operação para o usuário.
<b>Validações</b>	Verificar se já não há uma conta cadastrada para o <i>email</i> inserido, e se a senha possui mais de 8 caracteres contendo letras e números.

Fonte: Elaborado pelo autor (2022).

Tabela 6 – Caso de uso “Manter usuário - UC03”.

<b>Nome do caso de uso</b>	Manter dependentes – UC03
<b>Atores envolvidos</b>	Usuário, sistema e banco de dados.
<b>Objetivo</b>	Descreve a ação de editar os dados da conta do usuário cadastrada.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Pressionar o botão “meu perfil”.	
	2. Redirecionar o usuário para a interface “Meu perfil”.
3. Na interface, pressionar o botão com o ícone editar.	
	4. Exibir um modal com os campos de <i>email</i> , nome e um botão para alterar a senha.
5. No modal, alterar os dados dos campos e pressionar o botão salvar.	
	6. Validar as informações e com base no resultado da validação salvar ou não os dados alterados.
	7. Retornar o resultado da operação.
<b>Validações</b>	Validar se campo nome (não está vazio), se o campo e-mail contém um texto que representa um e-mail e caso a operação for troca de senha, se campo de senha atual corresponde a senha atual da conta e se os campos de nova senha e confirme nova senha são iguais.

Fonte: Elaborado pelo autor (2022).

Tabela 7 – Caso de uso “Consultar pet shops - UC04”.

<b>Nome do caso de uso</b>	Consultar pet shops - UC04
<b>Atores envolvidos</b>	Usuário, sistema, e serviço <i>Places API</i> do Google
<b>Objetivo</b>	Descreve a ação consultar pet shops na região.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Pressionar o botão de estabelecimentos próximos	
	2. Exibir a interface de estabelecimentos próximos
	3. Enquanto o serviço de localização do dispositivo não estiver ativo, exibir uma mensagem para o usuário ativar o serviço de localização
4. Ativar o serviço de localização do dispositivo.	
	5. Realizar a consulta enviando os parâmetros da localização do usuário para a <i>PlacesAPI</i>
	6. Retornar o resultado da consulta para o usuário.
<b>Validações</b>	Validar se o serviço de localização do dispositivo está ativado.

Fonte: Elaborado pelo autor (2022).

Tabela 8 – Caso de uso “Manter Pets - UC05”.

<b>Nome do caso de uso</b>	Manter Pets - UC05
<b>Atores envolvidos</b>	Usuário, sistema, e banco de dados.
<b>Objetivo</b>	Descreve a ação de criar/editar um pet.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Pressionar o botão de editar pet, ou novo pet.	
	2. Verificar se é edição de um pet, caso seja buscar os dados do pet.
3. Preencher os campos de imagem, nome, espécie, peso, data de nascimento e sexo do pet.	
	4. Validar as informações e com base no resultado da validação salvar ou não os dados.
	5. Retornar o resultado da operação para o usuário.
<b>Validações</b>	Validar se o pet possui uma foto, nome, espécie, peso, data de nascimento e sexo.

Fonte: Elaborado pelo autor (2022).

Tabela 9 – Caso de uso “Manter vacinas - UC06”.

<b>Nome do caso de uso</b>	Manter vacinas- UC06
<b>Atores envolvidos</b>	Usuário, sistema, e banco de dados.
<b>Objetivo</b>	Descreve a ação de criar/editar nova vacina de um pet.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Acessar a interface que mostra os dados do pet	
2. Pressionar o botão de editar ou criar novo registro de vacina	
	3. Verificar se é a edição de um registro, caso sim, buscar os dados do registro.
4. Preencher os campos de nome, data de aplicação e caso tenha data de renovação da vacina.	
	5. Validar as informações e com base no resultado da validação salvar ou não os dados.
	6. Retornar o resultado da operação para o usuário.
<b>Validações</b>	Validar se o campo nome não está vazio, se a data de aplicação não é maior do que a data de hoje e se a data de renovação não é anterior à data de aplicação, caso exista.

Fonte: Elaborado pelo autor (2022).



Tabela 10 – Caso de uso “Manter agenda - UC07”.

<b>Nome do caso de uso</b>	Manter agenda - UC07
<b>Atores envolvidos</b>	Usuário, sistema, e banco de dados.
<b>Objetivo</b>	Descreve a ação de criar/editar um compromisso do pet.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Acessar a interface que mostra os dados do pet	
2. Pressiona o botão de editar ou criar novo registro de compromisso	
	3. Verificar se é a edição de um registro, caso sim, buscar os dados do registro.
4. Preencher os campos de nome, descrição, local, data e horário do compromisso.	
	5. Validar as informações e com base no resultado da validação salva ou não os dados.
	6. Retornar o resultado da operação para o usuário.
<b>Validações</b>	Validar se o campo nome e local não está vazio, e se a data e horário são válidos.

Fonte: Elaborado pelo autor (2022).

Tabela 11 – Caso de uso “Manter despesas - UC08”.

<b>Nome do caso de uso</b>	Manter despesas - UC08
<b>Atores envolvidos</b>	Usuário, sistema, e banco de dados.
<b>Objetivo</b>	Descreve a ação de criar/editar uma despesa.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Acessar a interface que mostra os dados do pet	
2. Pressionar o botão de editar ou criar novo registro de despesa	
	3. Verificar se é a edição de um registro, caso sim, buscar os dados do registro.
4. Preencher os campos de nome, descrição, valor e data da despesa.	
	4. Validar as informações e com base no resultado da validação salva ou não os dados.
	5. Retornar o resultado da operação para o usuário.
<b>Validações</b>	Validar se campo nome não está vazio, o campo valor possui apenas números e se a data não é posterior a data de hoje.

Fonte: Elaborado pelo autor (2022).

Tabela 12 – Caso de uso “Manter indicadores - UC09”.

<b>Nome do caso de uso</b>	Manter indicadores - UC09
<b>Atores envolvidos</b>	Usuário, sistema, e banco de dados.
<b>Objetivo</b>	Descreve a ação de criar/editar uma despesa.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Acessar a interface que mostra os dados do pet	
2. Pressionar o botão para acessar a interface de indicadores do pet.	
	3. Exibir a interface que contém os indicadores do pet.
4. Pressionar o botão de editar ou criar novo registro de indicador	
	5. Verificar se é a edição de um registro, caso seja, buscar os dados do registro.
6. Preencher os campos de valor e data do indicador.	
	7. Verificar de qual indicador se trata (peso ou altura).
	8. Validar as informações e com base no resultado da validação salva ou não os dados.
	9.. Retornar o resultado da operação para o usuário.
<b>Validações</b>	Validar qual o tipo de indicador enviado, podendo ser indicador de peso ou altura, e se a data não é posterior à hoje.

Fonte: Elaborado pelo autor (2022).

Tabela 13 – Caso de uso “Gerar PDF do pet - UC10”.

<b>Nome do caso de uso</b>	Gerar PDF do pet - UC10
<b>Atores envolvidos</b>	Usuário, sistema, e banco de dados.
<b>Objetivo</b>	Descreve a ação de gerar um arquivo PDF – <i>Portable Document File</i> - com os dados do pet.
<b>Ações do ator</b>	<b>Ações do Sistema</b>
1. Acessar a interface que mostra os dados do pet	
2. Pressionar o botão para gerar o PDF do pet.	
	3. Realizar a consulta dos dados do pet.
	4. Preencher o layout do PDF desenvolvido com os dados do pet.
	5. Disponibilizar o PDF criado para o usuário.
<b>Validações</b>	Não há validações neste caso de uso.

Fonte: Elaborado pelo autor (2022).

### 3.3 BANCO DE DADOS

Segundo *Oracle* (2019), um Banco de Dados são várias informações aglomeradas em uma coleção estruturada, tipicamente armazenadas em um sistema de computador, posteriormente em um servidor. Geralmente, Banco de Dados são controlados por um Sistema Gerenciador de Banco de Dados (SGBD).

Dados são modelados em linhas e colunas em várias tabelas associadas a eles, podem ser acessados, modificados, adicionados e excluídos através de consultas (query), como *SELECT*, *INSERT*, *UPDATE*, *DELETE*, dentre outras milhares de consultas.

A grande maioria dos Bancos de Dados, utilizam um modelo estruturado de consulta, *Structured Query Language* (SQL), há alguns, no entanto, que utilizam a consulta por *Hibernate Query Language*, que é uma linguagem de consulta orientada a objetos, sendo sua sintaxe muito parecida com a linguagem SQL. (DEV MEDIA, 2022)

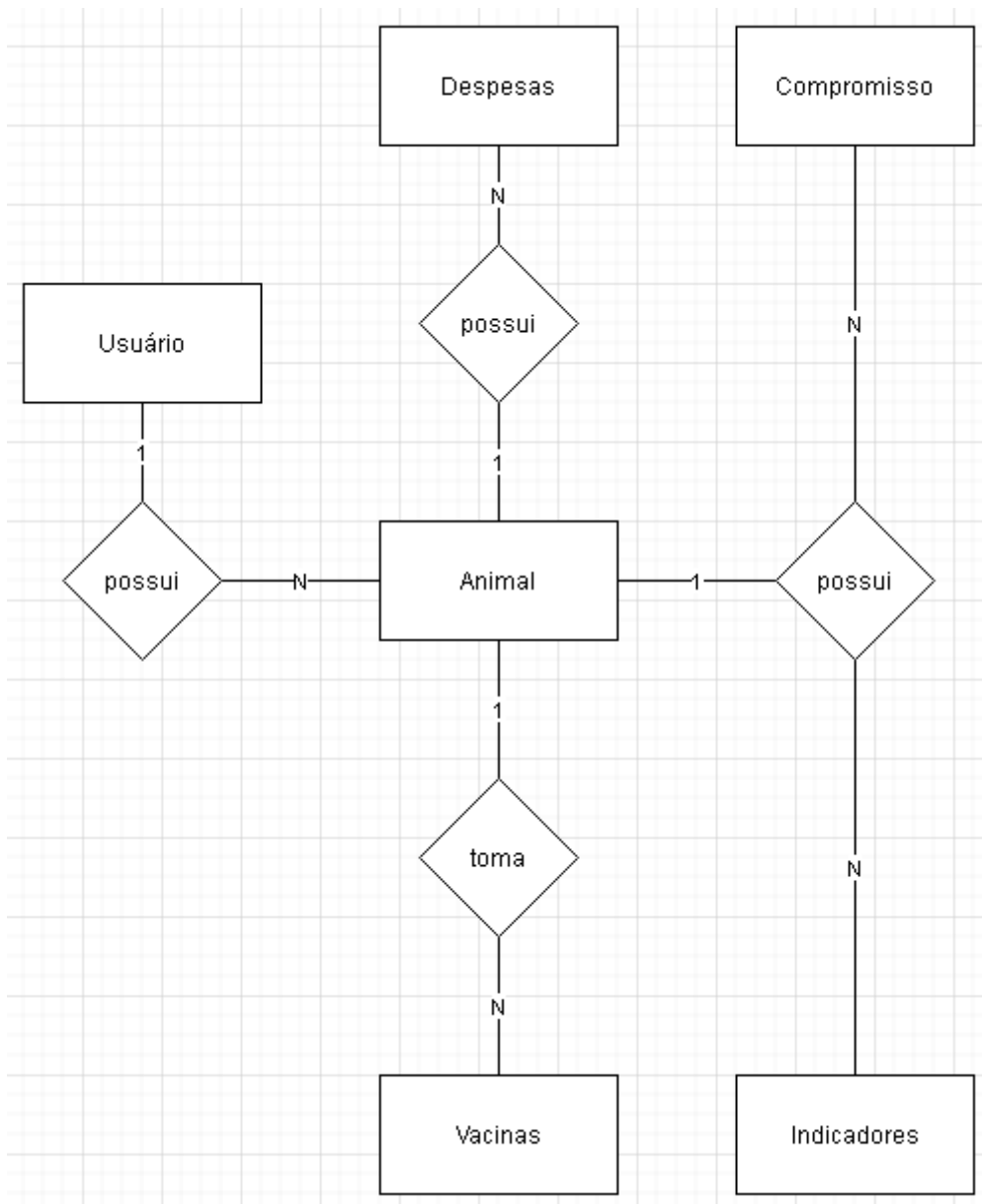
#### 3.3.1 DIAGRAMA ENTIDADE RELACIONAMENTO

Diagrama Entidade Relacionamento (DER) é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Ele é a principal representação do Modelo de Entidades e Relacionamentos. Sua maior aplicação é para visualizar o relacionamento entre tabelas de um banco de dados, no qual as relações são construídas através da associação de um ou mais atributos destas tabelas.

“O modelo entidade-relacionamento é uma ferramenta de modelagem usada para definir informações que serão necessárias a um modelo de dados baseado em entidades e relacionamentos” (LOBO, 2007).

A Figura 3 apresenta o DER do sistema proposto.

Figura 3 - Diagrama Entidade Relacionamento



Fonte: Elaborado pelo autor (2022).

### 3.3.3 DICIONÁRIO DE DADOS

O Dicionário de Dados (DD) consiste em uma lista organizada de todos os elementos de dados que são pertinentes para o sistema. As tabelas devem conter os seguintes campos:

**Entidade:** é o nome da entidade que foi definida no DER. A entidade é uma pessoa, objeto ou lugar que será considerada como objeto pelo qual temos interesse em guardar informações a seu respeito.

**Atributo:** Os atributos são as características da entidade cliente que desejamos guardar.

**Classe:** as classes podem ser: simples, composto, multivalorado e determinante. Simples indica um atributo normalmente. Composto indica que ele poderá ser dividido em outros atributos, como por exemplo, o endereço. Multivalorado é quando o valor do atributo poderá não ser único e determinante é um atributo que será usado como chave, como Cadastro de Pessoa Física (CPF), Código do cliente etc.

**Domínio:** podem ser numéricos, texto, data e booleano. Podemos chamar também de tipo do valor que o atributo irá receber. A definição desses tipos deve seguir um processo lógico, exemplo: nome é texto, salário é numérico, data de nascimento é data e assim por diante.

**Tamanho:** define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo. Geralmente o tamanho é definido apenas para atributos de domínio texto.

**Descrição:** é opcional e pode ser usado para descrever o que é aquele atributo ou dar informações adicionais que possam ser usadas futuramente pelo analista ou programador do sistema. (Oracle, 2022).

As Tabelas 14 a 19 apresentam o Dicionário de Dados de cada entidade.

Tabela 14 - Dicionário de dados da entidade Usuário

Entidade: Usuário				
Atributo	Classe	Domínio	Tamanho	Descrição
<i>Id</i>	Determinante	Texto	-	Identificador do registro
<i>Name</i>	Simple	Texto	-	Nome do usuário
<i>Email</i>	Simple	Texto	-	<i>Email</i> do usuário
<i>Photo</i>	Simple	Texto	-	Foto em Base64 do usuário
<i>LastSignIn</i>	Simple	Data	-	Data do último acesso do usuário
<i>CreationDate</i>	Simple	Data	-	Data de criação do registro
<i>LastUpdate</i>	Simple	Data	-	Data de última atualização do registro

Fonte: Elaborado pelo autor (2022).

Tabela 15 - Dicionário de dados da entidade Animal

Entidade: Animal				
Atributo	Classe	Domínio	Tamanho	Descrição
<i>Id</i>	Determinante	Texto	-	Identificador do histórico
<i>UserId</i>	Chave Estrangeira	Texto	-	Identificador do usuário dono do pet.
<i>Specie</i>	Simple	Texto	-	Espécie do usuário
<i>Weight</i>	Simple	Número	-	Peso do pet
<i>Photos</i>	Simple	Lista de Texto	-	Fotos em base64 do pet.
<i>Sex</i>	Simple	Enumerador	-	Sexo do pet: Macho ou Fêmea.
<i>Type</i>	Simple	Enumerador	-	Tipo do pet: cachorro, pássaro, ou outro
<i>IsFavorite</i>	Simple	Booleano	-	Indica se o pet foi favoritado no sistema pelo dono
<i>CreationDate</i>	Simple	Data	-	Data de criação do registro
<i>LastUpdate</i>	Simple	Data	-	Data de última atualização do registro



Fonte: Elaborado pelo autor (2022).

Tabela 16 - Dicionário de dados da entidade Vacina

Entidade: Vacina				
Atributo	Classe	Domínio	Tamanho	Descrição
<i>Id</i>	Determinante	Texto	-	Identificador da vacina
<i>PetId</i>	Chave Estrangeira	Texto	-	Identificador do pet correspondente.
<i>TakenDate</i>	Simple	Data	-	Data em que a vacina foi aplicada.
<i>RenovationDate</i>	Simple	Data	-	Data em que a próxima dose será aplicada
<i>Photo</i>	Simple	Texto	-	Foto do dependente.
<i>CreationDate</i>	Simple	Data	-	Data de criação do registro
<i>LastUpdate</i>	Simple	Data	-	Data de última atualização do registro

Fonte: Elaborado pelo autor (2022).

Tabela 17 - Dicionário de dados da entidade Despesa

Entidade: Despesa				
Atributo	Classe	Domínio	Tamanho	Descrição
<i>Id</i>	Determinante	Texto	-	Identificador da despesa
<i>Name</i>	Simple	Texto	-	Nome da despesa.
<i>Description</i>	Simple	Texto	-	Descrição da despesa.
<i>Value</i>	Simple	Número	-	Valor da despesa
<i>PetId</i>	Chave Estrangeira	Texto	-	Identificador de pet correspondente à despesa.
<i>Date</i>	Simple	Data	-	Data da despesa
<i>CreationDate</i>	Simple	Data	-	Data de criação do registro
<i>LastUpdate</i>	Simple	Data	-	Data de última atualização do registro

Fonte: Elaborado pelo autor (2022).

Tabela 18 - Dicionário de dados da entidade Compromisso

Entidade: Compromisso				
Atributo	Classe	Domínio	Tamanho	Descrição
<i>Id</i>	Determinante	Texto	-	Identificador do compromisso
<i>Name</i>	Simple	Texto	-	Nome do compromisso.
<i>Description</i>	Simple	Texto	-	Descrição do compromisso.
<i>Place</i>	Simple	Número	-	Local do compromisso
<i>PetId</i>	Chave Estrangeira	Texto		Identificador de pet correspondente ao compromisso.
<i>Date</i>	Simple	Data	-	Data do compromisso
<i>Time</i>	Simple	Texto	-	Horário do compromisso
<i>CreationDate</i>	Simple	Data	-	Data de criação do registro
<i>LastUpdate</i>	Simple	Data	-	Data de última atualização do registro

Fonte: Elaborado pelo autor (2022).

Tabela 19 - Dicionário de dados da entidade Indicador.

Entidade: Indicador				
Atributo	Classe	Domínio	Tamanho	Descrição
<i>Id</i>	Determinante	Texto	-	Identificador do compromisso
<i>PetId</i>	Chave Estrangeira	Texto	-	Identificador de pet correspondente ao indicador.
<i>Value</i>	Simple	Número	-	Valor do indicador
<i>Type</i>	Simple	Enumerador	-	Tipo de indicador: altura ou peso
<i>Date</i>	Simple	Data	-	Data em que o indicador foi inserido
<i>CreationDate</i>	Simple	Data	-	Data de criação do registro
<i>LastUpdate</i>	Simple	Data	-	Data de última atualização do registro

Fonte: Elaborado pelos autores (2021).

### 3.3.3 ESTRUTURA NOSQL

Para o desenvolvimento do aplicativo, utilizamos um Banco de Dados *Not only Structured Query Language* (NoSQL), sendo este o MongoDB. Por ser um Banco de Dados não relacional, este possui algumas vantagens que o relacional não possui, sendo elas: mais fácil de ser manipulado pelo usuário, mais flexível, possui alta escalabilidade e performance, são altamente funcionais, fornecendo ótimas *Application Programming Interface* (API's).

Há diversos tipos de Banco de Dados *NoSQL*, como chave-valor, gráfico, salvo em memória, pesquisar etc. Utilizamos o tipo Documento, para que os dados sejam visualizados no formato JSON (*JavaScript Object Notation*).

Sobre o modelo não relacional, pode-se afirmar que:

“Um banco de dados NoSQL usa uma variedade de modelos de dados para acessar e gerenciar os dados. Esses tipos de banco de dados são otimizados especificamente para aplicativos que exigem modelos de grande volume de dados, baixa latência e flexibilidade”. (AWS, 2020).

A seguir, serão apresentadas Figuras dos objetos guardados pela aplicação no banco de dados usando o software MongoDB Compass, que permite visualizar e gerenciar os dados da aplicação de forma direta.

A Figura 4 apresenta uma captura de interface no dia 06/04/2022 do objeto usuário, os campos atribuídos a ele e os devidos tipos de valores, sendo: *string*, *List<string>*, *date* e *boolean*.

Figura 4 - Representação do objeto Usuário

```
_id: ObjectId("6200579d4767e40794b279ad")
CreationDate: 2022-02-06T23:19:57.772+00:00
LastUpdate: 2022-04-06T13:29:15.228+00:00
UserId: null
Name: "Usuário de desenvolvimento"
Email: "desenv@gmail.com"
Password: "/B/+DxVfu/8T+mNLKN+BHthJMrQseZgvjX7PJKCI8Eo="
Photo: "iVBORw0KGgoAAAANSUHEUGAAATKAAAB1CAIAAAC6bgEFAAAAAXNSR0IArs4c6QAAAARnQU..."
Role: null
LastSignIn: 2022-04-06T03:00:00.000+00:00
```

Fonte: Elaborado pelo autor (2022).

A Figura 5 apresenta uma captura de interface no dia 06/04/2022 do registro de um animal, os campos atribuídos a ele e os devidos tipos de valores, sendo: *string*, *List<string>*, *date*, *boolean* e *int*. O objeto animal sempre vai possuir a chave estrangeira *UserId* que indica qual usuário é o dono do animal.

Figura 5 - Representação do objeto Animal

```

_id: ObjectId("6208184b0f41a869f792a217")
CreationDate: 2022-02-12T20:27:55.829+00:00
LastUpdate: 2022-04-05T20:53:23.252+00:00
UserId: "6200579d4767e40794b279ad"
Name: "teste"
Specie: "teste"
BirthDate: 2015-02-03T02:00:00.000+00:00
Weight: 15
~ Photos: Array
  0: "/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBxISEHUQEhAPEHUPEBUQEBAODwSPDxAQFR..."
Sex: 0
Type: 1
IsFavorite: true

```

Fonte: Elaborado pelo autor (2022).

A Figura 6 apresenta, por uma captura de interface tirada no dia 06/04/2022 o objeto Vacina. É importante lembrar que, uma vacina sempre pertence a um Pet, mas um Pet pode ter diversos registros de vacinas.

Figura 6 - Representação do objeto Vacina

```

_id: ObjectId("623e3f8e176da50efdfcbf32")
CreationDate: 2022-03-25T22:17:50.396+00:00
LastUpdate: 2022-03-25T22:17:50.396+00:00
UserId: "623b6f6fea30a1d482b928ff"
Name: "Raiva"
TakenDate: 2022-03-28T03:00:00.000+00:00
RenovationDate: 2023-03-28T03:00:00.000+00:00
PetId: "623babc0d3407ffa13c16f44"

```

Fonte: Elaborado pelo autor (2022).

A Figura 7 apresenta, por uma captura de interface tirada no dia 06/04/2022 do objeto Despesa. Como no objeto da vacina, uma despesa sempre pertence a um só Pet, mas um Pet pode ter diversos registros de despesas.

Figura 7 - Representação do objeto Despesa

```

_id: ObjectId("6209a7d2405b7c9b556c39b5")
CreationDate: 2022-03-28T22:11:42.137+00:00
LastUpdate: 2022-03-28T22:11:42.137+00:00
UserId: "6200579d4767e40794b279ad"
Name: "veterinário"
Description: "consulta de rotina"
Value: "50"
PetId: "6208184b0f41a869f792a217"
Date: 2022-02-13T03:00:00.000+00:00

```

Fonte: Elaborado pelo autor (2022).

A seguir na Figura 8, é apresentado o objeto Compromisso que também pertence sempre a um só pet.

Figura 8 - Representação do objeto Compromisso

```

_id: ObjectId("623cca03c55b1e875ed07cba")
CreationDate: 0001-01-01T00:00:00.000+00:00
LastUpdate: 2022-03-28T17:33:20.099+00:00
UserId: "623b6f6fea30a1d482b928ff"
Name: "castração"
Description: "Levar exames"
Place: "Americana"
Date: 2022-03-30T03:00:00.000+00:00
Time: "4:00 "
PetId: "623babc0d3407ffa13c16f44"

```

Por último, na Figura 9, é mostrado o objeto Indicador, que tem como finalidade registrar um valor que pode indicar tanto a altura ou peso do pet em uma determinada data. Assim, um indicador sempre pertencerá a um só pet, mas um pet pode ter vários indicadores.

Figura 9 - Representação do objeto Indicador

```

_id: ObjectId("624cd5890a509bc041f3571b")
CreationDate: 2022-04-05T23:49:29.219+00:00
LastUpdate: 2022-04-05T23:49:29.219+00:00
UserId: "6200579d4767e40794b279ad"
PetId: "6208184b0f41a869f792a217"
Value: 55
Date: 2022-04-05T03:00:00.000+00:00
Type: 0

```

Fonte: Elaborado pelo autor (2022).

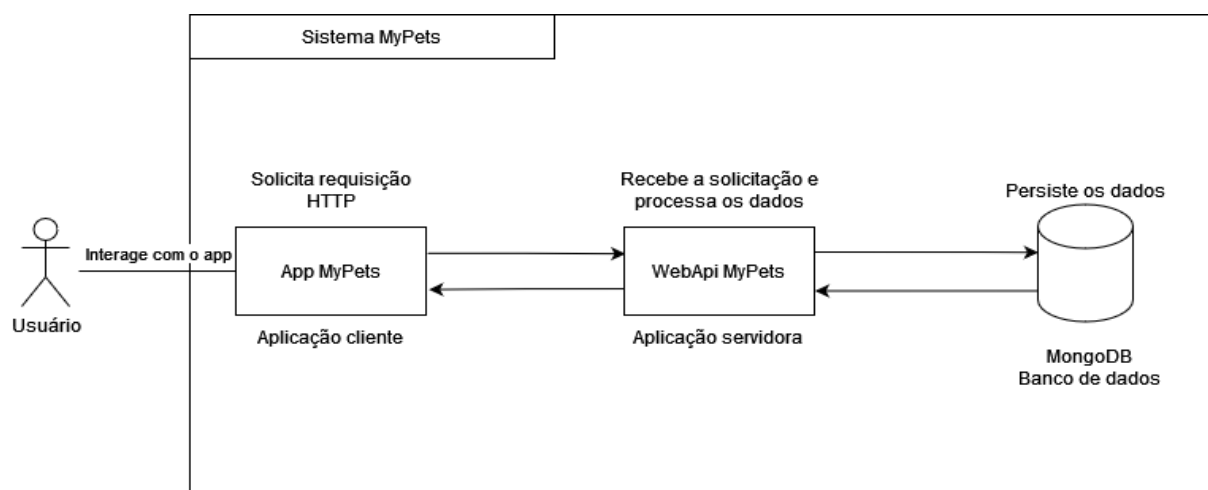
## 4. PROJETO

### 4.1 TECNOLOGIA EMPREGADA

O Sistema *MyPets* é composto por um aplicativo mobile desenvolvido com *Framework Flutter*, cuja função é ser o *front-end* do sistema, uma *WebApi* desenvolvida na tecnologia *Asp Net Core* e um Banco de Dados *MongoDB*, que juntos formam o *back-end* do sistema.

O aplicativo, tem como funcionalidade através da interface fazer a comunicação entre Cliente / Servidor com a *WebApi*, para solicitar por meio de requisições sob o protocolo HTTP o envio ou o retorno dos dados, conseqüentemente não possui nenhuma regra de negócio e não faz processamento ou manipulação dos dados, muito menos a comunicação com o Banco de Dados, conforme Figura 10, citado abaixo.

Figura 10 - Arquitetura do sistema



Fonte: Elaborado pelos autores (2022).

Todo o processamento e manipulação dos dados é feito por uma *WebApi RESTful* desenvolvida na tecnologia *Asp.Net Core*. A API recebe os dados dos clientes por meio das requisições direcionadas aos seus *endpoints*. Um *endpoint*

é um ponto que justamente conecta a aplicação cliente com a API e espera receber uma solicitação especificada.

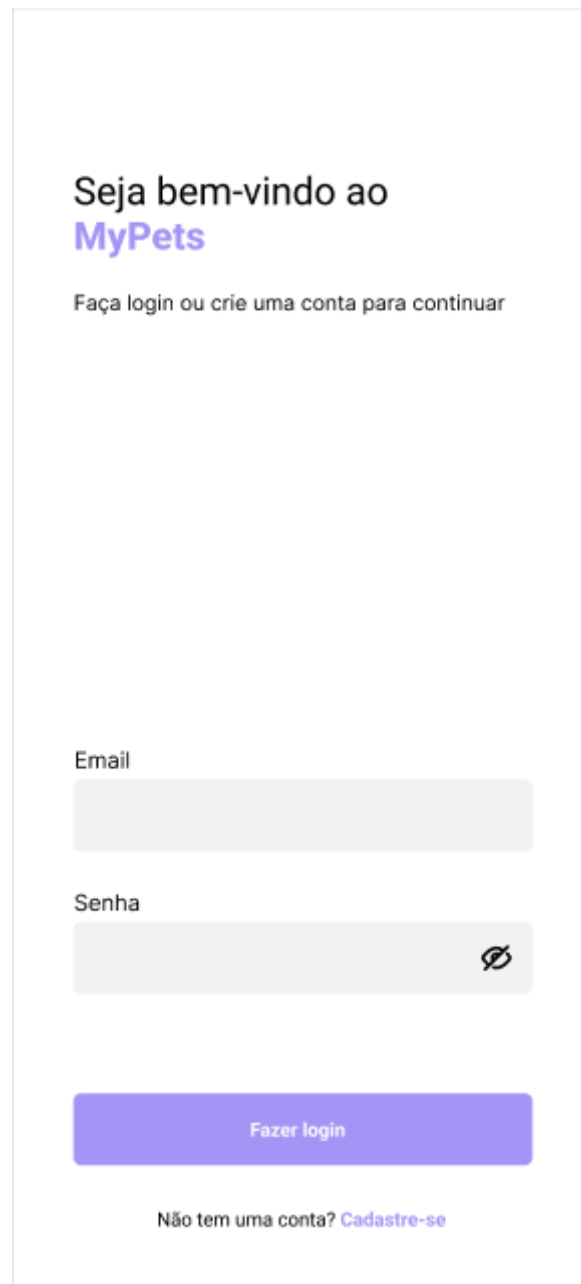
## 4.2 PRINCIPAIS INTERFACES DO PROJETO

Nesta seção, será apresentado a seguir as interfaces desenvolvidas no Projeto, que foram desenvolvidas respeitando os conceitos das heurísticas de Nielsen. As **Heurísticas de Nielsen** são 10 princípios de avaliação da usabilidade de interfaces de sites, criados pelo cientista de computação Jakob **Nielsen** em 1990. Estes princípios definem pontos importantes da composição de interfaces que devem ser considerados no momento da criação dos layouts.

### Interface de login

A primeira interface que o usuário visualiza ao acessar o aplicativo, é a interface de login, apresentada na Figura 11, que tem como principal finalidade autenticar o usuário no sistema, mas também redireciona o usuário para a criação de uma nova conta. Esta interface apresenta uma mensagem de saudação ao usuário, um formulário com os campos de usuário e senha e um botão de login que inicia o fluxo de login no sistema. Também há botões para redirecionar o usuário para o fluxo de recuperação de senha ou para o de criação de uma nova conta.

Figura 11 - Interface Login



The image shows a login interface for 'MyPets'. At the top, it says 'Seja bem-vindo ao MyPets' with 'MyPets' in a purple font. Below that, it says 'Faça login ou crie uma conta para continuar'. There are two input fields: 'Email' and 'Senha'. The 'Senha' field has a toggle icon for password visibility. Below the input fields is a purple button labeled 'Fazer login'. At the bottom, there is a link that says 'Não tem uma conta? Cadastre-se'.

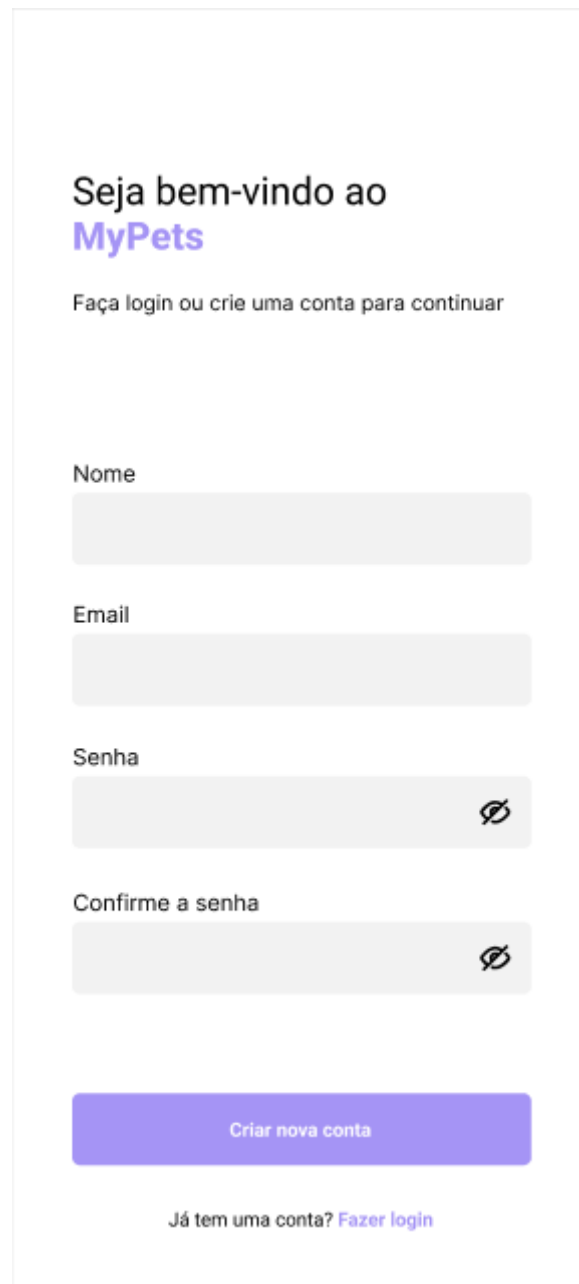
Fonte: Elaborado pelos autores (2022).

### Interface de novo usuário

Caso o usuário não tenha cadastro, ele deverá por meio da interface de login (Figura 11) se registrar pressionando o botão cadastre-se ao final da interface para que ele seja redirecionado para a interface de criação de novo usuário, ilustrada pela Figura 12 abaixo.



Figura 12 - Interface de criação do usuário



Seja bem-vindo ao  
**MyPets**

Faça login ou crie uma conta para continuar

Nome

Email

Senha

Confirme a senha

[Criar nova conta](#)

Já tem uma conta? [Fazer login](#)

Fonte: Elaborado pelos autores (2022).

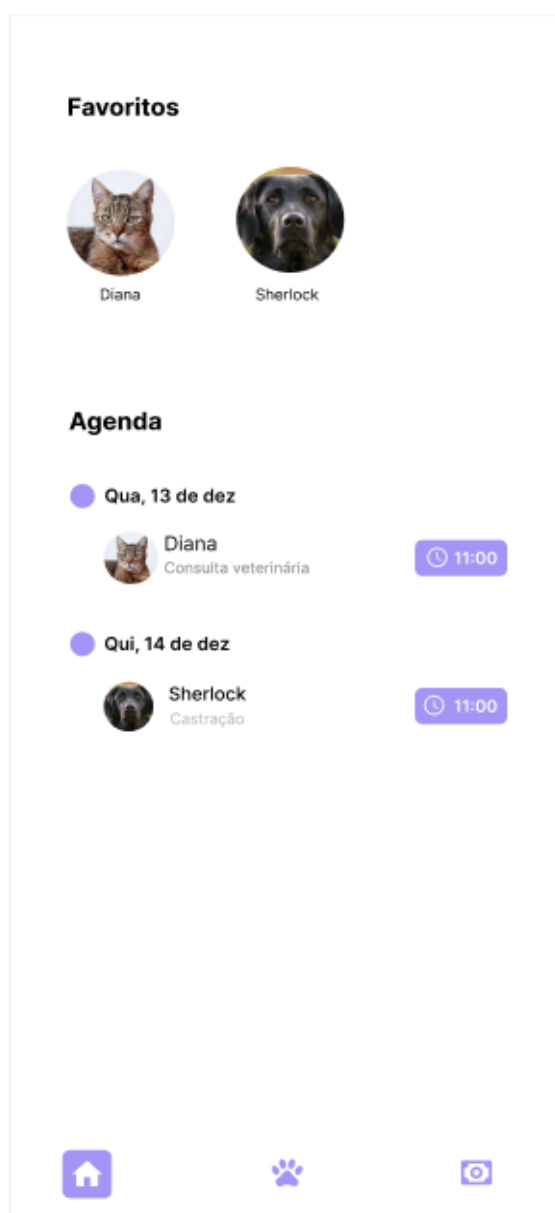
Esta interface apresenta um formulário com os dados necessários para o processo de criação do usuário, sendo eles: nome, e-mail e senha. Por segurança, a senha informada no campo é mascarada e o usuário deve confirmar a senha digitada no campo de confirmação abaixo. Ao pressionar o botão de criar conta, o sistema inicia o fluxo de registro e informa ao usuário o

resultado da operação por meio de uma mensagem. Em caso de sucesso, o usuário é redirecionado para a interface de login.

### Interface inicial (*Homepage*)

Após o processo de autenticação o usuário é redirecionado para a interface de *Homepage* que tem como finalidade centralizar as principais informações do usuário, como: os animais favoritos e os eventuais compromissos da semana dos mesmos, conforme a Figura 13.

Figura 13 - Interface Homepage



Fonte: Elaborado pelo autor (2022).

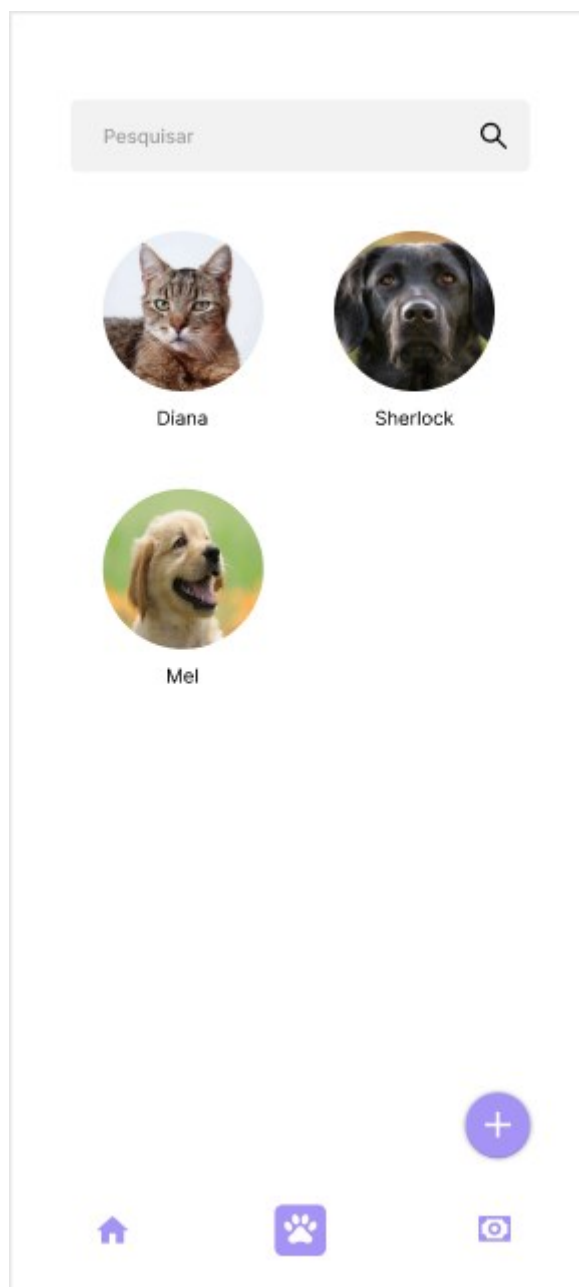
A seção de animais favoritos é uma lista horizontal composta de *cards* com a foto, nome e espécie do pet. Quando o *card* é pressionado pelo usuário, o sistema o redireciona para a interface de apresentação do pet.

A seção de agenda é composta por um componente que mostra os compromissos agrupados pelos dias da semana (por exemplo uma possível consulta de veterinário), com a foto do Pet, nome do compromisso e uma breve descrição.

Ao final da interface, há uma barra de navegação com a finalidade de indicar a interface atual em que o usuário se encontra e de redirecionar o usuário às outras interfaces do sistema, sendo estas: interface de listagem dos Pets e de gastos gerais.

## Interface de listagem de Pets

Figura 14 - Interface de listagem de Pets



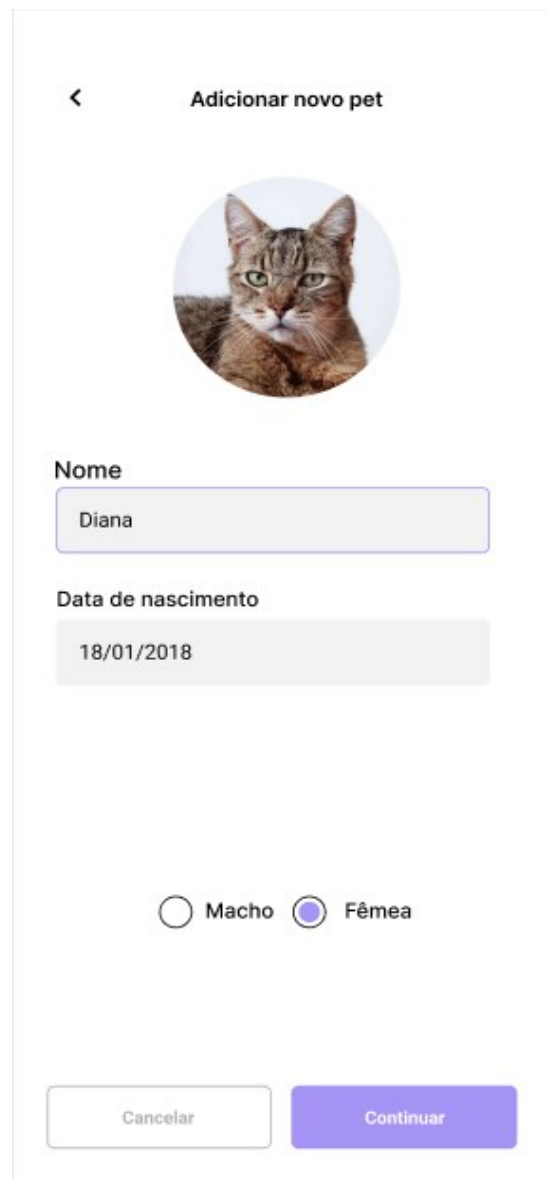
Fonte: Elaborado pelo autor (2022).

Conforme a Figura 14, esta interface reúne todos os Pets cadastrados no sistema por meio de um *card* com a foto e nome do Pet. É possível filtrar a busca inserindo o texto de pesquisa no campo de busca no topo da interface. Ao pressionar o *card* do respectivo pet, o usuário é redirecionado para a interface de apresentação do pet que será explanada em breve.


Na parte inferior direita da tela, há um botão para adicionar novos pets, quando este botão é pressionado pelo usuário, o sistema o redireciona para a interface de criação de pets, que será ilustrada na sequência. Ao final da interface está localizada a barra de navegação do sistema.

## Interface de criação/edição de pet

Figura 15 - Interface criação/edição de pet



< Adicionar novo pet



Nome  
Diana

Data de nascimento  
18/01/2018

Macho  Fêmea

Cancelar Continuar

Fonte: Elaborado pelo autor (2022).

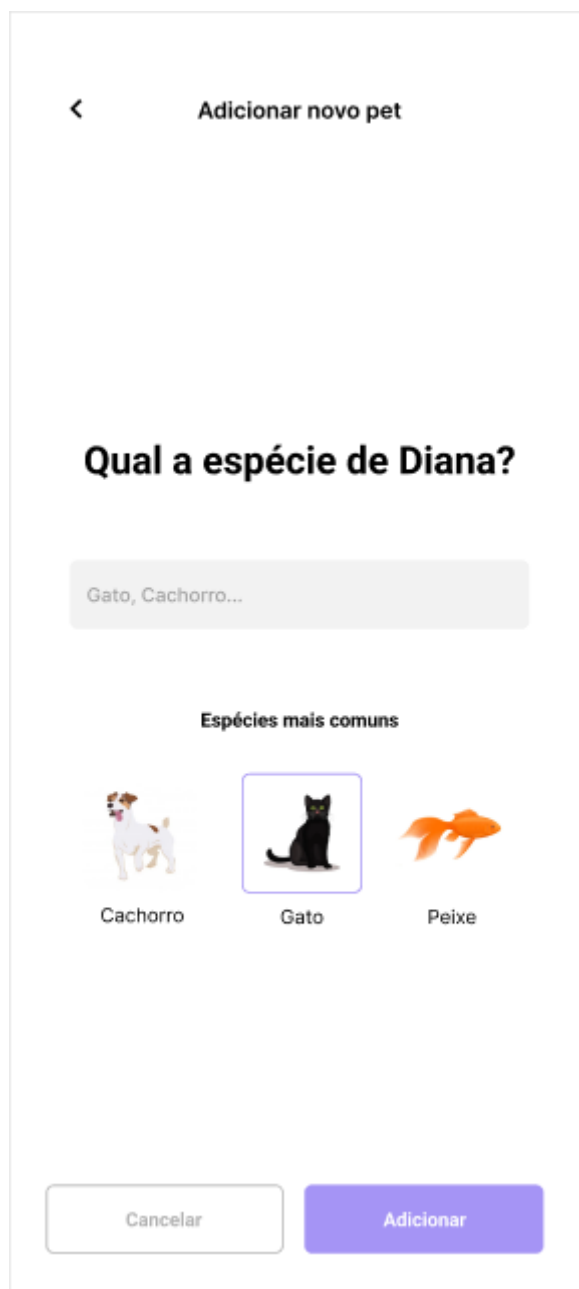
Conforme a Figura 15, esta interface é responsável por coletar os dados do usuário de um novo registro de um pet ou de atualizar o registro de um pet existente, informando qual operação está sendo feita (adição ou edição) por meio de uma mensagem no topo da interface, junto de um botão para retornar à interface anterior.

Abaixo é exibido um campo para o usuário inserir a foto do animal, enquanto a imagem não for selecionada será exibido um ícone de adição e uma mensagem para o usuário escolher a foto.

No campo de inserção da imagem, há um formulário com os campos de nome, data de nascimento e sexo do animal, este último sendo um botão de opção e os demais campos sendo de texto.

Na parte inferior da interface há um conjunto de dois botões, o primeiro da esquerda para a direita cancela a operação e o segundo continua o fluxo de criação/edição do pet exibindo a próxima interface, conforme a Figura a seguir.

Figura 16 - Interface de escolha de espécie



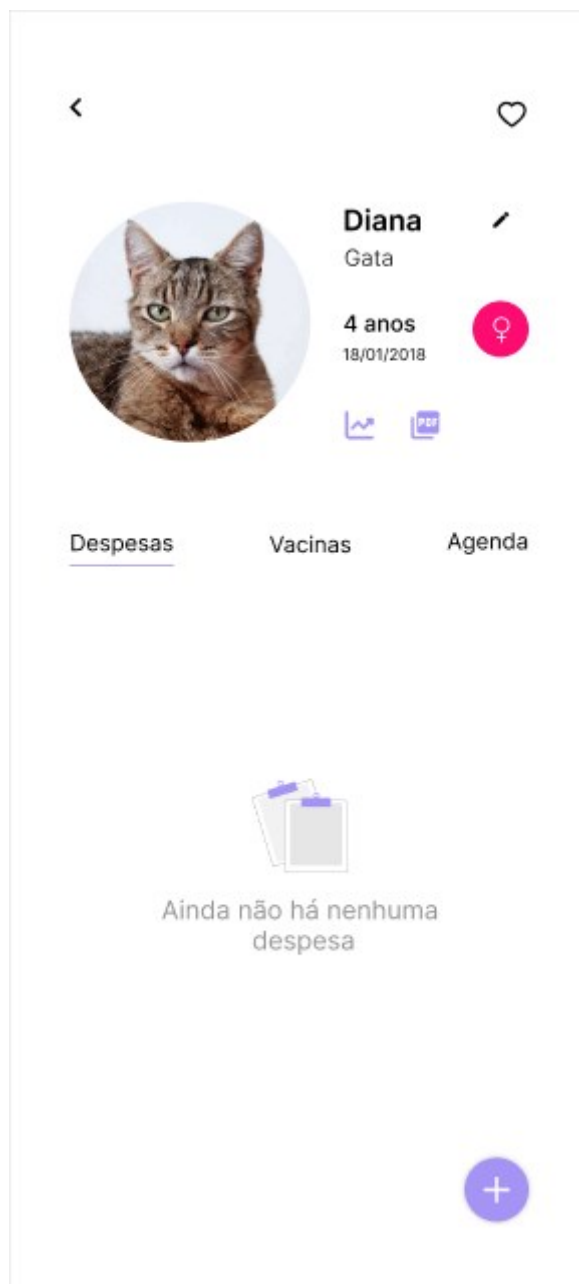
Fonte: Elaborado pelo autor (2022).

Na Figura 16, o usuário pode digitar a espécie do animal ou escolher entre três opções, que são: cachorro, gato ou peixe. Ao pressionar o botão de continuar, o aplicativo fará a solicitação para a API salvar os dados do pet. O usuário pode também cancelar toda a operação pressionando o botão cancelar ou voltar para a interface anterior pressionando o botão com o ícone de voltar junto do título da interface na parte superior.



## Interface de Apresentação do Pet

Figura 17 - Interface de apresentação do pet



Fonte: Elaborado pelo autor (2022).

A interface de apresentação do Pet, Figura 17, é carregada quando o usuário clica no *card* de um Pet que pode ser exibido tanto na interface *Homepage* quanto na interface de listagem dos Pets. Esta interface tem como finalidade todas as informações cadastradas do Pet, além de manter as despesas, vacinas e compromissos do Pet.

Em sua parte superior, é exibido um botão com o ícone de voltar, para o usuário navegar para a interface anterior e um botão com o ícone de coração que permite ao usuário favoritar o Pet. Os Pets favoritos são aqueles que são exibidos na interface inicial (*Homepage*).

Abaixo é exibido as informações primárias do Pet, como foto, nome, espécie, data de nascimento, idade e sexo. Ao lado do nome (em negrito) há um botão que leva o usuário para a interface de cadastramento/edição do pet para que essas informações possam ser atualizadas.

Há também abaixo da idade do Pet, um botão com um ícone de gráfico de linha e um botão com o ícone de arquivo PDF (*Portable Document File*). O primeiro leva o usuário para a interface de evolução da altura e peso do pet (próxima interface a ser mostrada) e o segundo gera um arquivo no formato PDF com os dados do animal.

Por último, há um sistema de três abas que exibem as despesas, vacinas e agendas, respectivamente, desta forma, a aba selecionada possui um traço na cor magenta para indicar ao usuário que está ativa no momento. Todas as abas exibem uma lista de *cards* que possuem o conteúdo de acordo com a respectiva aba.

É possível cadastrar uma nova despesa, vacina ou compromisso pressionando primeiramente a aba correspondente ao tipo de cadastro desejado e posteriormente no botão de adicionar ao final da interface, assim o sistema exibirá um modal que contém na parte superior na um botão para o usuário voltar caso desejado, o título do modal e um botão ao final da linha para salvar os dados do formulário correspondente ao tipo de cadastro. As Figuras abaixo mostram os modais de criação de nova despesa, vacina ou compromisso do pet.

Figura 18 - Modais de nova despesa, vacina e compromisso

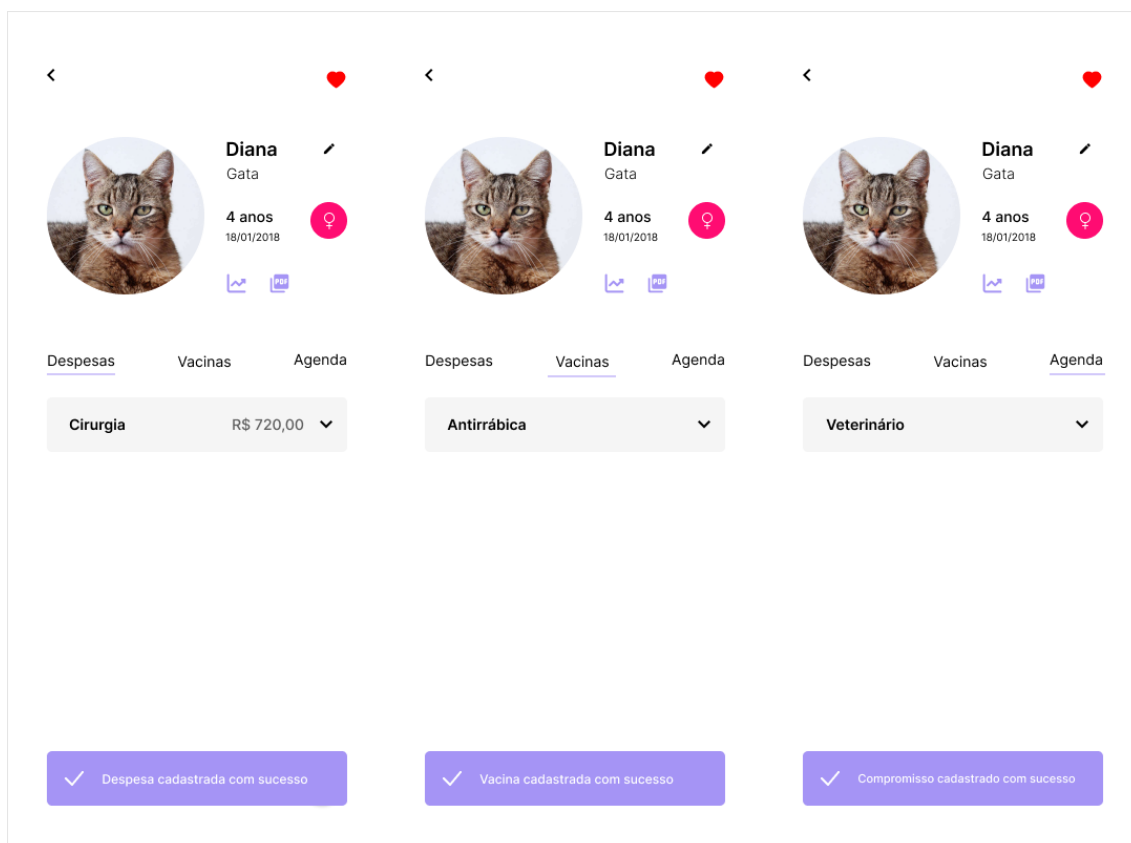
The image displays three overlapping modal forms for recording information for a cat named Diana. Each modal has a header with a back arrow, a heart icon, and the cat's name 'Diana' with a pencil icon for editing. Below the name is a circular profile picture of the cat, the text 'Gata', '4 anos', and '18/01/2018', and a pink female gender icon. At the bottom of each modal are three tabs: 'Despesas', 'Vacinas', and 'Agenda'. The forms are as follows:

- Cadastrar despesa:** Fields include 'Nome' (Cirurgia), 'Descrição' (Cirurgia ocular feita na clínica XPTO), 'Valor' (R\$ 720,00), and 'Data' (15/01/2021).
- Cadastrar vacina:** Fields include 'Nome' (Antirrábica), 'Data' (18/07/2018), and 'Data de renovação' (25/07/2018). There is a checkbox for 'Cadastrar data de renovação na agenda' which is unchecked.
- Cadastrar compromisso:** Fields include 'Nome' (Veterinário), 'Descrição' (Consulta veterinária para exames), 'Local' (Clínica WXYZ), 'Data' (20/02/2022), and 'Horário' (13h30).

Fonte: Elaborado pelo autor (2022).

Ao tocar em salvar o sistema iniciará o fluxo de cadastro da informação cadastrada e informará o resultado da operação por meio de uma mensagem na parte inferior da interface. A Figura 19 mostra as mensagens que retornam quando os cadastros foram realizados com sucesso.

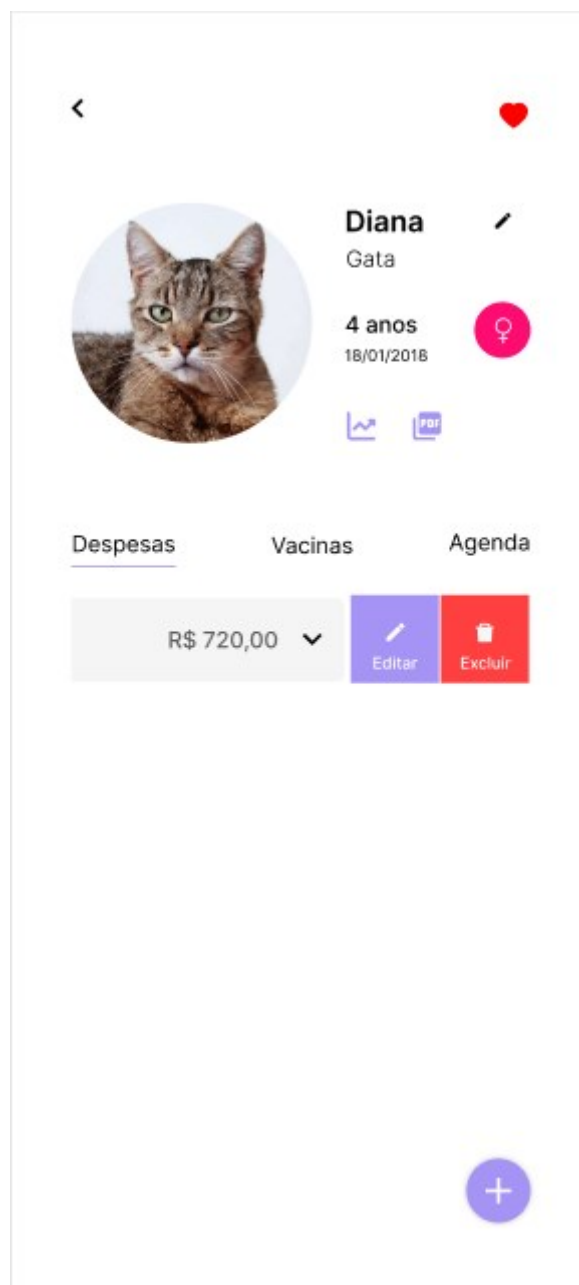
Figura 19 - Resultado cadastro de despesa, vacina e compromisso



Fonte: Elaborado pelo autor (2022).

Para realizar a edição ou a exclusão de um registro, o usuário deve deslizar o card para a direita (independente da aba selecionada) para que sejam exibidos os botões de edição e exclusão, e por último selecionar a operação correspondente. A Figura 20, ilustra abaixo, exibe a operação em um registro de despesa

Figura 20 - Editar ou excluir despesa



Fonte: Elaborado pelo autor (2022).

Ao pressionar o botão de editar o modal de manter despesa é aberto com as informações carregadas do registro, assim o usuário deve alterar as informações e pressionar o botão de salvar. Assim o sistema atualizará os dados da despesa e informará o resultado da operação.

Por fim, na opção de deletar, o sistema iniciará o fluxo de exclusão do registro e ao final informará o resultado da operação.

## Interface de evolução de peso e altura do pet

Esta interface tem a finalidade de mostrar a evolução do peso e altura de um determinado é acessada a partir da interface de detalhes (explicada anteriormente) ao pressionar no ícone de gráfico de linha logo abaixo da idade do Pet.

Figura 21 - Interface evolução de peso e altura do pet



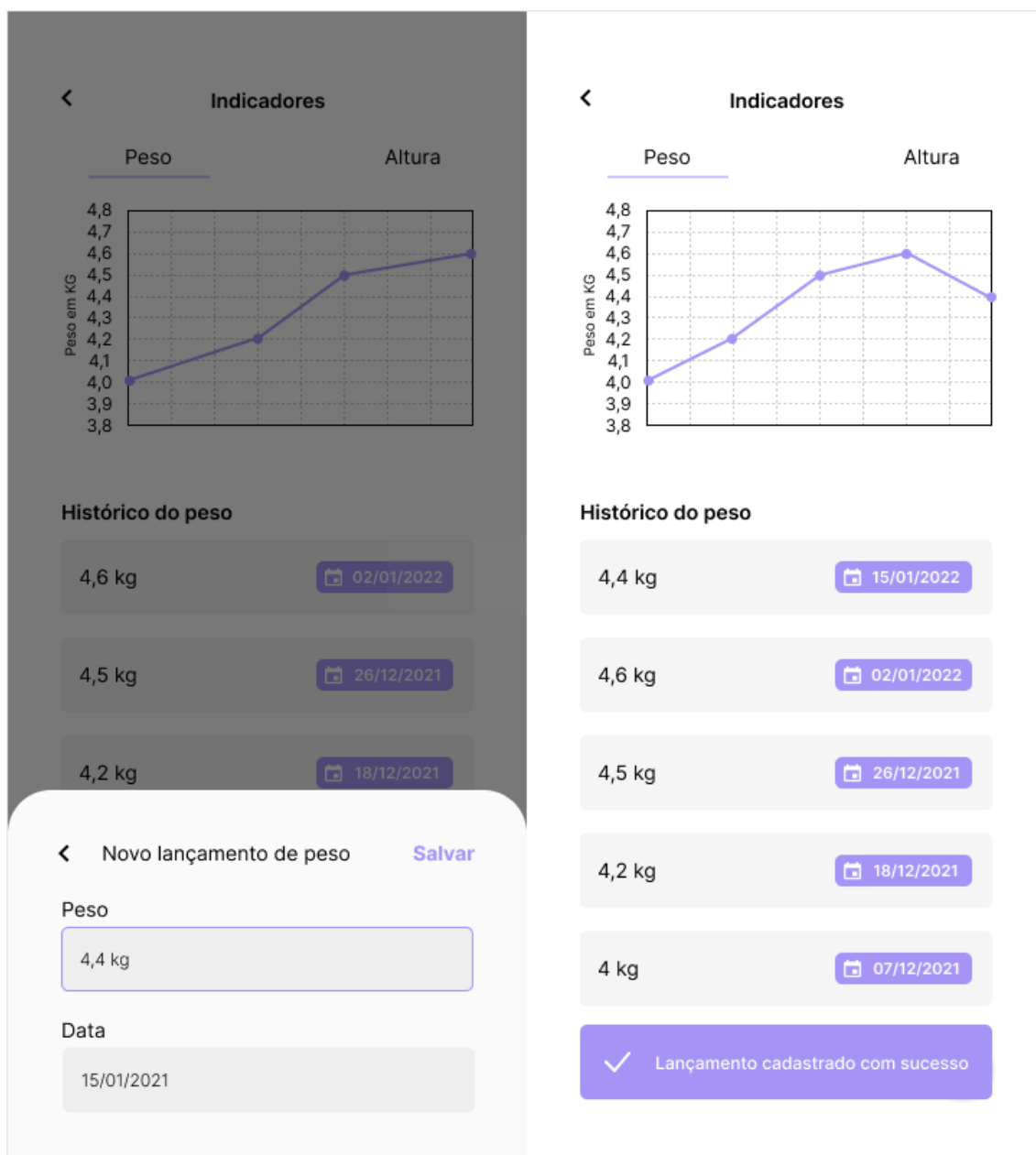
Fonte: Elaborado pelo autor (2022).

Conforme a Figura 21, são exibidas duas abas, sendo a primeira correspondente ao peso e a segunda à altura do Pet. Cada aba exibe um gráfico a partir dos registros cadastrados de peso ou altura do animal. Estes registros são exibidos logo após o gráfico e podem ser cadastrados selecionando a aba desejada e pressionando o botão de adicionar na parte inferior da interface.

Ao pressionar o botão de adicionar um registro, um modal para o cadastro da informação será exibido na parte inferior da interface com base na aba selecionada. Como nos outros fluxos de cadastro, o sistema exibirá uma mensagem com o resultado da operação e em caso de sucesso atualizará o gráfico com o novo registro inserido.

A Figura 22 abaixo mostra o cadastro de um novo registro de peso do pet e seu resultado.

Figura 22 - Cadastro de peso e resultado



Fonte: Elaborado pelo autor (2022).

Também é possível editar um registro deslizando o seu *card* para a esquerda, conforme a Figura 23 abaixo, assim serão exibidos os botões de edição e exclusão, assim como em outros fluxos o botão de editar exibirá o modal de cadastramento/edição do registro na parte inferior com os dados do registro carregados e o botão de deletar iniciará o fluxo de exclusão do registro.



Figura 23 - Editar ou excluir de registro de indicador



Fonte: Elaborado pelo autor (2022).

## Interface de Despesas Gerais

A Figura 24, exibe a interface com as informações sobre as despesas dos Pets do usuário. Na parte superior é exibida uma seção de resumo com o total gasto e a data da última despesa. Logo abaixo é mostrado um gráfico de setores, mostrando a participação de cada pet nas despesas no respectivo setor do pet

e valor em reais por meio das legendas, e terminando por uma lista dos registros das últimas três despesas, conforme ilustra a imagem abaixo, contido pela foto do pet, nome, data e valor da despesa.

Figura 24 - Interface de despesas

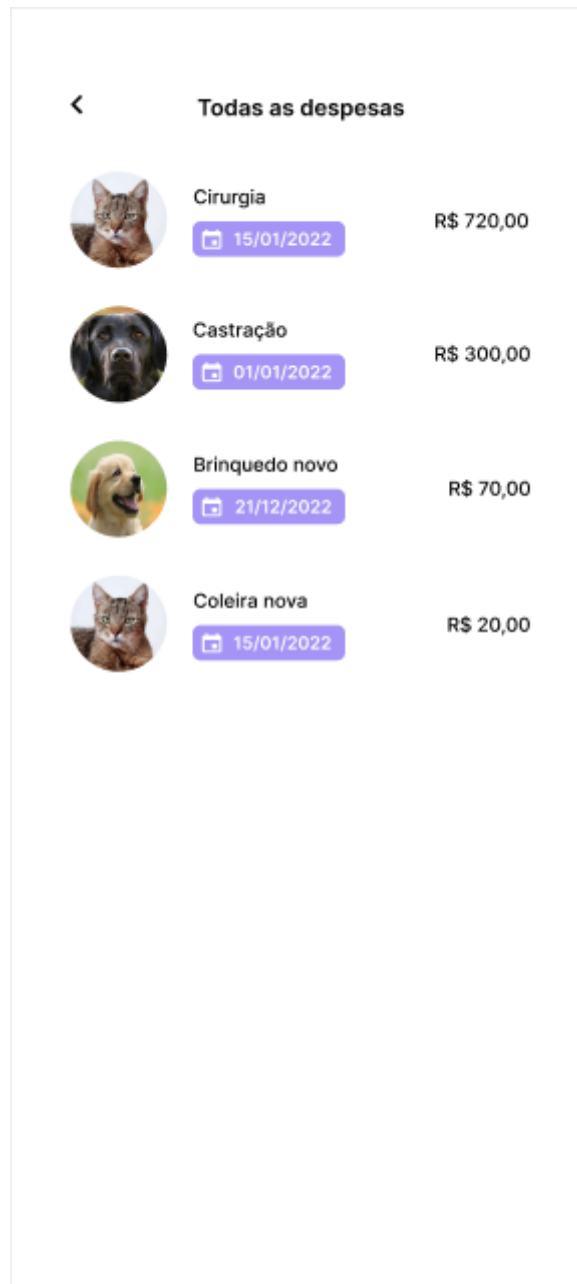


Fonte: Elaborado pelo autor (2022).

Ao final da interface também é exibido o componente de navegação, que permite ao usuário acessar as outras interfaces do sistema.

Na lista das últimas despesas há um botão que redireciona o usuário para uma interface que apresenta o registro de todas as despesas, ilustrada na Figura a seguir.

Figura 25 - Interface de todas as despesas



Fonte: Elaborado pelos autores (2022).

Nesta interface há um botão para voltar para a interface anterior e uma lista completa de todas as despesas do usuário, que também conta com a foto do pet, nome, data e valor da despesa.

### 4.3 RECURSOS E FERRAMENTAS

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do sistema:

- **Visual Studio Code:** Editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, *snippets* (código-fonte reutilizável) e refatoração de código.
- **Flutter:** Framework para o desenvolvimento de aplicações mobile, web e desktop criada pelo Google que utiliza a Linguagem *Dart*. Com o framework, um único *codebase* (código base) pode se tornar um aplicativo tanto para dispositivos Android quanto para dispositivos IOS.
- **Git:** Ferramenta para controle e versionamento de código que permite o desenvolvimento de projetos nos quais diversas pessoas podem contribuir simultaneamente, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas, registrando o histórico de edições e atualizações dos arquivos do projeto.
- **ASP.NET Core:** estrutura de código aberto de alto desempenho e plataforma cruzada para a criação de aplicativos modernos, habilitados para nuvem e conectados à Internet que permite a criação de: criar aplicativos Web e serviços, aplicativos relacionados à Internet das Coisas (*IoT*) e aplicações servidoras móveis.
- **Figma:** Ferramenta online de design gráfico para a prototipagem de interfaces para dispositivos mobile e computadores.
- **Diagrams.net:** Software online para o desenvolvimento de diagramas, fluxogramas, organogramas relacionados à modelagem de sistemas.
- **Mongo DB:** Software de banco de dados não relacional e orientado à documentos, em que cada documento pode ter a sua própria estrutura, diferente de um banco de dados relacional.

## 5. CONCLUSÃO

Ao longo do desenvolvimento deste Projeto, tivemos que desenvolver um sistema que atendesse as dificuldades dos usuários em centralizar as informações dos seus Pets em um só ambiente, adicionando a gestão de despesas, cuidados médicos e fotos para gerenciamento de Pets que superasse outros aplicativos já existentes no mercado. Com isso, durante algumas etapas do desenvolvimento nos deparamos com dificuldades de conhecimento sobre a tecnologia da Microsoft Asp.Net Core, visto que desenvolvemos uma API, que tem como objetivo concentrar as regras do negócio do sistema e se comunicar diretamente com o Banco de Dados. Assim, sem nenhum conhecimento prévio, superamos essa dificuldade por meio de pesquisas na documentação oferecida pela própria tecnologia e por meio da comunidade *web*, por meio de fóruns de tecnologia e videoaulas.

Desta forma, o sistema foi desenvolvido contemplando as funcionalidades propostas e esperamos implantar novas funcionalidades no sistema em um futuro e publicar para usuário interessados em adquirir por meios das lojas de aplicativos ou dos sistemas *Android* e *IOS*.

## 6. REFERÊNCIAS

VISUAL STUDIO CODE. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 26 de abr. 2022.

Postman. Disponível em: <<https://www.postman.com/>>. Acesso em: 16 de abr. 2022.

FLUTTER. Disponível em: <<https://flutter.dev/>>. Acesso em: 20 de fev. 2022.

GIT. Disponível em: <<https://git-scm.com/>>. Acesso em: 01 de jan. 2022.

ASP.NET CORE. Disponível em: <<https://docs.microsoft.com/pt-br/aspnet/core>>. Acesso em: 07 de jan. 2022

SOMMERVILLE, I. **Engenharia de Software**. 8ª Edição. Editora: Pearson Addison-Wesley. São Paulo, 2007.

PLACES API. Disponível em: <<https://developers.google.com/maps/documentation/>>. Acesso em: 11/03/2022.

SciElo. Disponível em: <https://www.scielo.br/j/rae/a/J3vT3wZMCNdyQdvtngQrQpH/?lang=pt>. Acesso em: 07/05/2022.

Gestão de Segurança Privada. Disponível em: <https://gestaodesegurancaprivada.com.br/sistema-de-informacao-o-que-e-conceitos/>. Acesso em: 07/05/2022

Material Design. Disponível em: <https://material.io/design>. Acesso em: 07/05/2022.

Dev Media - Introdução ao Hibernate Query Language (HQL). Disponível em: <https://www.devmedia.com.br/introducao-ao-hibernate-query-language-hql/29581>. Acesso em: 07/05/2022.

Oracle - The Data Dictionary. Disponível em: [https://docs.oracle.com/cd/B10501\\_01/server.920/a96524/c05dicti.htm](https://docs.oracle.com/cd/B10501_01/server.920/a96524/c05dicti.htm). Acesso em 20/05/2022.

Guedes, Gilleanes T.A. **UML 2 – Guia de Consulta Rápida** –2ª Edição. São Paulo: Novatec Editora Ltda, Novembro de 2005.