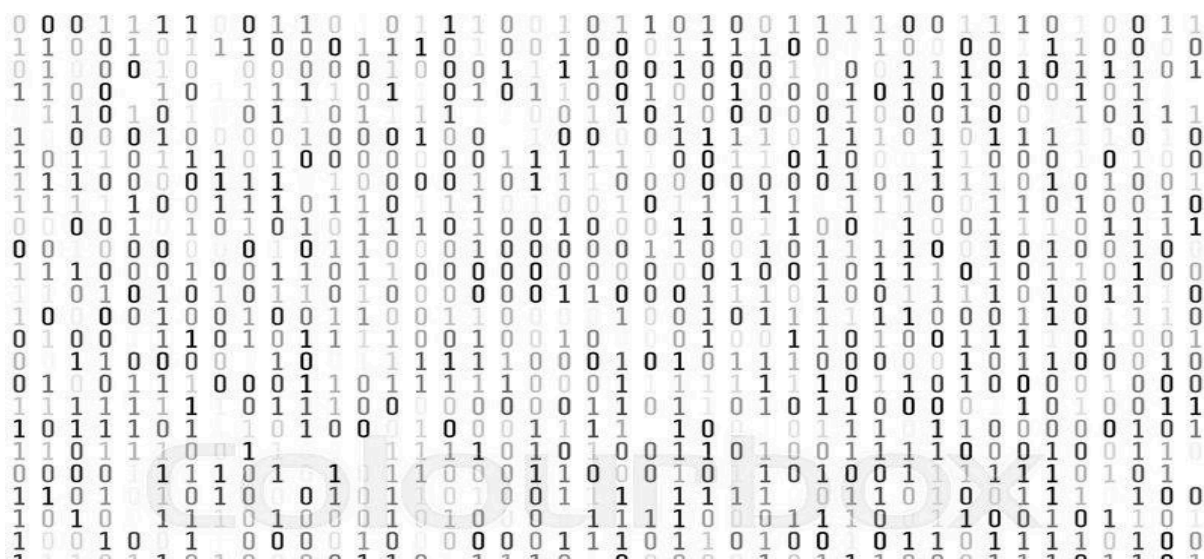


Relatório Final de Laboratório de Projeto Integrado

Projeto Vets4Pets

Aplicação móvel + Web Site + API



UNIVERSIDADE
FERNANDO PESSOA
WWW.UFP.PT

Alunos:

Miguel Costa 38573
Vasco Silva 38292
José Rodrigues 38099

Documento elaborado no âmbito da disciplina de Laboratório de Projeto Integrado, que tem como finalidade mostrar as características e funcionalidades de uma aplicação móvel/website de gestão de clínicas veterinárias. A mesma conta com a junção de um fórum de adoções para animais, assim como, um fórum de perguntas e respostas. Este relatório irá conter detalhadamente todas as funções e as diversas implementações que foram feitas ao longo da realização deste projeto.

Projeto com orientação de:

Professor Christophe Soares (csoares@ufp.edu.pt)

Professor Alessandro Moreira (afmoreira@ufp.edu.pt)

Docente da cadeira de LPI:

Professor Paulo Rurato (prurato@ufp.edu.pt)

Índice

1.	Introdução	7
1.1.	Contextualização	7
1.2.	Propósito	7
1.3.	Âmbito	7
1.4.	Definições, acrónimos e abreviaturas	9
1.5.	Referências	10
1.6.	Visão geral do presente documento	11
2.	Descrição	12
2.1.	Descrição API	12
2.1.1.	Formato dos pedidos	12
2.1.2.	JSON Web Token	13
2.1.3.	Visão geral da API	15
2.1.4.	Host	16
2.2.	Descrição da aplicação mobile para android	18
2.2.1.	Registo e Login	18
2.2.2.	HomePage e Navegação	20
2.2.2.1.	Abas	21
2.2.3.	Gestão dos animais de estimação	22
2.2.3.1.	Informações/Estatísticas de um animal de estimação	24
2.2.3.2.	Obtenção dos dados	25
2.2.4.	Clínicas e consultas	26
2.2.4.1.	Marcar uma consulta	27
2.2.5.	Perfil do utilizador	28
2.2.6.	Fóruns	29
2.2.6.1.	Fórum de Adoções	29
2.2.6.1.1.	Informações das adoções	29
2.2.6.2.	Fórum de Perguntas e Respostas	31
2.2.7.	Veterinários perto de si	32
2.3.	Descrição da aplicação web	33
2.3.1.	Registo e Login	33
2.3.2.	HomePage e Navegação	34
2.3.3.	Gestão dos animais de estimação	36

2.3.3.1.	Informações/Estatísticas de um animal de estimação	36
2.3.3.2.	Obtenção dos dados	38
2.3.4.	Clínicas e consultas	39
2.3.4.1.	Marcar uma consulta	40
2.3.5.	Perfil do utilizador	41
2.3.6.	Fóruns	42
2.3.6.1.	Fórum de Adoções	43
2.3.6.1.1.	Informações das adoções	43
2.3.6.2.	Fórum de Perguntas e Respostas	44
3.	Requisitos	45
3.1.	Requisitos Funcionais	45
3.1.1.	Aplicação Móvel	45
3.1.2.	Website	52
3.2.	Requisitos não funcionais	54
4.	Planeamento	55
4.1.	Tarefas	55
4.2.	Gráfico de Gantt	56
4.3.	GIT	56
4.4.	Tecnologias	57
5.	Modelização	58
5.1.	Modelos de Base de Dados	58
5.2.	Casos de Uso	63
5.2.1.	Administrador e Veterinário	63
5.2.2.	Utilizador	64
5.3.	Diagrama de Classes	65
6.	Demos e testes	67
6.1.	Aplicação móvel	67
6.1.1.	Dados teste	67
6.1.2.	Demos	67
7.	Conclusão	68
8.	LogBook	69

Índice de Figuras

1. Exemplo de uma resposta JSON	12
2. Exemplo de um resposta no Login de um Utilizador	13
3. Decodificação de um JWT	14
4. Exemplo de um pedido utilizando JWT no header	14
5. Visão geral de todos os EndPoints implementados no Swagger	15/16
6. Visão geral do container Docker	17
7. Visão detalhada de cada imagem do container	17
8. Api a correr no servidor Amazon AWS	17
9. Exemplo de Login (App)	18
10.Exemplo de Registo (App)	18
11. Validações (App)	19
12. Validação do email (App)	19
13. Validação do username (App)	19
14. Aplicação - HomePage (App)	20
15. Aplicação - NavBar (App)	20
16. Abas escondidas (App)	21
17. Animais do Utilizador (App)	22
18. Perfil do animal (App)	22
19. Adição de um animal (App)	23
20. Remoção de um animal (App)	23
21. Edição de um animal (App)	23
22. Informações gerais de um animal (App)	24
23. Detalhes da consulta de um animal (App)	24
24. Gráfico de peso do animal (App)	25
25. Listagem das clínicas (App)	26
26. Perfil da clínica (App)	26
27. Passos para marcar uma consulta (App)	27
28. Selecionar animal (App)	27
29. Selecionar veterinário (App)	27
30. Selecionar data/hora (App)	27
31. Detalhes da consulta (App)	28
32. Perfil do utilizador (App)	28
33. Calendário da aplicação (App)	28
34. Últimas adoções (App)	30
35. Informações da adoção 1 (App)	30
36. Informações da adoção 2 (App)	30
37. Estados de uma adoção (App)	30
38. Fórum (App)	31
39. Detalhes de uma pergunta (App)	31
40. Veterinários perto de si (App)	32
41. Formulário de Login (Website)	33

42. Formulário de Registo (Website)	33
43. Formulário de Alterar Password (Website)	33
44. HomePage para Utilizador do tipo "normal" (Website)	34
45. HomePage para Utilizador do tipo "vet" (Website)	35
46. HomePage para Utilizador do tipo "admin" (Website)	35
47. Lista de Animais (Website)	36
48. Perfil de um animal (Website)	36
49. Vacina (Website)	37
50. Detalhes da consulta do animal (Website)	37
51. Gráfico de peso do animal (Website)	37
52. Detalhes da Consulta (Website)	38
53. Formulário de adição de vacina (Website)	38
54. Formulário de adição de medicação (Website)	39
55. Lista de Clínicas (Website)	39
56. Perfil da Clínica (Website)	40
57. Passos para marcar uma consulta (Website)	40
58. Detalhes da Consulta no fim da marcação (Website)	41
59. Detalhes da Consulta do lado do utilizador (Website)	41
60. Todas as consultas marcadas do Utilizador (Website)	42
61. Página de Adoções (Website)	43
62. Página de Adoções (Website)	43
63. Foto Adoção Maior (Website)	43
64. Fórum de Perguntas e Respostas (Website)	44
65. Detalhes de Pergunta (Website)	44
66. Organização de Tarefas	56
67. Gráfico de Gantt	56
68. Modelo de Base de Dados	58
69. Casos de Uso para o Veterinário e para o Administrador	63
70. Casos de Uso para o utilizador	64
71. Diagrama de Classes relativo à aplicação	65
72. Código QR para instalar a aplicação	67

1. Introdução

1.1 Contextualização

Este projeto foi uma autoproposta pensada por todos os elementos do grupo, tendo mais tarde aprovação por parte dos respetivos orientadores do projeto. O mesmo surgiu após diversas conversações entre todos os constituintes do “Vets4Pets”.

Desta forma, a ideia do projeto resultou dos seguintes motivos:

- Criação de algo inovador;
- Potencializar os conhecimentos obtidos na licenciatura;
- Adição de mais conhecimento sobre o nosso mundo, nomeadamente o mundo da programação.

1.2 Propósito

O objetivo deste documento é fornecer uma descrição detalhada dos requisitos e funcionalidades implementados no projeto em si, descrição essa relativa à aplicação mobile/website.

1.3 Âmbito

“Vets4Pets” foi o nome escolhido para este projeto que, como o nome indica, significa Veterinários para Animais.

O principal objetivo desta “ideia” seria criar uma fácil interação entre utilizadores e os respetivos veterinários, tal como um acompanhamento dos dados dos animais de estimação desses mesmos utilizadores. Da mesma forma, é também possível encontrar um fórum de adoção de animais, tanto quanto um fórum de perguntas e respostas, onde um utilizador pode colocar as suas dúvidas sobre algo relacionado com o seu animal de estimação. Sempre que alguém coloca determinada dúvida pode obter respostas de outros utilizadores ou até mesmo de um veterinário registado na aplicação.

Para além disso, fornecer uma interface user-friendly (tanto na aplicação como no website) para a realização das diversas atividades como, por exemplo, a marcação de uma consulta para um dado animal de estimação ou a colocação de um animal para adoção. Permite também uma gestão fácil por parte do veterinário pois este ao estar registado na aplicação/website conseguirá fazer uma gestão das suas consultas, consultas essas feitas por vários utilizadores e os seus respetivos animais.

Para que um veterinário esteja registado na aplicação/website terá que estar associado primeiramente a uma clínica veterinária.

Este projeto suporta inúmeras clínicas veterinárias, onde cada uma possui o seu perfil, existindo também um número indefinido de Veterinários.

Por fim, esta associação entre veterinário e clínica é feita por parte do Administrador, ele que possui o seu painel de controle onde pode fazer todo o tipo de operações, como por exemplo criar uma clínica e associar vários veterinários à mesma.

Em suma, existem 3 tipos de utilizadores:

- normal - utilizar normal (UI implementados na aplicação/website).
- vet - conta de um utilizador veterinário (UI implementado apenas no site).
- admin - conta que faz a gestão de clínicas (UI implementado apenas no site).

Mais detalhes sobre cada um dos tipos de utilizadores irão ser mostrados ao longo do documento.

1.4 Definições, acrônimos e abreviaturas

Utilizador	Indivíduo que interage com as aplicações implementadas.
Veterinário	Tipo de utilizador (por norma um veterinário registado que está associado a uma clínica).
Administrador	Tipo de utilizador cuja principal função é fazer uma gestão das várias clínicas.
Aplicação	Conhecida também por app, é um software desenvolvido para ser instalado num dispositivo móvel.
Website	Referência a uma página ou a um agrupamento de páginas relacionadas entre si, acessíveis na internet através de um determinado endereço.
API	Conjunto de definições e protocolos usado no desenvolvimento e na integração de software de aplicações. API é um acrónimo em inglês que significa interface de programação de aplicações.
UI	Cuja abreviatura significa “User Interface” faz referência ao processo que os designers usam para construir interfaces em software ou dispositivos computadorizados, com foco na aparência ou no estilo.
Host	Máquina/Computador conectado a uma rede que oferece informações, recursos, serviços e aplicações aos usuários ou outros nós na rede.
Navbar	Barra lateral de navegação, tanto na aplicação móvel como na aplicação web
User-Friendly	Fácil de usar, aprender ou perceber
Timestamp	Instância de tempo

1.5 Referências

ESA - Board for Software Standardisation and Control. (2021). Guide to applying the ESA software engineering standards to small software projects. Obtido de ESA: <ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/Bssc962.pdf>

Soares, C. (2020). Apontamentos da Disciplina de Laboratório de Programação. Porto, Portugal: Universidade Fernando Pessoa.

Pereira, I. (2020). Apontamentos da Disciplina de Engenharia de Software. Porto, Portugal: Universidade Fernando Pessoa.

Gouveia, F. (2020). Apontamentos da Disciplina de Base de Dados. Porto, Portugal: Universidade Fernando Pessoa.

Foram também utilizados como referências todos os documentos disponibilizados pelo professor Paulo Rurato, que se encontram na plataforma.

1.6 Visão geral do presente documento

Nas próximas páginas são descritas detalhadamente todo o tipo de funcionalidade e implementações sobre este projeto, detalhes esses que fazem referência as 3 partes principais do projeto em si, sendo elas:

- API
- Aplicação móvel (Android)
- Website

Esta descrição minuciosa irá se encontrar no capítulo 2, estando numericamente dividida (2.1 - API | 2.2 - Aplicação movel android | 2.3 - Website). Em seguida, no capítulo 3 serão expostos todos os Requisitos Funcionais e Requisitos Não-Funcionais, estando todos por uma ordem lógica identificados por um ID único. No capítulo 4 estará presente a fase de planejamento do projeto, nomeadamente as tarefas, gráfico de Gantt bem como todas as tecnologias e ferramentas utilizadas. No capítulo 5 estão presentes alguns tipos de diagramas, nomeadamente diagrama de Casos de Uso, de classes e base de dados. No capítulo 6 irão ser mostrados alguns dados teste, assim como um QR code para fazer download da aplicação, para que os respetivos avaliadores deste documento, caso necessitem, possam testar o projeto. Dados estes serão dados para fazer login na aplicação/website. Irão também ser colocados links para vídeos do youtube, vídeos esses que mostram todo o tipo de funcionalidade tanto no website como na aplicação. Por fim, o capítulo 7, irá fazer referência à conclusão.

2. Descrição

Como já foi referido este capítulo foca numa descrição detalhada de todo o tipo de funcionalidade do projeto, dividido em 3 partes.

2.1 Descrição API

API é utilizada para fazer a gestão de todo tipo de informação, através de pedidos, pedidos esses que são sincronizados entre a aplicação e o website. Esta API foi feita com recurso à linguagem GO, linguagem criada pela Google.

Por se tratar de uma API não possui qualquer tipo de interface de usuário, sendo apenas um conjunto de ENDPOINTS, cada um estando relacionado a um determinado pedido, por exemplo:

- Realizar um registo/Autenticação.
- Realizar uma operação de adição/remoção/edição (um animal por exemplo).
- Realizar um pedido para obter qualquer tipo de informação.

A informação encontra-se armazenada numa base de dados, escrita em PostgreSQL, que se encontra dividida em diversas tabelas, tabelas essas que irão ser explicadas no decorrer deste documento (capítulo 5).

2.1.1 Formato dos pedidos

Todos os pedidos foram realizados em formato **JSON**, sigla para um acrônimo de “*JavaScript Object Notation*”, que é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida entre sistemas.

```
{
  "data": [
    {
      "ID": 6,
      "CreatedAt": "2021-06-06T16:18:20.473516Z",
      "UpdatedAt": "2021-06-06T16:18:20.473516Z",
      "DeletedAt": null,
      "showedUp": true,
      "AnimalID": 1,
      "date": "2021-09-23T14:00:00Z",
      "VetID": 5,
      "canceled": false
    }
  ],
  "status": 200
}
```

Figura 1. Exemplo de uma resposta JSON

Neste exemplo encontra um pedido de uma consulta para um determinado animal, neste caso o animal com ID=1. Pela visualização da figura é possível observar os diferentes campos, cada um deles representando um tipo de informação.

ID: Identificador único daquela consulta.

CreatedAt: Data e hora exata da criação daquela consulta

UpdateAt: Data e hora exata da edição/alteração de algum tipo de informação daquela consulta.

DeleteAt: No caso de uma remoção do conjunto de informação é detalhado a data e hora, neste caso como não houve remoção o campo encontra-se a null.

showedUp: Variável booleana caso o animal da consulta tenha aparecido ou não.

date: Data da consulta.

VetID: ID do veterinário.

canceled: Variável booleana caso a consulta seja cancelada.

2.1.2 JSON Web Token

Cerca de 90% dos pedidos à API são realizados com **JWT**, abreviatura para “*JSON Web Token*”, que um padrão da Internet para a criação de dados com assinatura opcional e/ou criptografia cujo payload contém o JSON que afirma algum número de declarações. O token é obtido na resposta JSON sempre que um utilizador se autentica na aplicação/website. Sempre que um dado utilizador realizar um pedido, o token será enviado na header desse pedido.

```
...  
"status": 200,  
  "token":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im1pZ3VlbGNv  
c3RhMzAiLCJlbWFPbCI6Im1pZ3VlbnC5jb3N0YTMwMzAzQGdtYWlsLmNvbSIsIlVzZX  
JJRCIMSwibmFtZSI6Ik1pZ3VlbnCBGZXJyZWlyYSBDd3N0YSIsImNvdnRhY3QiOiI5  
Mzg2MDc3OTYiLCJnZW5kZXIIoMZhbnHN1LCJleHAiOjE2MjQxMjUwMjh9.7tLEHzpVQ  
tM5GkZLJQNlxbDzJNEq7PEV12ory6C5R5s",  
  "userType": "normal"  
}
```

Figura 2. Exemplo de um resposta no Login de um Utilizador

No token estão presentes diversos tipos de informação como mostra a próxima figura:

Encoded

PASTE A TOKEN HERE

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im1pZ3VlbGNvc3RhMzAiLCJlbWFpbCI6Im1pZ3VlbC5jb3N0YTMwMzAzQGdtYWlsLmNvbSIsIlVzZXJJRCI6MSwiImFtZSI6ImNvbnRhY3QiOjE1Mzg2MDc3OTYiLCJnZW5kZXIiOmZhbnHNlLCJleHAiOjE2MjQxMjUwMjUwLjU0LEhzcVQtM5GkZLJQNlbnDzJNEq7PEV12ory6C5R5s

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

{
 "alg": "HS256",
 "typ": "JWT"
}

PAYLOAD: DATA

{
 "username": "miguelcosta30",
 "email": "miguel.costa3030@gmail.com",
 "UserID": 1,
 "name": "Miguel Ferreira Costa",
 "contact": "938607796",
 "gender": false,
 "exp": 1624125028
}

VERIFY SIGNATURE

HMACSHA256(
 base64UrlEncode(header) + "." +
 base64UrlEncode(payload),

) ☐ secret base64 encoded

Figura 3. Decodificação de um JWT

GET	/appointmentOfuser/{id}	Exibe as consultas de um User
Exibe as consultas de um User		
Parameters		
Name	Description	
Authorization * required	Token	
string (header)	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ	
id * required	ID do User	
integer (path)	1	

Figura 4. Exemplo de um pedido utilizando JWT no header

Nota: Apenas os pedidos de autenticação, registo e alterar a palavra passe não utilizam JWT.

Numa primeira fase antes das respetivas interfaces de utilizador estarem implementadas os pedidos eram feitos via Swagger (como mostra na figura 4) ou PostMan. Depois de implementadas os respetivos UI todos os pedidos eram feitos de forma dinâmica.

2.1.3 Visão geral da API

No total foram criados 48 endpoints cada um com a sua tarefa específica.

POST	/auth/login	Realizar autenticação	
PUT	/auth/refresh_token	Atualiza token de autenticação	
POST	/auth/register	Realizar registro	
POST	/auth/forgot_password	Forgot password	
POST	/auth/validate_reset_token"	JWT Validate	
POST	/auth/reset_password"	JWT Validate	
POST	/animal	Create Animal	🔒
DELETE	/animal/{id}	Exclui um animal pelo ID	🔒
PUT	/animal/{id}	Atualiza um Animal	🔒
GET	/animal/{id}	Recupera um Animal pelo id	🔒
GET	/animal/{id}/{userID}	Recupera um Animal pelo id	🔒
GET	/user/{id}	Recupera um Usuario pelo id	🔒
PUT	/user/{id}	Atualiza um Usuario	🔒
GET	/user	Recupera os Users	🔒
GET	/userNormal	Recupera os Users Normal	🔒
GET	/userVet	Recupera os Users Normal	🔒
GET	/userAnimals/{id}	Recupera os Animais pelo id do Usuario	🔒
POST	/prescription	Create Prescription	🔒
GET	/prescription/{id}	Recupera uma prescription por id	🔒
DELETE	/prescription/{id}	Exclui uma prescription pelo ID	🔒
GET	/prescriptionUser/{id}	Recupera uma prescription por id	🔒
POST	/question	Create Question	🔒
GET	/question/{id}	Recupera uma pergunta por id	🔒
DELETE	/question/{id}	Exclui uma pergunta pelo ID	🔒
PUT	/question/{id}	Edita Pergunta	🔒
GET	/questions	Exibe as perguntas ordenadas por data	🔒
POST	/answer	Adds a answer	🔒
DELETE	/answer/{id}	Exclui uma resposta pelo ID	🔒

DELETE	/answer/{id}	Exclui uma resposta pelo ID	🔒
GET	/answer/{id}	Exibe as repostas de uma pergunta	🔒
GET	/answersByUser/{id}	Exibe as repostas de um user	🔒
GET	/questionsByUser/{id}	Exibe as perguntas de um user	🔒
POST	/vaccine	Adds a vaccine	🔒
DELETE	/vaccine/{id}	Exclui uma vacina pelo ID	🔒
GET	/vaccine/{id}	Exibe as vacinas de um animal	🔒
POST	/clinic	Adds a clinic	🔒
GET	/clinic	Exibe as clínicas ordenadas por id	🔒
DELETE	/clinic/{id}	Exclui uma Clínica pelo ID	🔒
PUT	/clinic/{id}/{UserID}	Adiciona um veterinario a uma clínica	🔒
PUT	/clinicRem/{UserID}	Remove um veterinario a uma clínica	🔒
GET	/vetsClinic/{id}	Exibe os vets de uma clínica	🔒
POST	/adoption	Adds a adoption	🔒
GET	/adoption/{id}	Exibe as adoções criadas por um user	🔒
PUT	/adoption/{id}	Atualiza uma Adoption	🔒
DELETE	/adoption/{id}	Exclui uma adoção pelo ID	🔒
GET	/adoptionByTime	Exibe as adoções ordenadas por data	🔒
POST	/appointment	Create Appointment	🔒
PUT	/appointment/{id}	Atualiza um Appointment	🔒
GET	/appointment/vet/{id}	Exibe os appointments de um vet	🔒
GET	/appointmentOfuser/{id}	Exibe as consultas de um User	🔒

Figura 5. Visão geral de todos os EndPoints implementados no Swagger

É possível, através da observação da figura identificar os pedidos que utilizam JWT (os que possuem um cadeado) e os que não utilizam.

2.1.4 Host

Como já referido, primeiramente a API ficou localizada na maquina da cada membro do grupo, com recurso a **Docker**, sendo assim cada um possuía informações diferentes pois possuía diferentes bases de dados.

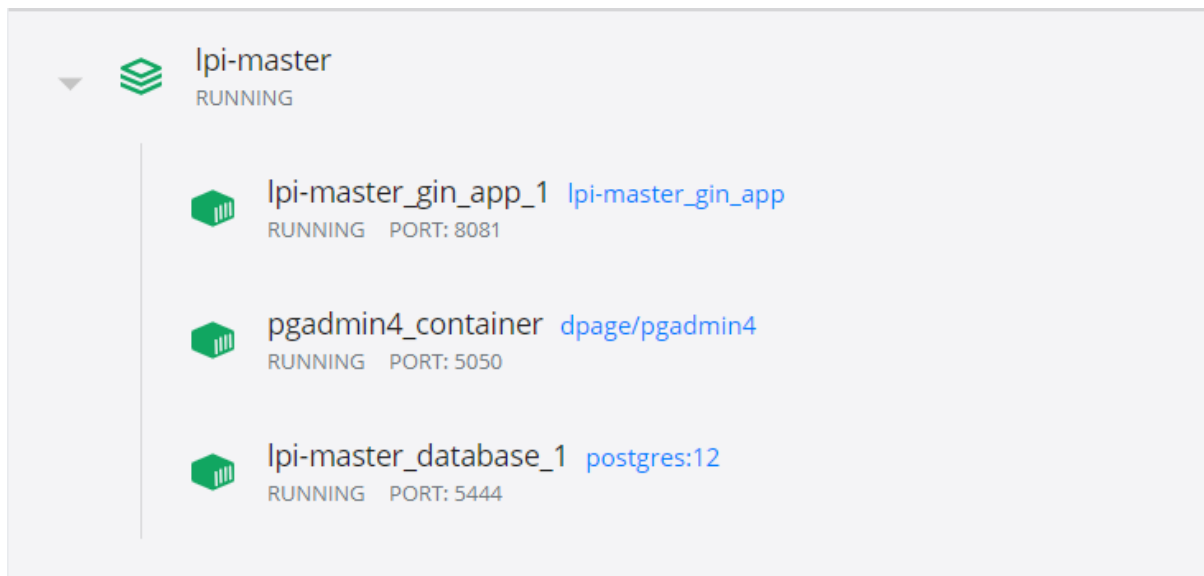


Figura 6. Visão geral do container Docker.

		TAG	IMAGE ID	CREATED	SIZE
lpi-master_gin_app	IN USE	latest	a78617db8225	3 months ago	1.47 GB
postgres	IN USE	12	925b50fe0dbd	3 months ago	314.09 MB
dpage/pgadmin4	IN USE	latest	636dbafc286f	3 months ago	243.28 MB

Figura 7. Visão detalhada de cada imagem do container.

lpi-master_gin_app: Imagem onde está a correr a API.

postgres: Imagem onde se encontra a base de dados e toda a informação.

dpage/pgadmin4: Imagem auxilia para visualização e edição das tabelas da base de dados.

Numa segunda fase migrou-se para um servidor AWS EC2 para que torna-se mais fácil a gestão de informação e a "conexão" entre a aplicação móvel e o site fosse realizada.

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	-	i-04b7765fa18d2de1a	Running	t2.micro	2/2 checks passed	No alarms	eu-west-3b

Figura 8. Api a correr no servidor Amazon AWS.

2.2 Descrição da aplicação mobile para android

Nesta secção irão estar ilustradas todas as funcionalidades presentes da aplicação móvel para android que foi construída, com todos os seus detalhes.

A aplicação é reservada exclusivamente para utilizadores do tipo “normal”, não tendo qualquer UI implementada para utilizadores “admin” ou “vet”.

Para a exemplificação foram utilizados 2 smartphones: Samsung Galaxy S10 e Google Pixel 3XL.

2.2.1 Registo e Login

Dado um utilizador, o primeiro passo será criar uma conta, ou no caso de já possuir uma realização o login.

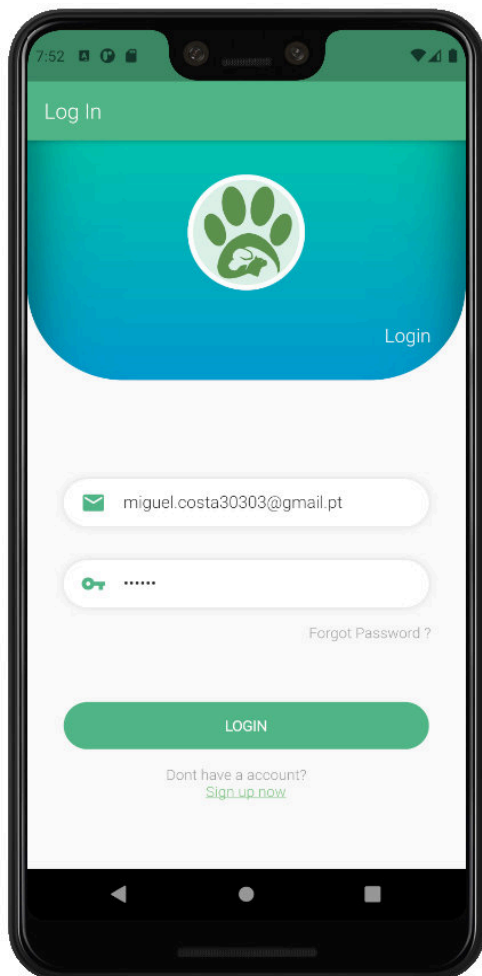


Figura 9. Exemplo de Login

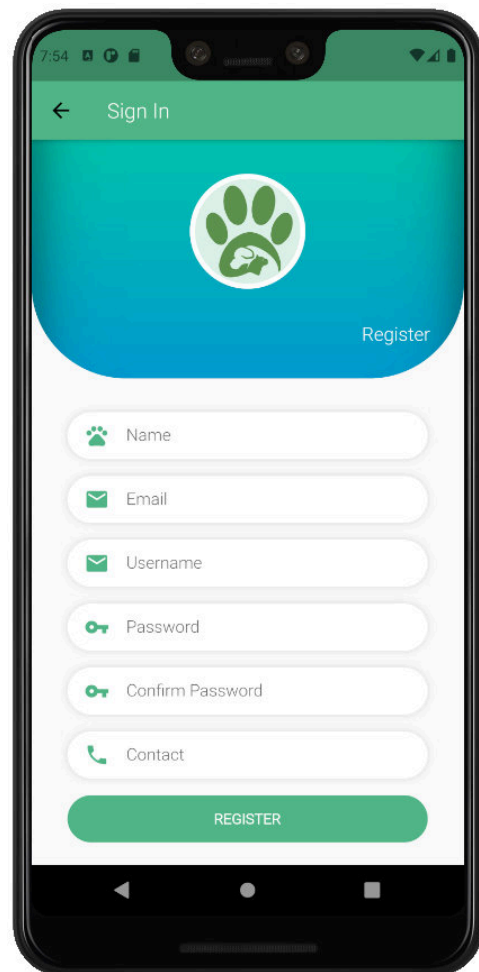


Figura 10. Exemplo de Registo

O utilizador deve todas colocar as informações pedidas, caso contrário irão ser acionadas as respectivas validações para cada um dos campos.

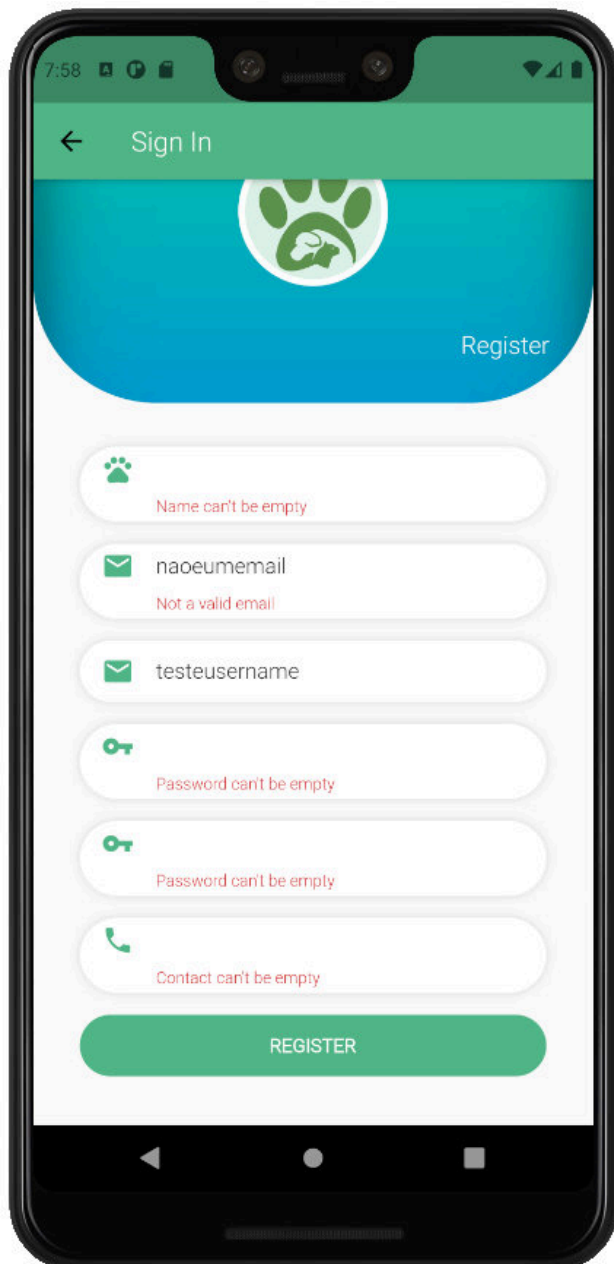


Figura 11. Validações

Todas as validações implementadas foram:

- Um dado campo não pode estar vazio ("Name can't be empty").
- Verificação de email.
- Verificação de contacto telefónico.
- Verificação para ver se as palavras passes coincidem ou não.
- Verificação se o username ou email já está registrado ou não.

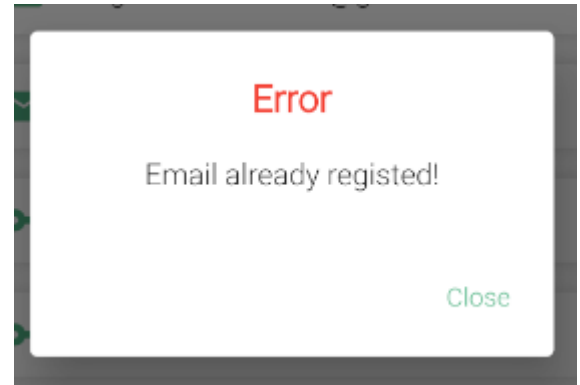


Figura 12. Validação do email

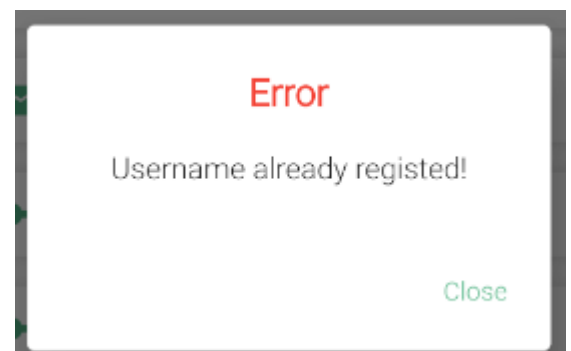


Figura 13. Validação do username

2.2.2 HomePage e Navegação

Quando um utilizador faz login ele será redirecionado para a homepage, onde estão presentes pequenos atalhos como a criação de um animal clicando no ícone de adicionar. Estão presentes também alguns animais do utilizador (no máximo 3) assim como as futuras consultas marcadas.

Toda a navegação é feita por uma barra de navegação presente no canto esquerdo do ecrã do telemóvel, que pode ser exposta clicando no "hamburger" ou arrastando para a direita (→).

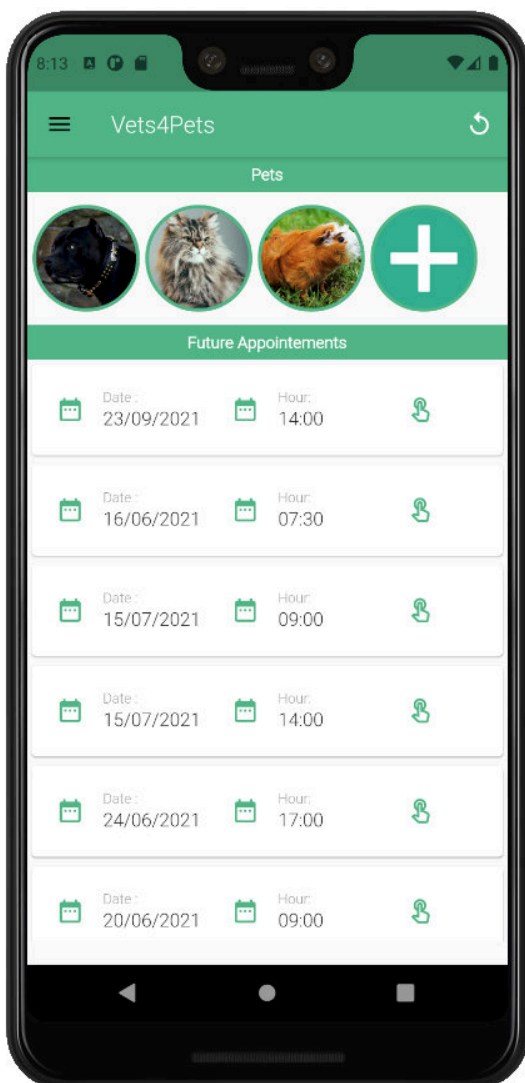


Figura 14. Aplicação - HomePage

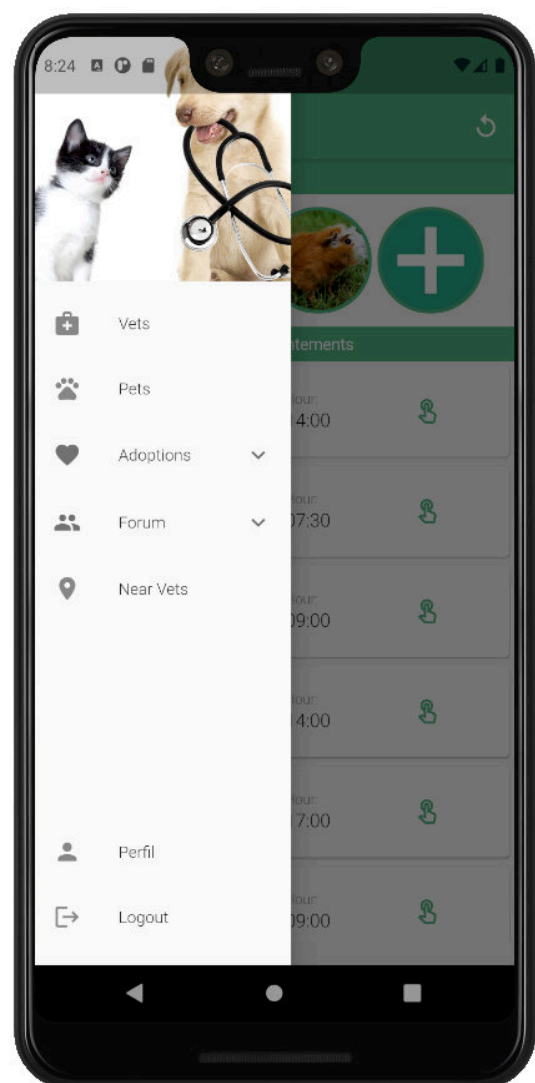
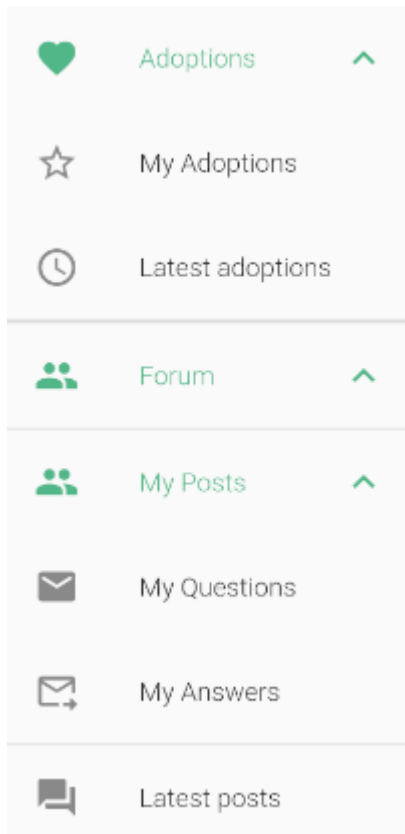


Figura 15. Aplicação - NavBar



Existem 7 componentes principais, das quais mais 5 são expansíveis, sendo assim são 13 no total, sendo elas:

- Aba Vets
- Aba Pets
- Adoptions
 - My Adoptions
 - Latest adoptions
- Fórum
 - My Posts
 - My Questions
 - My Answers
 - Latest posts
- Near Vets
- Perfil
- Logout

Figura 16. Abas escondidas

2.2.2.1 Abas

Breve explicação sobre cada aba:

Vets - Onde o utilizador poderá escolher uma clínica e marcar um consulta

Pets - Onde o utilizador poderá criar/editar remover os seus animais de estimação.

Adoptions - Onde o utilizador cria/edita/remove as suas adoções e consulta adoções de outros utilizadores.

Fórum - Onde o utilizador cria/edita/remove as suas perguntas/perguntas e consulta perguntas de outros utilizadores, assim como as respetivas respostas

Near Vets - Onde o utilizador vê as clínicas veterinárias perto da sua localização.

Perfil - Onde o utilizador edita o seu perfil e consulta todas as suas consultas e respetivos detalhes (consultas passadas e futuras).

Logout - Onde o utilizador termina a sessão, pois a aplicação possui um mecanismo de "Remember me".

2.2.3 Gestão dos animais de estimação

Uma das funcionalidades mais importantes é a gestão dos animais de estimação, onde o utilizador consegue listar, criar, editar e remover os seus animais de estimação também denominados como “pets”.

Para aceder a esta componente basta clicar na aba “Pets” da barra de navegação.

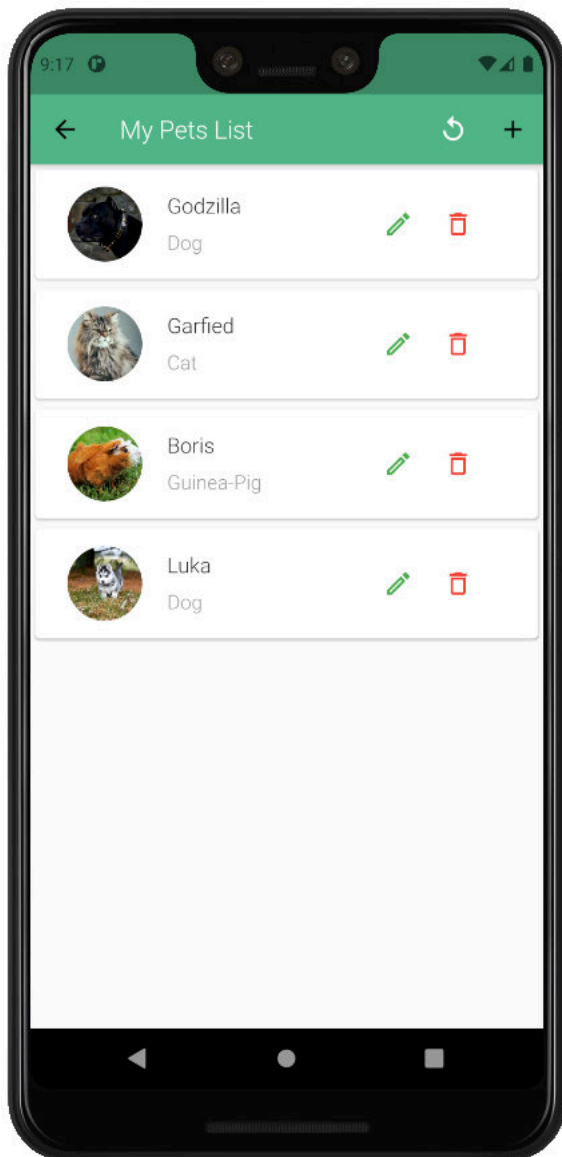


Figura 17. Animais do Utilizador

Ao clicar num dado animal irá surgir um pequeno perfil contendo as informações mais relevantes como tipo de animal de raça.

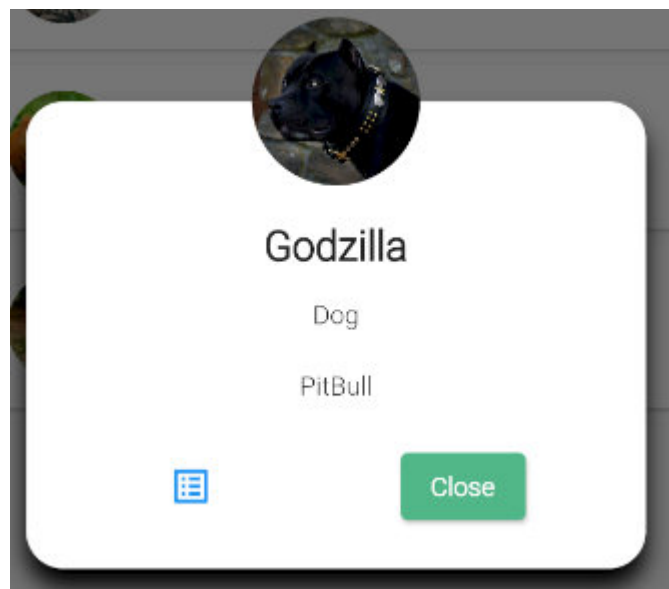


Figura 18. Perfil do animal

Aqui o utilizador poderá verificar as informações do animal registadas clicando no ícone azul de lista .

A adição de um animal é feita de forma simples, apenas é necessário clicar no ícone (+) no canto superior direito, introduzir o nome, tipo de animal e raça do animal de estimação. O utilizador, se pretender, poderá adicionar também uma foto de modo a construir um perfil mais apelativo. Para remoção e edição do animal apenas é necessário clicar no ícone **vermelho** 🗑 ou no **verde** ✎ .

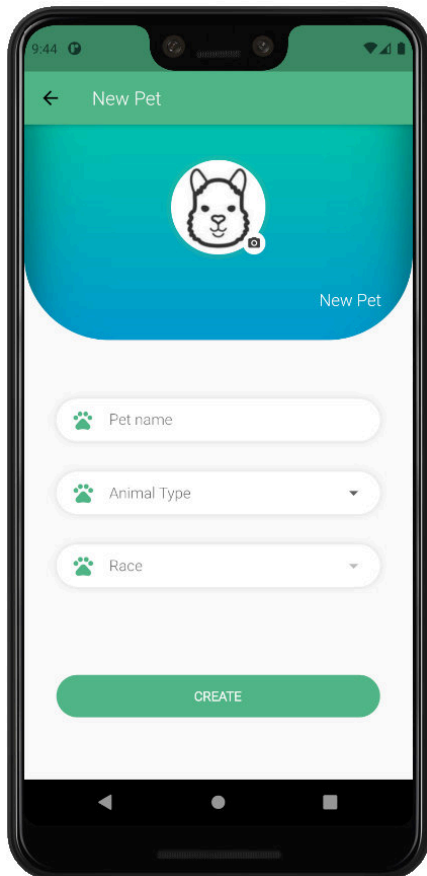


Figura 19. Adição de um animal

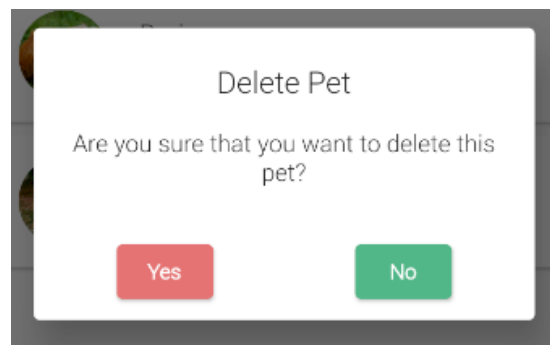


Figura 20. Remoção de um animal

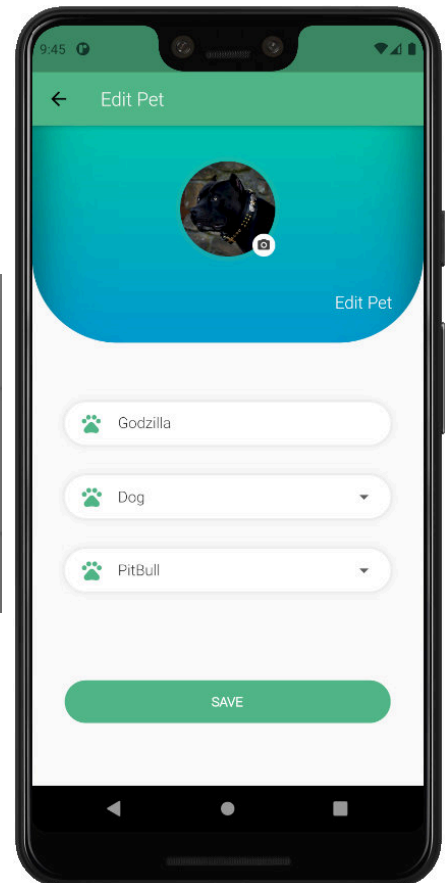


Figura 21. Edição de um animal

Existem todo o tipo de validações onde todos os atributos são necessários, apenas a foto é opcional, foto essa que deve ser escolhida pelo utilizador vinda da galeria do seu telemóvel.

2.2.3.1 Informações/Estatísticas de um animal de estimação

Como já foi referido, o utilizador poderá manter um registo de diversas informações sobre o seu animal, nomeadamente **vacinas** registadas, **histórico de consultas**, histórico esse que contém informações sobre:

- Data da consulta
- Peso do animal na consulta
- Preço da consulta
- Descrição da consulta
- Medicações prescritas e suas durações e horários.

Para aceder aos detalhes da consulta basta clicar no ícone **verde**  .

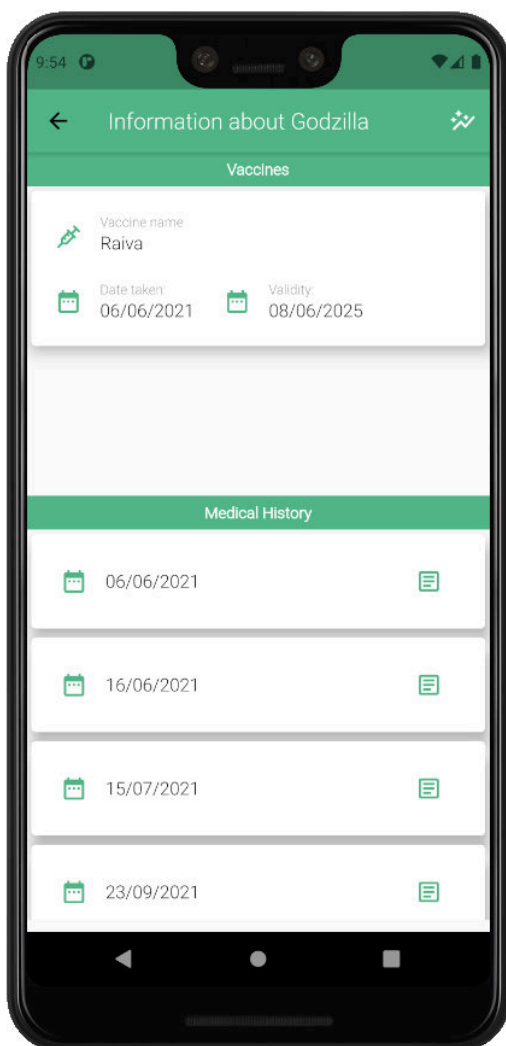


Figura 22. Informações gerais de um animal

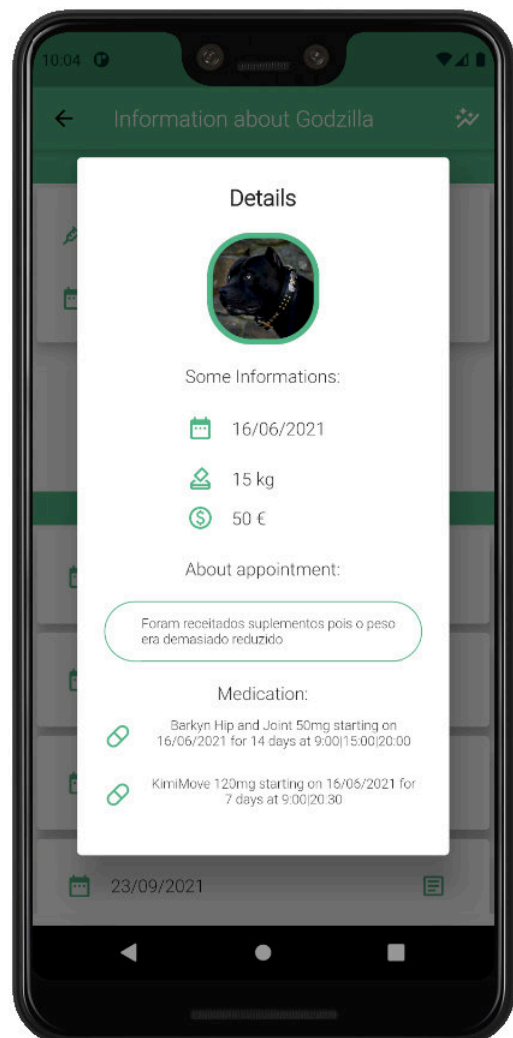
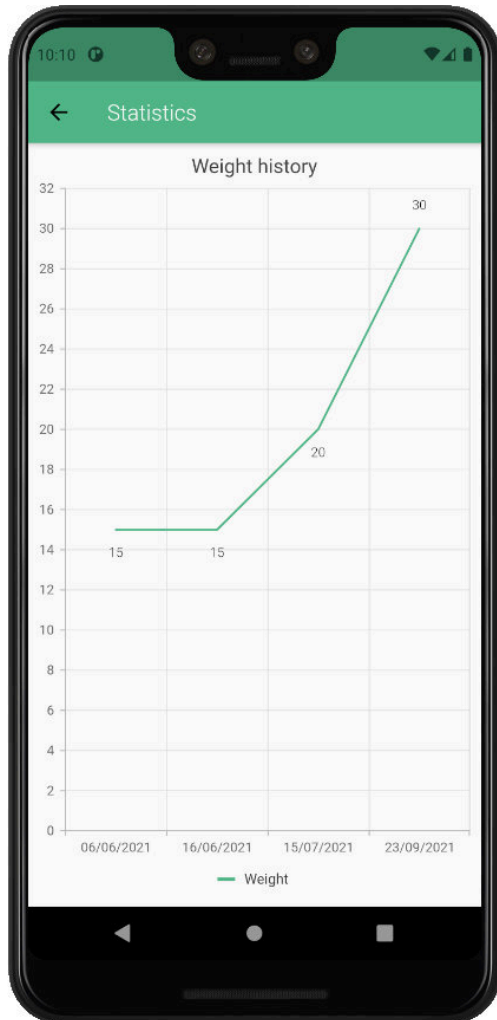


Figura 23. Detalhes da consulta de um animal

Também é possível verificar o peso de forma mais ilustrativa com recurso a um gráfico com a variação do peso à medida do tempo clicando no ícone do lado direito superior  .



Neste contexto, o animal do exemplo seria um animal que estaria abaixo do peso, no dia 16/06/2021 foi-lhe receitado suplementos de modo a aumentar o peso. Através da análise do gráfico é possível ver a curva de evolução do peso do animal nas 2 próximas consultas

Nota: Os valores do exemplo são valores futuros ao deste documentos pois são apenas valores teste.

Figura 24. Gráfico de peso do animal

2.2.3.2 Obtenção dos dados

Todos os dados previamente descritos são colocados na base de dados pelo respetivo veterinário da consulta, depois do animal efetuar a consulta é-lhe passado uma prescrição onde que é um somatório de todo o tipo de informação do animal naquela data. Esta prescrição é feita através do website na UI do veterinário.

2.2.4 Clínicas e consultas

Uma das principais, se não a mais importante, funcionalidade deste projeto é a marcação de consultas. Esta função encontra-se na aba “Vets”.

O processo de marcação de uma consulta é bastante simples, com o objetivo de ser user-friendly, onde, com apenas 6 “clicks” é possível a marcação de uma consulta numa determinada clínica com um determinado veterinário.

Clicando na aba “Vets” irá ser listado ao utilizador todas as clínicas registadas na aplicação, assim como o seu nome e localização. Seguidamente, o utilizador apenas tem de clicar na clínica desejada, onde irá surgir um pequeno perfil da clínica selecionada. Nesta página irão estar presentes diversos conjuntos de informação sobre a clínica, como por exemplo contactos, localização e uma pequena descrição. No canto superior direito ou no fundo da página encontram-se botões que redirecionam para a página de marcação de consulta.

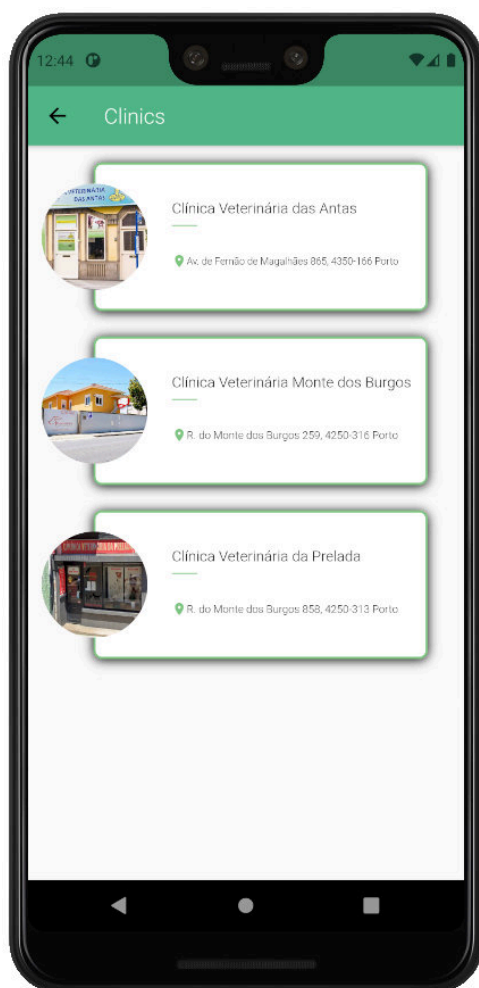


Figura 25. Listagem das clínicas

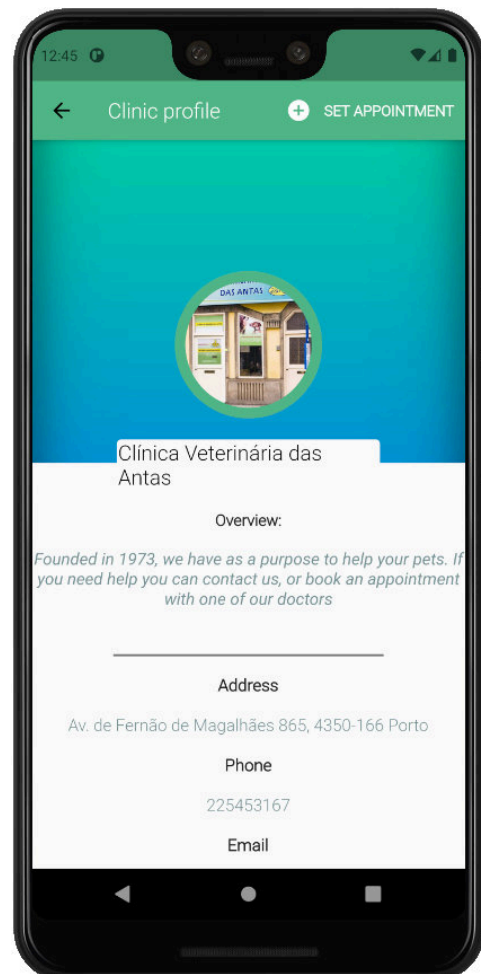


Figura 26. Perfil da clínica

2.2.4.1 Marcar uma consulta

A marcação de uma consulta é feita em 4 passos

1. Selecionar um Pet (os dos animais de estimação do utilizador).
2. Selecionar um Vet (os dos veterinários associados à clínica escolhida).
3. Selecionar um dia.
4. Selecionar uma hora (apenas os slots disponíveis).

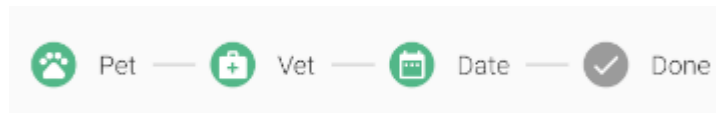


Figura 27. Passos para marcar uma consulta

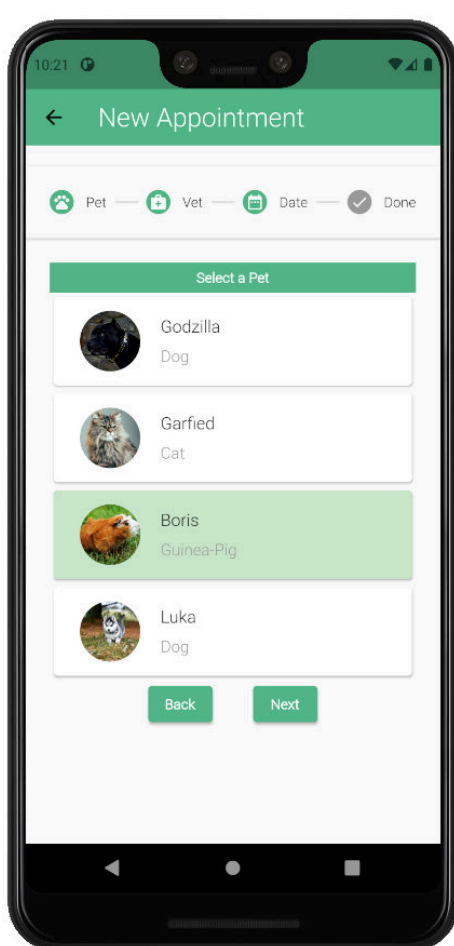


Figura 28. Selecionar animal

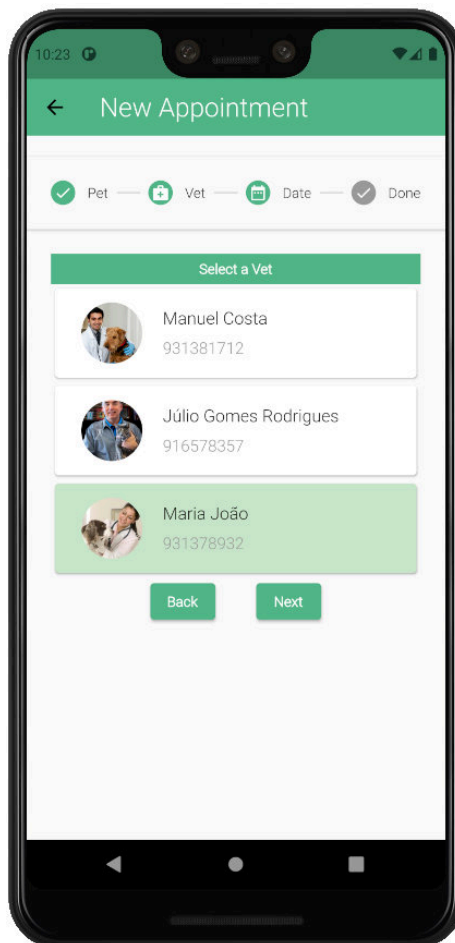


Figura 29. Selecionar veterinário

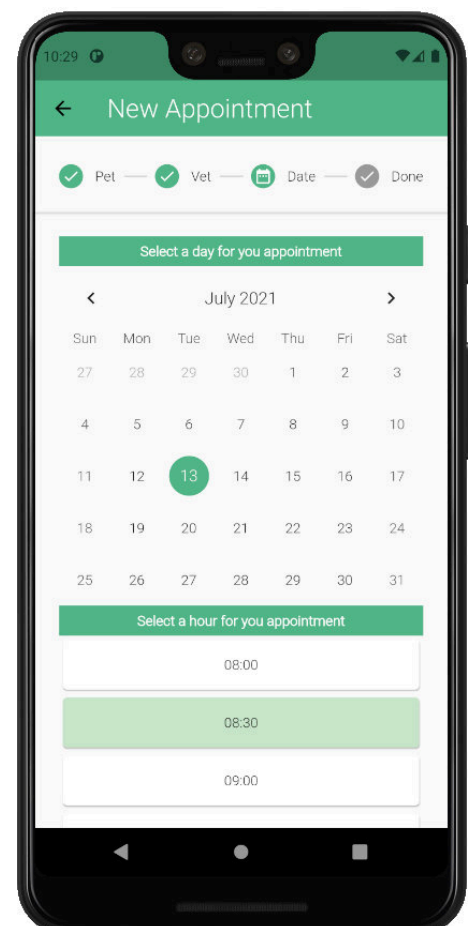


Figura 30. Selecionar data/hora

A seleção da data e horas é feita através de slots, onde só é apresentado ao utilizador os slots de horas disponíveis no dia selecionado. Esta implementação foi feita de modo a não causar duas ou mais consultas no mesmo intervalo de tempo.

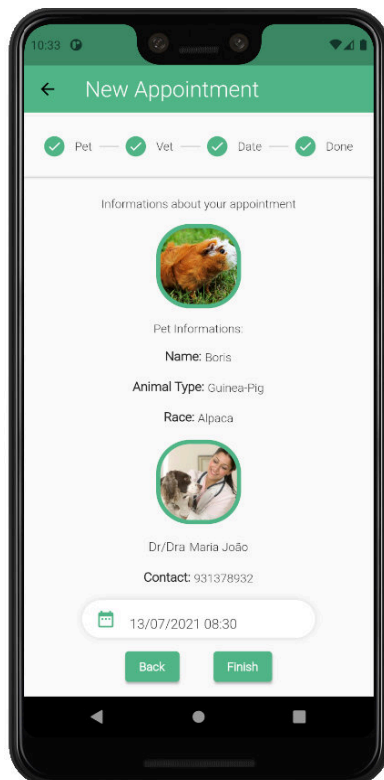


Figura 31. Detalhes da consulta

No final irá surgir uma tela onde é possível ver toda a informação sobre a consulta marcada, caso o utilizador deseje mudar alguma coisa ele poderá fazê-lo clicando em “Back”, caso contrário ele clicará em “Finish”, marcando a consulta.

Ao marcar a consulta, será aberto o calendário da Google caso o utilizador deseje guardar no seu calendário local para mais tarde receber um lembrete da consulta.

2.2.5 Perfil do utilizador

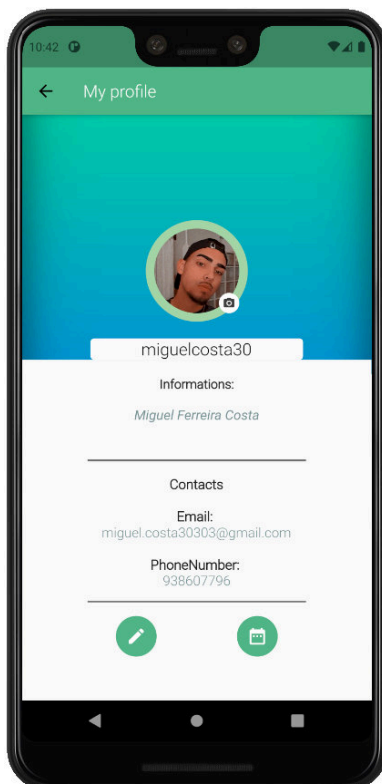


Figura 32. Perfil do utilizador

Nesta aba, o utilizador poderá consultar as suas próprias informações e fazer edições das mesmas.

Também poderá consultar todas as suas consultas realizadas na aplicação assim como informações detalhadas das mesmas, clicando no ícone do calendário .

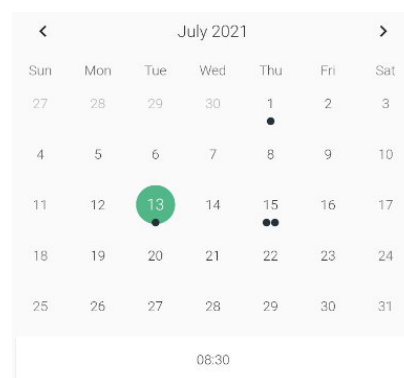


Figura 33. Calendário da aplicação

2.2.6 Fóruns

Como já foi referido, neste projeto existem dois tipos de fórum, um destinado para adoções de animais, onde é possível ver um conjunto animais que se encontram para adoção, sendo possível também ver o seu atual estado (adotado ou não adotado). O outro fórum faz referência a um fórum de perguntas e respostas, onde os vários utilizadores podem interagir entre si, fazer questões e responder, que assim como no fórum de adoções é possível ver o estado de uma data pergunta (fechada ou não fechada).

2.2.6.1 Fórum de Adoções

Encontra-se na aba “Adoptions” e encontra-se dividido nas respetivas abas “My Adoptions”, onde o utilizador gere as suas adoções, criando-as, removendo-as ou editando-as e “Latest Adoptions” onde é possível observar as últimas adoções registadas na aplicação, vendo também o seu estado (Adopted  ou Not Adopted ).

Cada adoção possui inúmeros tipos de informação para que um dado utilizador que esteja interessado naquele animal possa obter o máximo de informação possível.

2.2.6.1.1 Informações das adoções

Para ser criada uma adoção, o utilizador deve preencher um número de campos obrigatório (similar à criação dos “Pets”). Esses campos são:

- Título da adoção
- Tipo de animal
- Raça do animal
- Breve descrição do animal/razão da adoção
- Ano de nascimento do animal (Não é obrigatório)
- Email
- Contacto telefónico
- Cidade (onde está a ser feita a adoção)

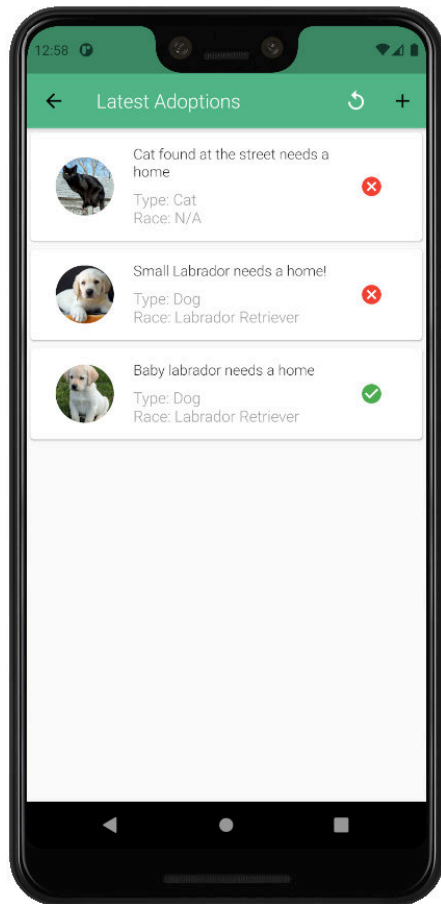


Figura 34. Últimas adoções

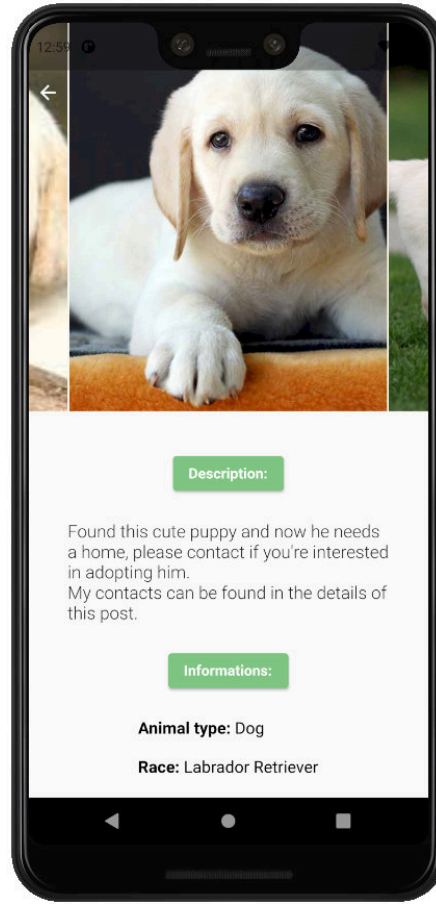


Figura 35. Informações da Adoção (1)

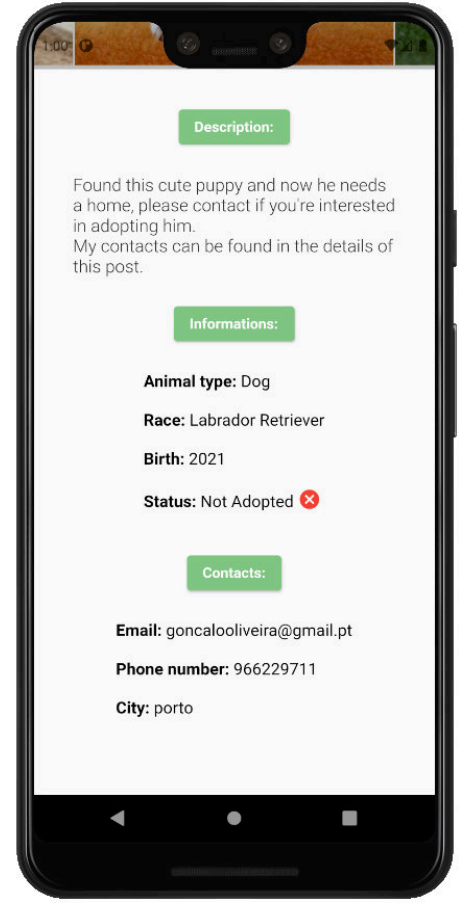


Figura 36. Informações da Adoção (2)

Se uma dada adoção de um dado utilizador for realizada, ou seja, alguém mostrar interesse no animal para adoção, este utilizador poderá fechar a adoção, colocando o animal como adotado.



Figura 37. Estados de uma adoção

2.2.6.2 Fórum de Perguntas e Respostas

Como já foi salientado, esta aba é onde todos os utilizadores podem interagir entre si, é possível criar, remover e editar o estado de uma pergunta, assim como criar e remover respostas.

Uma pergunta pode ter ou não ter uma imagem acoplada à pergunta.

Na aba My Posts, o utilizador pode ver todas as perguntas que realizou assim como todas as respostas, por outro lado na aba “Latest Posts” o utilizador tem acesso a uma listagem das últimas perguntas feitas por outros utilizadores sobre dúvidas/questões de mundo animal.



Figura 38. Fórum



Figura 39. Detalhes de uma pergunta

Observando a figura 36, é notório a presença de uma resposta diferente. Essa resposta foi feita por um veterinário registado na aplicação, sendo-lhe atribuída uma cor de destaque assim como um ícone de verificado . O veterinário possui acesso ao fórum através de website, componente essa que será explicada no decorrer deste documento.

Para responder a uma pergunta basta clicar no ícone de resposta presentes na caixa da pergunta ou no canto superior direito .

2.2.7 Veterinários perto de si

Por fim, a aplicação conta também com uma aba denominada “Near Vets” à qual o utilizador consegue ver todas as clínicas veterinárias perto de si, num raio de 20km. Esta implementação foi feita com o auxílio da API Google Maps.



Para poder utilizar esta funcionalidade o utilizador terá de dar permissões de localização à aplicação, caso contrário não lhe será mostrado um mapa.

Cada marcador vermelho representa uma clínica, contendo as informações mais relevantes da mesma (nome e morada).

Ao clicar num marcador, irão ser apresentados 2 ícones no canto inferior direito, e que, clicando no primeiro , será aberto o Google Maps com as direções para a clínica selecionada.

Figura 40. Veterinários perto de si

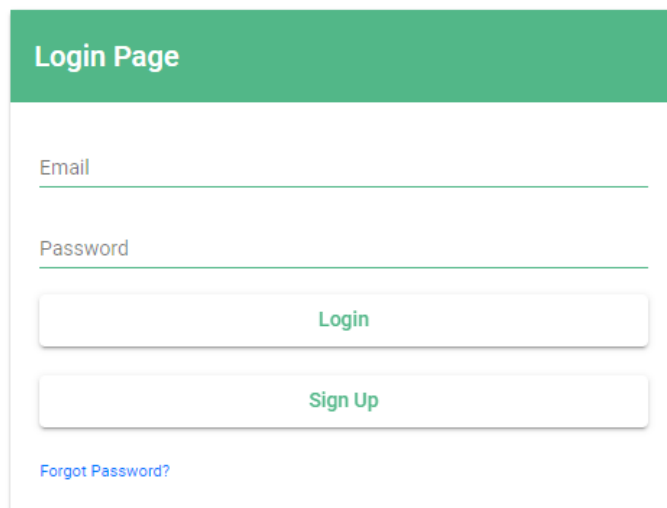
2.3 Descrição da aplicação web

Nesta secção irão estar ilustradas todas as funcionalidades presentes da aplicação web, que contém todas as funcionalidades da aplicação android a complementar com alguns extras.

A aplicação web pode ser utilizada por utilizadores do tipo “normal”, “vet” e “admin”.

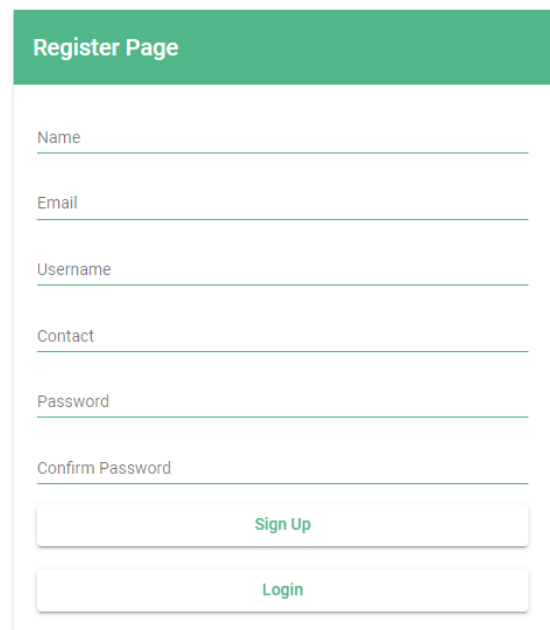
2.3.1 Registo e Login

Primeiro passo, tal como na aplicação android é criar uma conta, ou no caso de já possuir uma, realizar a autenticação. É também possível fazer a recuperação da password. Foram feitas também todas as validações que foram feitas na aplicação mobile em todos os formulários da aplicação web.



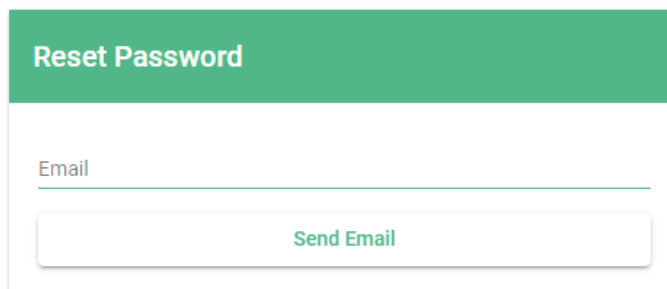
The Login Page form has a green header with the text 'Login Page'. Below the header, there are two input fields: 'Email' and 'Password'. Below the 'Password' field, there are two buttons: 'Login' and 'Sign Up'. At the bottom left, there is a link that says 'Forgot Password?'.

Figura 41. Formulário de Login



The Register Page form has a green header with the text 'Register Page'. Below the header, there are five input fields: 'Name', 'Email', 'Username', 'Contact', and 'Password'. Below the 'Password' field, there is another input field labeled 'Confirm Password'. Below these fields, there are two buttons: 'Sign Up' and 'Login'.

Figura 42. Formulário de Registo



The Reset Password form has a green header with the text 'Reset Password'. Below the header, there is one input field labeled 'Email'. Below the 'Email' field, there is a button labeled 'Send Email'.

Figura 43. Formulário de Alterar Password

2.3.2 HomePage e Navegação

Quando o utilizador faz login, dependendo do seu tipo ele é redirecionado para uma homepage diferente, não só os utilizadores de tipos diferentes têm homepages diferentes, mas também possuem uma barra lateral de navegação (navbar) diferente com diferentes possibilidades de navegação.

Na homepage do utilizador “normal” apenas é mostrado um pequeno atalho onde é possível clicar para ir para o perfil do animal, e um botão  onde é possível clicar para criar um novo animal.

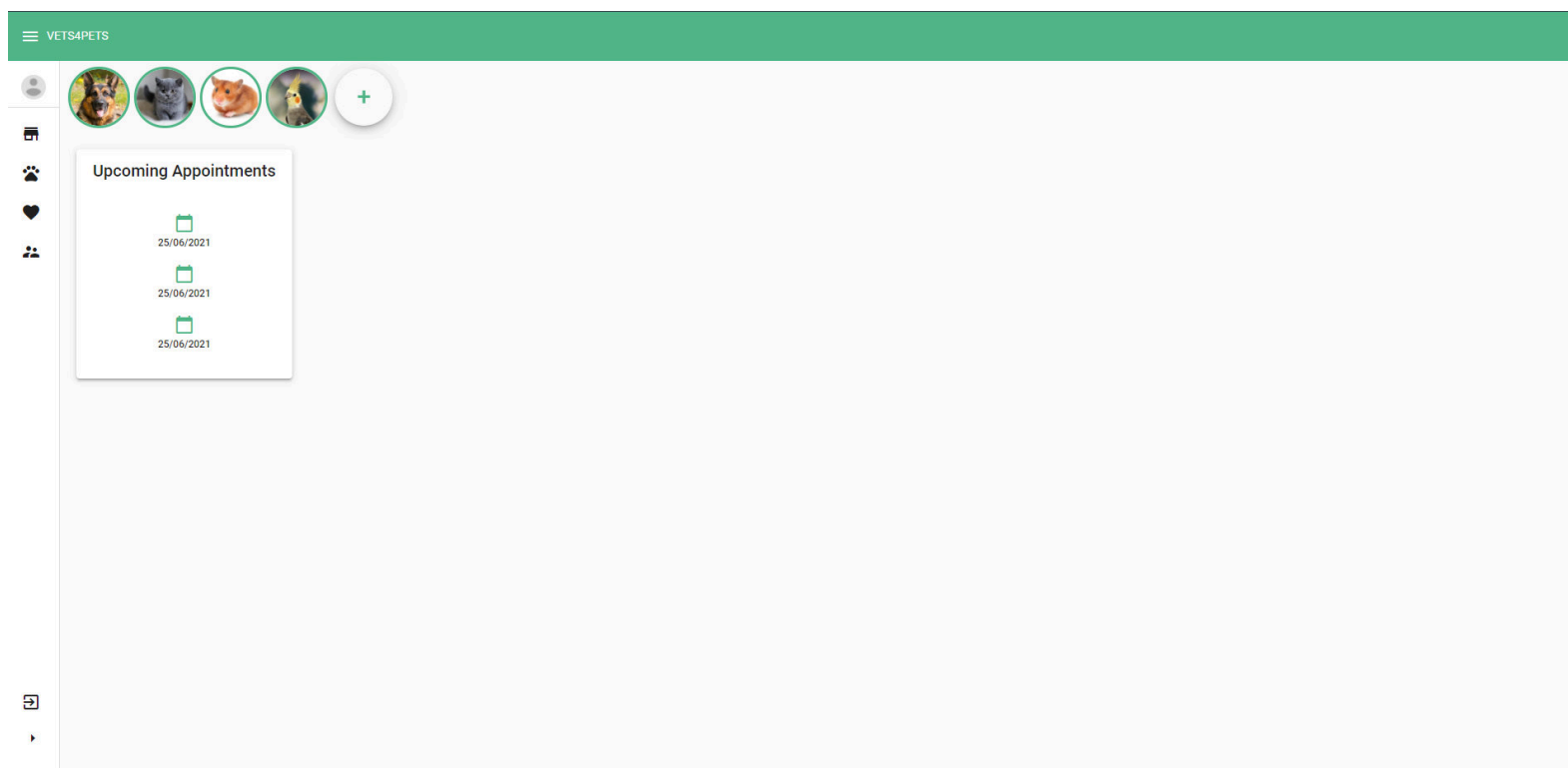


Figura 44. HomePage para Utilizador do tipo “normal”

Quando o utilizador a fazer login é do tipo “vet” é redirecionado para uma página onde é possível ver o seu calendário com todas as suas consultas. É possível mudar a vista do calendário para vista de semana, vista de dia, ou vista de mês.

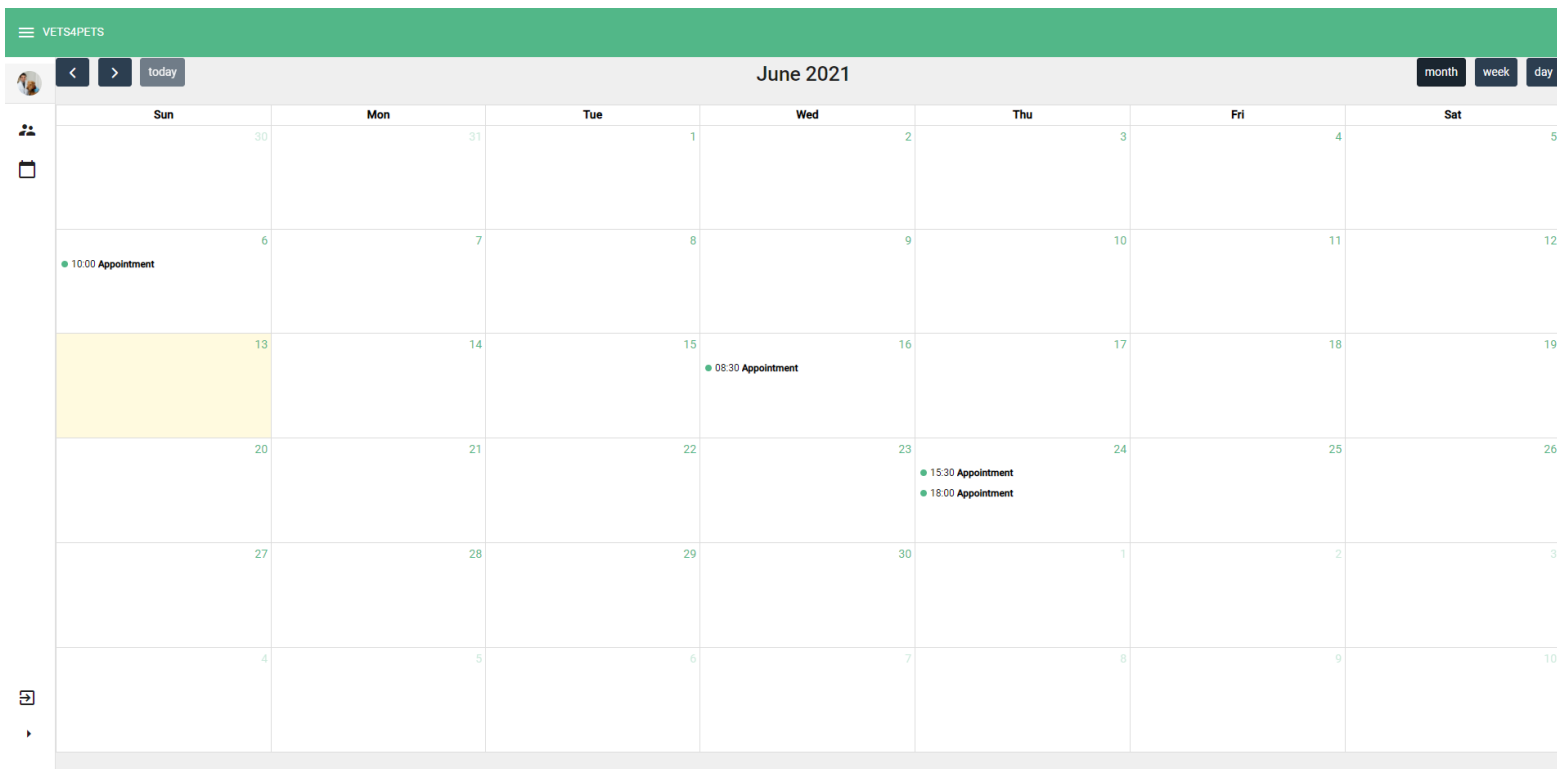


Figura 45. HomePage para Utilizador do tipo “vet”

Já quando o utilizador a fazer login é um admin, ele é redirecionado para a seguinte página. Esta página contém uma tabela, onde o administrador consegue adicionar, editar e remover clínicas. É também possível adicionar e/ou remover veterinários a essas clínicas.

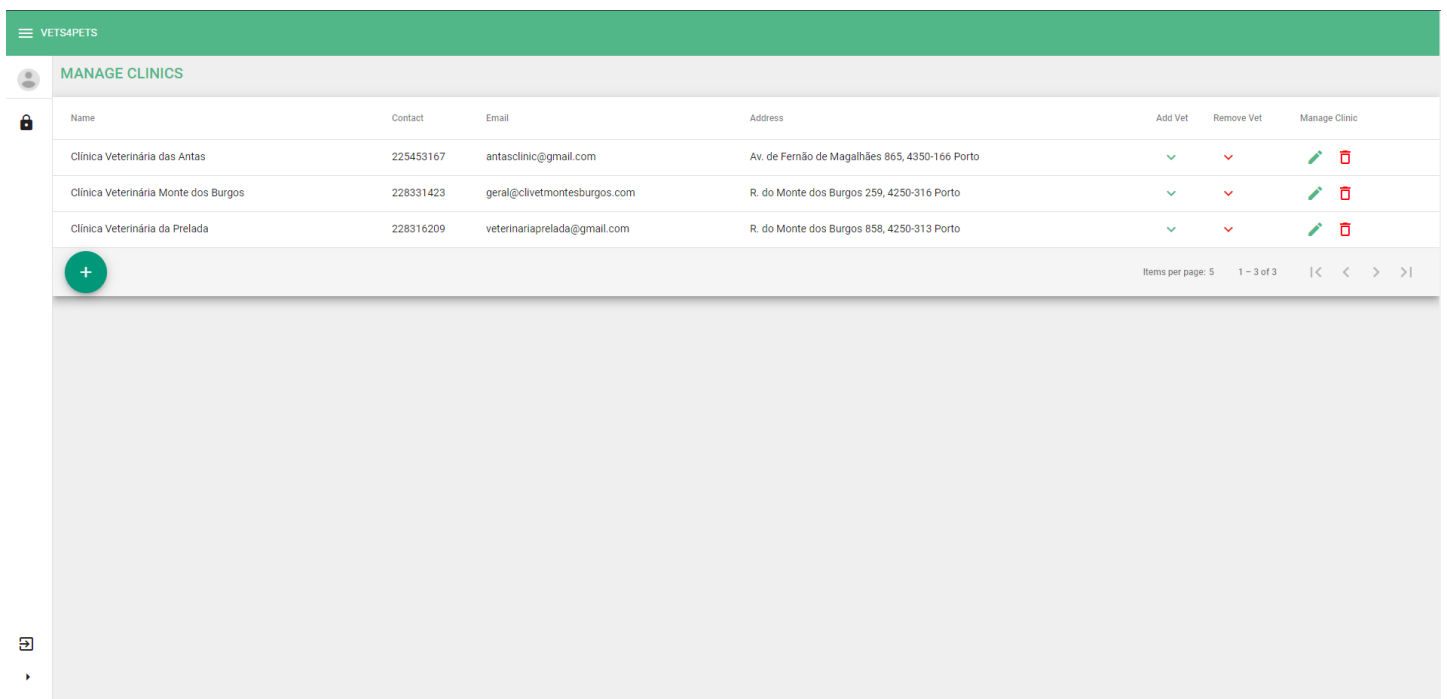


Figura 46. HomePage para Utilizador do tipo “admin”

2.3.3 Gestão dos animais de estimação

Como também já foi demonstrado na aplicação, o utilizador consegue adicionar, remover, editar e listar os seus animais de estimação.

Para aceder a esta página basta clicar  na navbar. Aqui vai conseguir ver todos os seus animais registados na aplicação.

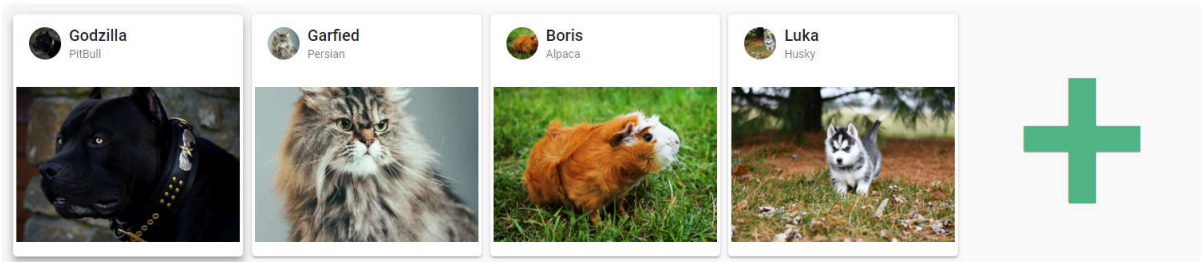


Figura 47. Lista de Animais

Clicando no botão com o (+) é aberto um formulário para proceder à adição do animal.

Clicando em qualquer um dos animais, vai ser levado para o seu perfil.

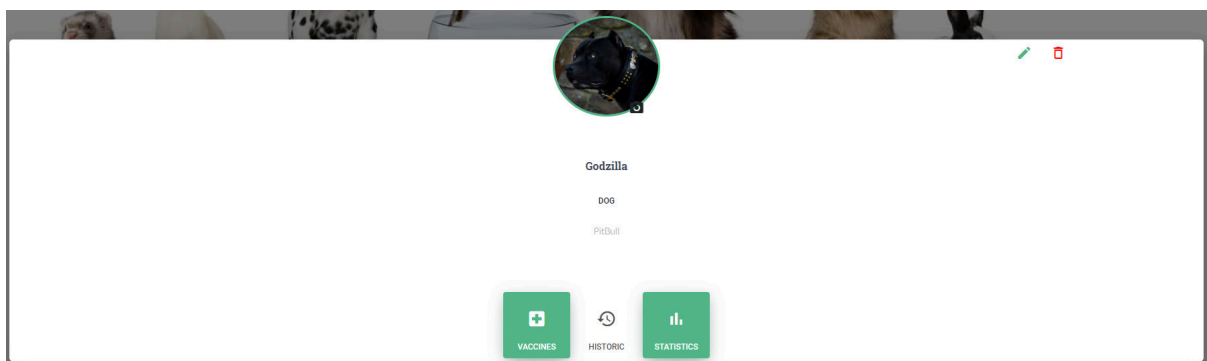


Figura 48. Perfil de um animal

Aqui é possível fazer operações relativas ao animal, como clicando no ícone **vermelho**  para remover o animal ou no **verde**  para editar as informações do animal.

2.3.3.1 Informações/Estatísticas de um animal de estimação

Como já foi referido, o utilizador consegue aceder ao perfil do animal, neste perfil é possível ver informações que são mantidas sobre o seu animal, como **vacinas** registadas, **histórico de consultas**.

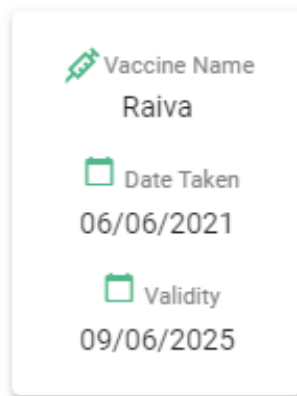


Figura 49. Vacina

Aqui é possível ver um exemplo de vacina tomada pelo animal, e também os detalhes de uma consulta na qual foi prescrita alguma medicação para o animal.

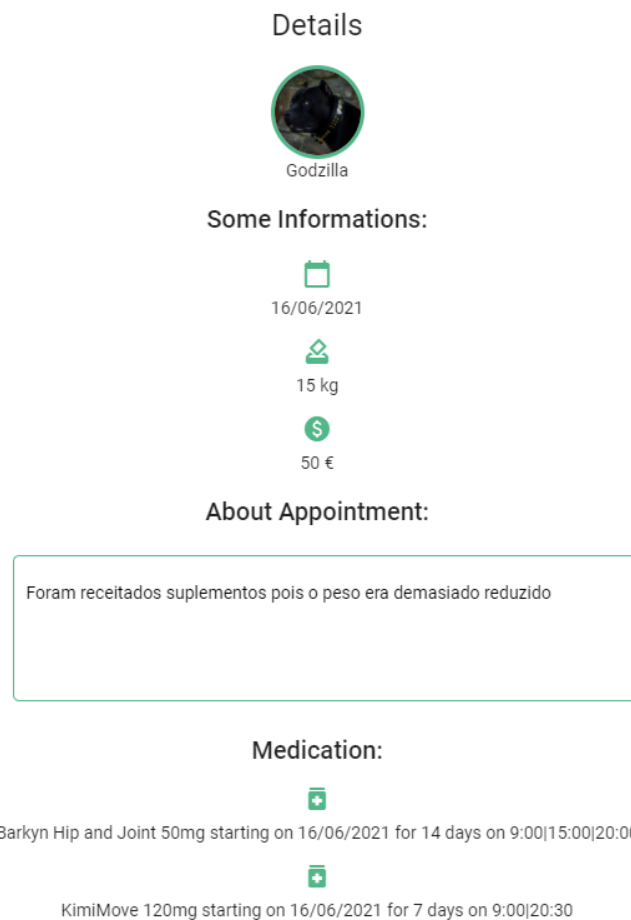


Figura 50. Detalhes da consulta do animal

É também possível ainda no perfil do animal, ver um peso com a evolução do peso de um animal, ao longo das suas consultas.

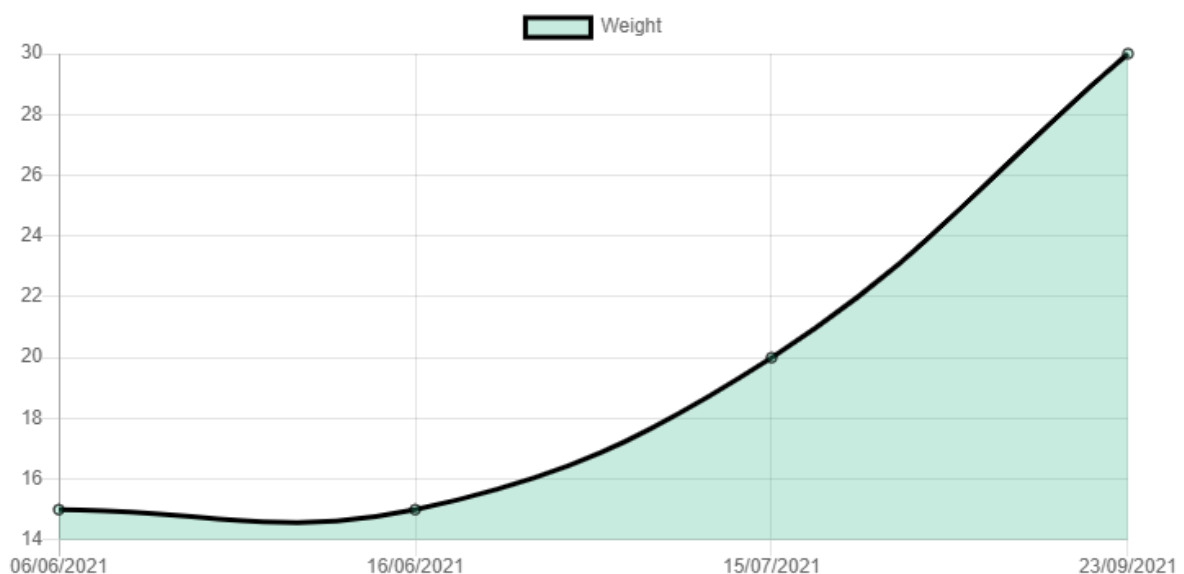
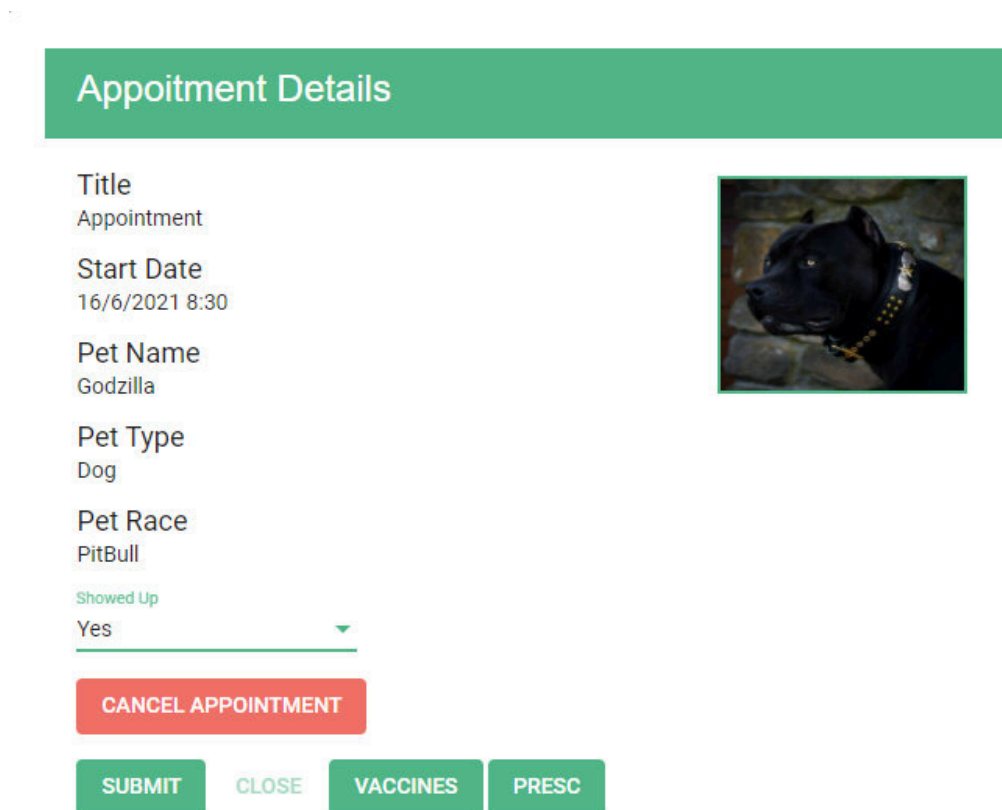


Figura 51. Gráfico de peso do animal

2.3.3.2 Obtenção dos dados

Todos os dados são obtidos através do UI do veterinário. Como já foi previamente referido, o veterinário possui um calendário de consultas. Clicando numa das consultas aparece uma pequena descrição sobre a consulta, data, hora e animal. Aqui o veterinário consegue dizer se o animal compareceu ou não na consulta, cancelar a consulta, adicionar vacinas e também adicionar uma prescrição.



Appoitment Details

Title
Appointment

Start Date
16/6/2021 8:30

Pet Name
Godzilla

Pet Type
Dog

Pet Race
PitBull

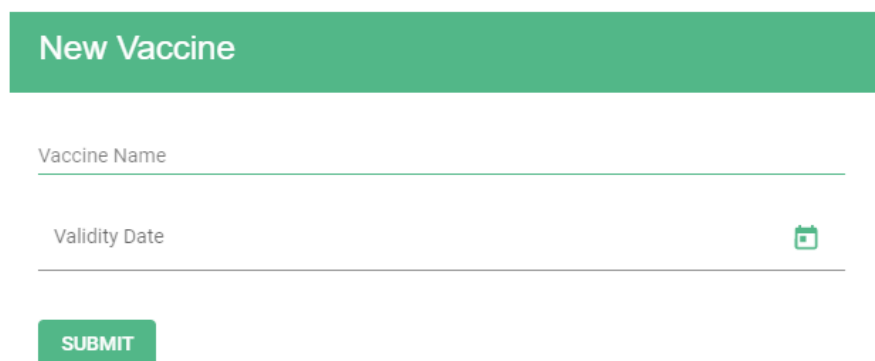
Showed Up
Yes

CANCEL APPOINTMENT

SUBMIT **CLOSE** **VACCINES** **PRESC**

Figura 52 - Detalhes da Consulta

É possível adicionar uma vacina clicando no botão **"Vaccines"**.



New Vaccine

Vaccine Name

Validity Date

SUBMIT

Figura 53. Formulário de adição de vacina

É ainda possível adicionar medicação à consulta clicando no botão **“Presc”**. O veterinário pode prescrever medicação ao animal aparecendo da primeira vez apenas um campo para medicação. Se for preciso é possível fazê-lo clicando no botão **“Add Medicine”**. Isto cria um linha igual à primeira, sendo possível remover clicando no **✗**.

Os únicos campos que não podem faltar são o peso, preço e uma pequena descrição da consulta.

Figura 54. Formulário de adição de medicação

2.3.4 Clínicas e consultas

Como já foi demonstrado anteriormente na aplicação é possível marcar uma consulta. Para este feito, basta apenas clicar  na navbar. Aqui aparecem todas as clínicas registadas na aplicação.



Figura 55. Lista de Clínicas

Clicando em qualquer uma das clínicas, vai ser levado para o seu perfil.



Figura 56. Perfil da Clínica

2.3.4.1 Marcar uma consulta

A marcação de consulta é feita em 4 passos

1. Selecionar um Pet.
2. Selecionar um Vet.
3. Selecionar um dia e uma hora (Apenas horas disponíveis naquele dia).



Figura 57. Passos para marcar uma consulta

A seleção das horas é feita tal e qual como na aplicação, aparecendo apenas as horas disponíveis no dia selecionado.

No final surge um pequeno cartão onde é possível ver todos os detalhes da consulta, como o Veterinário da consulta, o Pet da consulta, e a data e hora de começo da consulta.

Ao fim da marcação da consulta o utilizador consegue visualizar todas as marcações feitas no seu perfil.

SUCCESS, CHECK YOUR PROFILE TO SEE ALL YOUR APPOINTMENTS

VET PICTURE



VET NAME

Dr.Manuel Costa

PET PICTURE



PET NAME

Godzilla

DATE

16/6/2021 16:30

Figura 58. Detalhes da Consulta no fim da marcação

2.3.5 Perfil do utilizador

No perfil do utilizador é onde o utilizador consegue consultar as suas informações. Como já referido pode ver todas as suas consultas marcadas na aplicação, informações detalhadas sobre as mesmas e cancelar a consulta.

Appoitment Details

Title
Doctors Appointment

Start Date
16/6/2021 16:30

Pet Name
Godzilla

Pet Type
Dog

Pet Race
PitBull

CLOSE

CANCEL APPOINTMENT

Figura 59. Detalhes da Consulta do lado do Utilizador

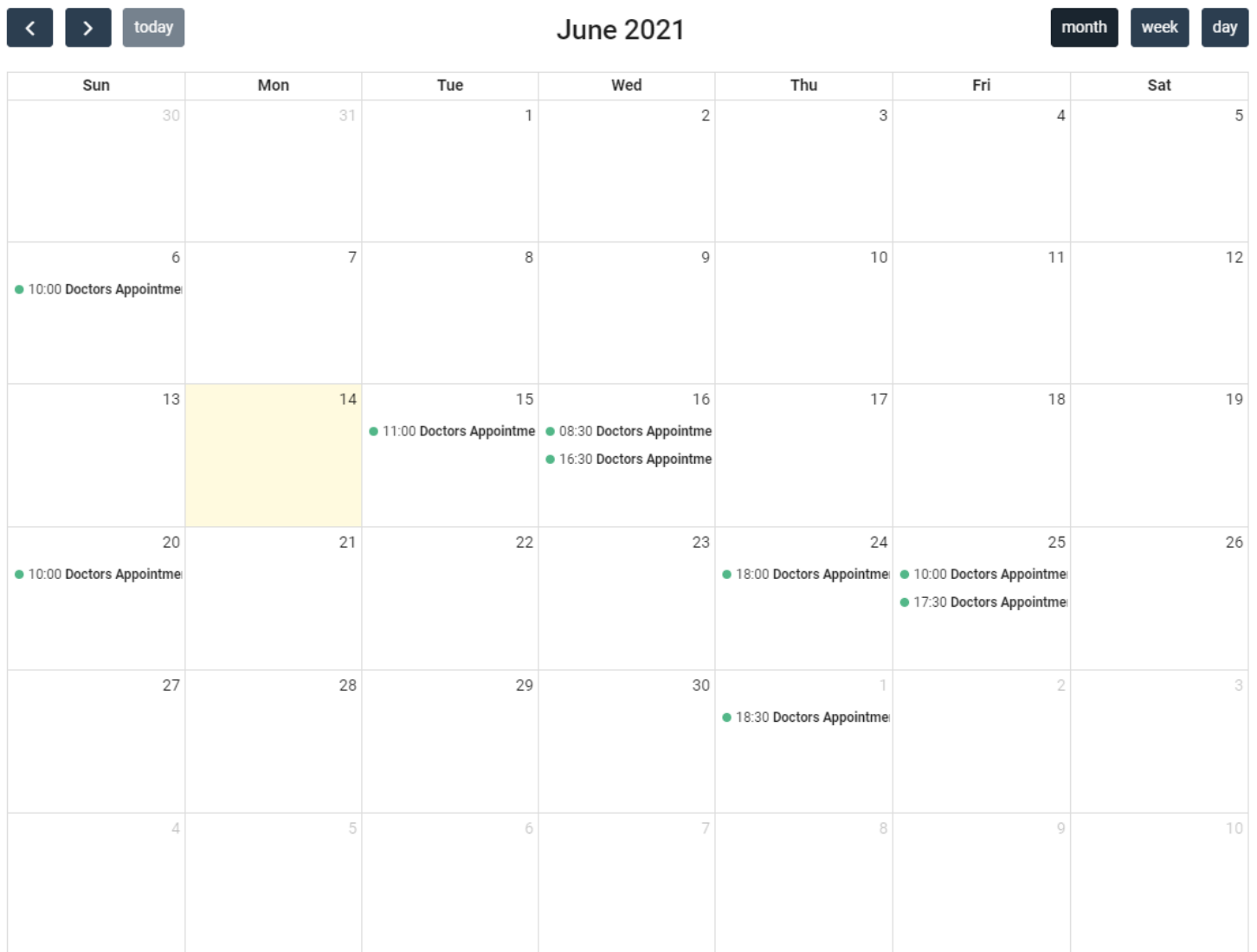


Figura 60. Todas as consultas marcadas do Utilizador

2.3.6 Fóruns

Bem como na aplicação existem dois tipos de fóruns, um para adopções, onde é possível ver um conjunto de animais que se encontra para adopções. O outro fórum é de perguntas e respostas, onde os utilizadores conseguem interagir entre si para fazer questões e responder a questões. É também possível alterar o estado da adopção (adoptado ou não adoptado) e também alterar o estado da pergunta (aberta ou fechada).

2.3.6.1 Fórum de Adoções

É possível aceder às adoções através da navbar clicando . O utilizador terá duas opções “Latest Adoptions”, onde é possível ver as últimas adoções que foram registadas na aplicação e também possui a opção “My Adoptions” onde consegue adicionar, editar e remover adoções.

Menu	Last Adoptions				
🕒 Last Adoptions ★ My Adoptions	Name	Picture	Animal Type	Race	Birth
	Cat found at the street needs a home		Cat	N/A	2021
	Small Labrador needs a home!		Dog	Labrador Retriever	2021
	Baby labrador needs a home		Dog	Labrador Retriever	2021
<div>Items per page: 6 1 - 3 of 3 < < > > </div>					

Figura 61. Página de Adoções

2.3.6.1.1 Informações das adoções

Adoption Details

Name
Cat found at the street needs a home

Birth
2021

Animal Type
Cat

Race
N/A

Description
Cat found at the street needs a new home. Cat is extremely gentle and has all the vaccination up to date. If you're interested contact me

Phone number/Email
937171681/diogosilva@gmail.pt

City
viana_do_castelo






CLOSE

Figura 62. Página de Adoções

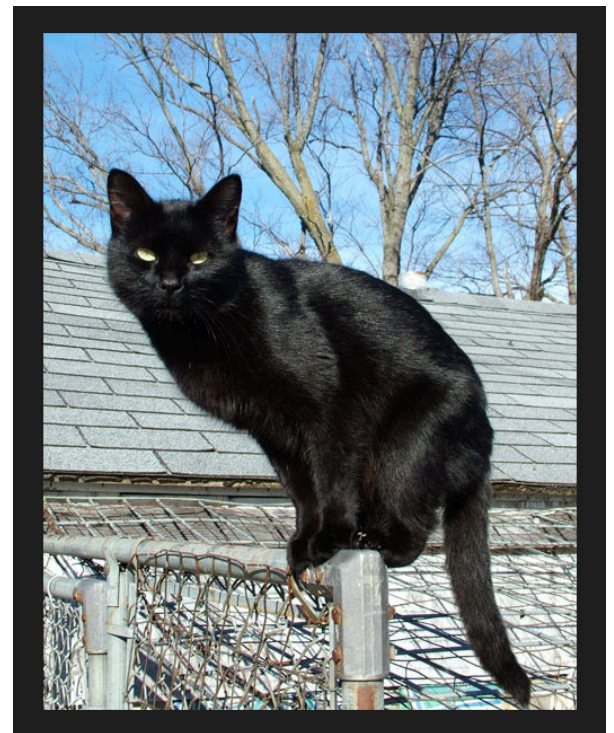


Figura 63. Foto Adoção Maior

As adoções possuem as mesmas informações que estão na aplicação sobre as adoções, sendo também possível visualizar as fotos em maior qualidade clicando nas fotos. As adoções podem também ser fechadas, caso o animal tenha sido adotado.

2.3.6.2 Fórum de Perguntas e Respostas

Como já referido o fórum de perguntas e respostas é também muito parecido com o da aplicação android. Apenas de referir que quando o utilizador é vet aparece o símbolo  a dizer que o utilizador é verificado.

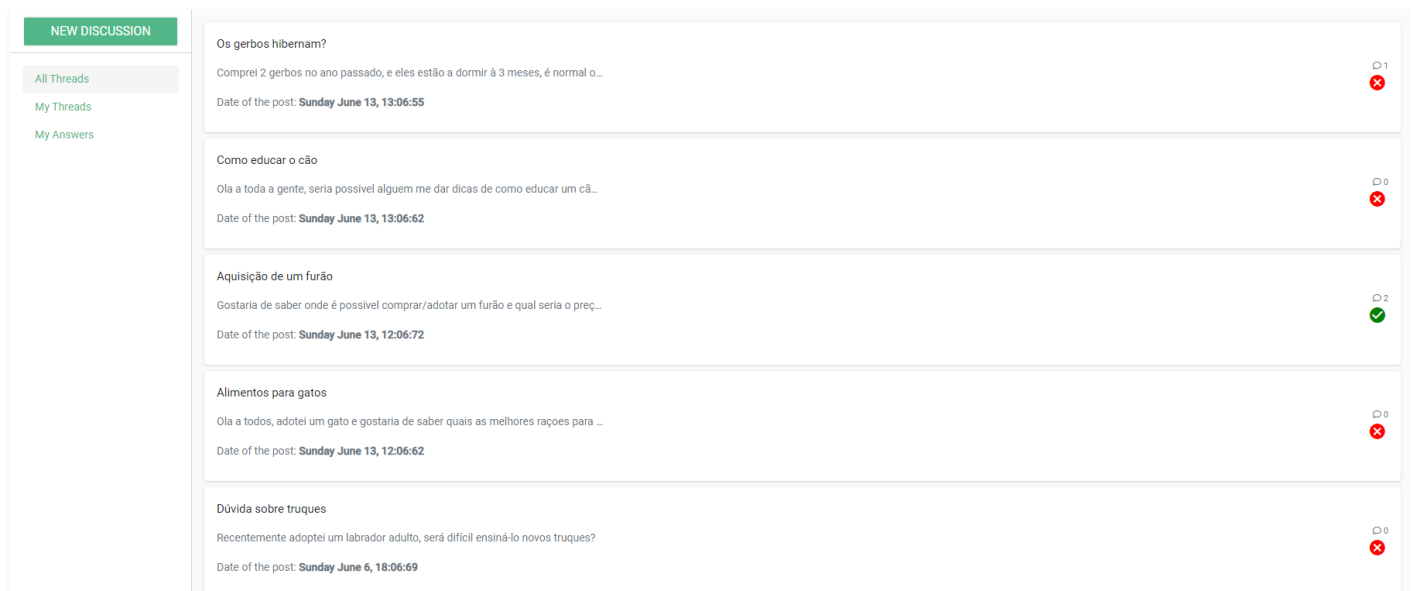


Figura 64. Fórum de Perguntas e Respostas

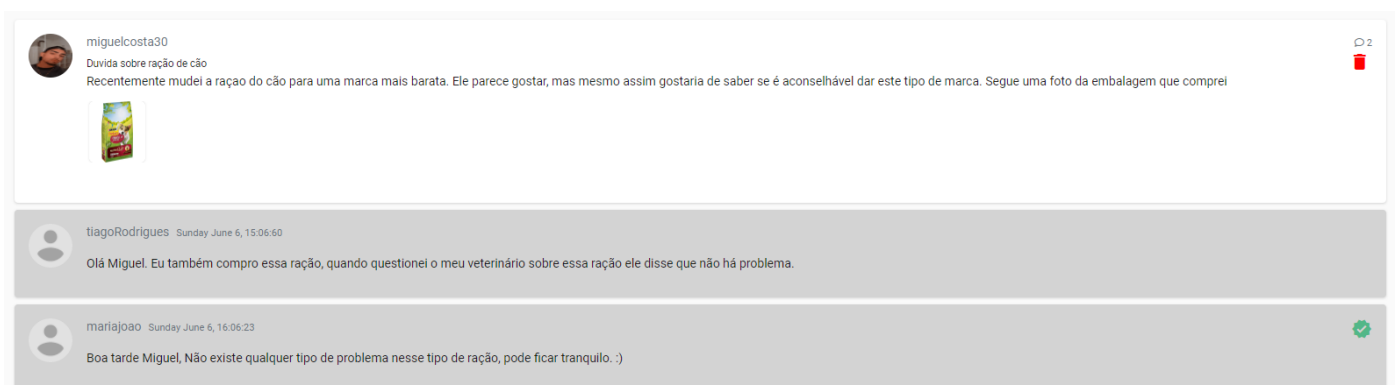


Figura 65. Detalhes de Pergunta

3. Requisitos

Neste capítulo serão apresentados todos os requisitos funcionais e não funcionais do projeto, descrevendo detalhadamente o sistema e as suas funcionalidades.

3.1 Requisitos Funcionais

3.1.1 Aplicação Móvel

ID: RF1.1

Título: Autenticação de Utilizadores

Descrição: As funcionalidades da aplicação apenas podem ser acessíveis por Utilizadores Autenticados, pelo que todos (independentemente do seu nível) terão de se autenticar para aceder à HomePage.

ID: RF1.1.1

Título: Autenticação de Utilizadores deve ser duradoura

Descrição: De como a facilitar a utilização da aplicação, existe um mecanismo de “remember me” onde o utilizador não necessita de fazer login sempre que abre a aplicação.

Dependências: RF1.1

ID: RF1.1.2

Título: Término de sessão

Descrição: O utilizador poderá terminar a sua sessão sempre que desejar, caso deseje mudar de telemóvel ou por outro motivo qualquer.

Dependências: RF1.1

ID: RF1.2

Título: Registo de Utilizadores

Descrição: No caso de ausência de uma conta, o Utilizador poderá criar uma, colocando todo o tipo de informação necessário para a realização deste processo.

ID: RF1.3

Título: Apresentação de informação.

Descrição: O utilizador poderá, após autenticação, visualizar as suas consultas futuras e alguns dos seus animais de estimação, possuindo também um atalho para adicionar outro animal caso pretenda .

Dependências: RF1.1

ID: RF1.4

Título: Acesso a várias abas, cada uma com a sua funcionalidade.

Descrição: O utilizador poderá fazer inúmeras operações após fazer a sua autenticação, podendo navegar na aplicação através de uma barra de navegação.

Dependências: RF1.1

Gestão de animais de estimação

ID: RF1.4.

Título: Gestão dos animais.

Descrição: O utilizador poderá visualizar todos os seus animais de estimação e realizar inúmeras operações.

Dependências: RF1.1

ID: RF1.4.1

Título: Criação de um animal.

Descrição: O utilizador poderá criar um animal de estimação que será uma que atua como uma cópia virtual do seu animal verdadeiro. Ele poderá inserir uma pequena foto de para uma visualização mais apelativa, assim como as informações mais relevantes (nome, tipo de animal, raça).

Dependências: RF1.1, RF1.4

ID: RF1.4.2

Título: Remoção de um animal.

Descrição: O utilizador poderá remover um animal de estimação, caso se tenha enganado na criação ou por qualquer outro motivo.

Dependências: RF1.1, RF1.4, RF1.4.1

ID: RF1.4.3

Título: Edição de um animal.

Descrição: O utilizador poderá editar um animal de estimação. Poderá editar todo o tipo de informação (nome, tipo de animal, raça) assim como a sua foto de perfil também.

Dependências: RF1.1, RF1.4, RF1.4.1

ID: RF1.4.3

Título: Visualização de dados de um animal.

Descrição: O utilizador poderá visualizar informações sobre o seu animal de estimação.

Dependências: RF1.1, RF1.4, RF1.4.1

ID: RF1.4.3.1

Título: Visualizar as vacinas.

Descrição: O utilizador poderá visualizar todas as vacinas registadas na aplicação para um dado animal, vacinas essas que foram registadas por um veterinário do lado do website.

Dependências: RF1.1, RF1.4, RF1.4.1, RF1.4.3

ID: RF1.4.3.1

Título: Visualizar histórico de consultas.

Descrição: O utilizador poderá visualizar todas antigas de um determinado animal. Poderá também obter informação detalhada sobre uma dada consulta, nomeadamente o veterinário da consulta, o que foi feito na consulta e a medicação que foi receitada naquela consulta (caso tenha sido receitada), peso do animal na data da consulta e preço da consulta.

Dependências: RF1.1, RF1.4, RF1.4.1, RF1.4.3

ID: RF1.4.3.1

Título: Acompanhamento de dados estatísticos do animal.

Descrição: O utilizador poderá acompanhar os dados estatísticos do seu animal, nomeadamente a evolução do peso do animal ao longo das várias consultas. Isto pode ser feito através da visualização de um gráfico.

Dependências: RF1.1, RF1.4, RF1.4.1, RF1.4.3

Clínicas e Consultas

ID: RF1.5

Título: Marcação de consultas numa determinada clínica.

Descrição: O utilizador poderá marcar uma consulta de forma rápida e eficaz, seleccionando a clínica que deseja, clínica essa que se encontra registada na aplicação.

Dependências: RF1.1, RF1.4

ID: RF1.5.1

Título: Visualização de informações clínicas.

Descrição: O Utilizador poderá visualizar diversas informações sobre uma determinada clínica, nomeadamente uma descrição, contactos e morada.

Dependências: RF1.1, RF1.4, RF1.5

ID: RF1.5.2

Título: Passos para marcação de uma consulta.

Descrição: Aposto selecionar a clínica desejada, através de uma interface bastante user-friendly, o utilizador cria uma consulta , primeiramente selecionando os dos seus animais de estimação, de seguida um veterinário registado que está associado à clínica selecionada e por fim data e hora desejada. A data e hora serão apenas os “slots” livres do veterinário escolhido .

Dependências: RF1.1, RF1.4,RF1.5

ID: RF1.5.3

Título: Notificação de uma dada consulta.

Descrição: Após marcar uma consulta, automaticamente será sugerido ao utilizador guardar um lembrete dessa consulta no calendário local do dispositivo, para no futuro receber uma notificação/lembrete dessa consulta.

Dependências: RF1.1, RF1.4,RF1.5

Perfil do utilizador

ID: RF1.5

Título: Gestão do Perfil do Utilizador.

Descrição: O utilizador possui uma aba para todo o tipo de edições do perfil (por exemplo mudar a foto de perfil para uma visualização mais apelativa). Poderá também consultar todos as consultas de todos os seus animais, tanto consultas passadas como consultas futuras.

Dependências: RF1.1

Fórum de Adoções

ID: RF1.6

Título: Fórum de Adoções.

Descrição: O utilizador poderá interagir com um fórum de adoções bastante simples.

Dependências: RF1.1

_____ **ID:** RF1.6.1

Título: Adoções.

Descrição: O Utilizador terá acesso a uma lista de adoções, onde estão presentes adoções de outros Utilizadores assim como o estado em que se

encontram. Ao escolher irá para a página da adoção selecionada, onde poderá consultar todo o tipo de informação relevante do animal que está interessado.

Dependências: RF1.1, RF1.6

ID: RF1.6.2

Título: Gestão das adoções.

Descrição: O utilizador poderá visualizar todas suas adoções e realizar inúmeras operações.

Dependências: RF1.1, RF1.6

ID: RF1.6.2.1

Título: Criação de uma adoção.

Descrição: O utilizador poderá criar uma adoção. Ele terá de inserir quatro fotos onde é bem visível o animal que está para adoção e também terá de colocar diversas informações sobre esse animal, como por exemplo tipo de animal, raça, ano de nascimento, breve descrição, cidade da adoção e os respectivos contactos. Caso o utilizador não saiba algum tipo de informação, nomeadamente raça do animal existem campos com a designação "Other" onde o atributo do animal será indefinido.

Dependências: RF1.1, RF1.6

ID: RF1.6.2.2

Título: Remoção de uma adoção.

Descrição: O utilizador poderá remover uma adoção caso pretenda.

Dependências: RF1.1, RF1.6, RF1.6.2.1

ID: RF1.6.2.3

Título: Edição do estado de uma adoção.

Descrição: O utilizador poderá editar o estado de uma adoção. Caso algum outro Utilizador tenha interesse na sua adoção e o tenha contactado pelos contactos presentes nos detalhes, ele poderá alterar o estado de Not Adopted para Adopted, ou vice-versa caso algum imprevisto aconteça. Futuramente ambos os Utilizadores irão combinar como, quando e onde irá ocorrer a Adoção.

Dependências: RF1.1, RF1.6, RF1.6.2.1

Fórum de Perguntas e Respostas

ID: RF1.7

Título: Fórum de Perguntas e Respostas.

Descrição: O utilizador poderá interagir com outros utilizadores com recurso a um fórum de perguntas e respostas.

Dependências: RF1.1

_____ **ID:** RF1.7.1

Título: Perguntas a respostas.

Descrição: O Utilizador terá acesso a uma lista de perguntas, onde estão presentes outras perguntas de outros Utilizadores assim como o estado em que se encontram (Pergunta fechada/aberta). Ao escolher, irá para a página da pergunta selecionada, onde poderá consultar todo o tipo de informação, nomeadamente a pergunta em si e as diversas respostas.

Dependências: RF1.1, RF1.7

ID: RF1.7.2

Título: Gestão das perguntas e respostas.

Descrição: O utilizador poderá visualizar todas as suas perguntas, assim como todas as suas respostas.

Dependências: RF1.1, RF1.7

ID: RF1.7.2.1

Título: Criação de uma pergunta.

Descrição: O utilizador poderá fazer uma pergunta no fórum de forma rápida e eficaz, onde apenas terá de introduzir o título da pergunta, a pergunta em si e caso pretenda anexar alguma imagem auxiliar.

Dependências: RF1.1, RF1.7

ID: RF1.7.2.2

Título: Criação de uma resposta.

Descrição: O utilizador poderá responder a uma pergunta de outro utilizador na página da própria pergunta.

Dependências: RF1.1, RF1.7

ID: RF1.7.2.4

Título: Remoção de uma pergunta.

Descrição: O utilizador poderá remover uma adoção caso pretenda.

Dependências: RF1.1, RF1.7, RF1.7.2.1

ID: RF1.7.2.4

Título: Remoção de uma resposta.

Descrição: O utilizador poderá remover uma resposta caso pretenda.

Dependências: RF1.1, RF1.7, RF1.7.2.2

ID: RF1.7.2.5

Título: Edição do estado de uma pergunta.

Descrição: O utilizador poderá editar o estado de uma pergunta. Caso ele obtenha uma resposta que responda à sua dúvida, ele poderá fechar a pergunta mudando o estado para Closed.

Dependências: RF1.1, RF1.7, RF1.7.2.1

ID: RF1.7.2.5

Título: Respostas verificadas.

Descrição: O utilizador após expor uma dúvida no fórum pode obter uma resposta de um Veterinário registado na aplicação, resposta essa que irá aparecer como comentário destacado e verificado.

Dependências: RF1.1, RF1.7, RF1.7.2.1.

Veterinários perto do Utilizador

ID: RF1.8

Título: Veterinários Perto do Utilizador.

Descrição: O utilizador ver quantos e quais os veterinários que se encontram perto de si num raio de 20km. Para esta funcionalidade resulta é necessário ativar a localização do dispositivo móvel.

Dependências: RF1.1

_____ **ID:** RF1.8.1

Título: Visualização das clínicas mais perto.

Descrição: O Utilizador visualiza todas as clínicas perto dele, seleccioná-las para obter o nome e a respetiva morada.

Dependências: RF1.1, RF1.8

ID: RF1.8.2

Título: Direções para uma clínica.

Descrição: Após clicar numa clínica, e em seguida no botão de direções, irá automaticamente abrir a aplicação Google Maps com as direções para a clínica selecionada.

Dependências: RF1.1, RF1.8, RF1.8.1

3.1.2 Website

Todos os requisitos funcionais da aplicação móvel mencionados no ponto anterior (3.1.1) irão espelhar a maioria dos requisitos funcionais, portanto, nesta seção apenas irão estar os requisitos funcionais que diferem entre eles.

Veterinário

ID: RF1.9

Título: Vista do Veterinário

Descrição: Quando o utilizador é do tipo “vet” consegue fazer operações exclusivas, como visualizar as suas consultas.

Dependências: RF1.1

_____ **ID:** RF1.9.1

Título: Visualização dos detalhes da consulta

Descrição: O veterinário consegue ver todos os detalhes da consulta

Dependências: RF1.1, RF1.9

ID: RF1.9.1.1

Título: Adicionar uma vacina

Descrição: Após consultar os detalhes da consulta o veterinário consegue adicionar uma vacina.

Dependências: RF1.9.1

ID: RF1.9.1.2

Título: Adicionar uma prescrição

Descrição: Após consultar os detalhes da consulta o veterinário consegue adicionar uma prescrição.

Dependências: RF1.9.1

ID: RF1.9.1.2.1

Título: Adicionar uma medicação

Descrição: É possível adicionar medicação conforme o necessário.

Dependências: RF1.9.1.2

ID: RF1.9.1.2.2

Título: Remover uma medicação

Descrição: É possível remover medicação conforme o necessário.

Dependências: RF1.9.1.2

ID: RF1.9.1.3

Título: Cancelar Consulta

Descrição: Após consultar os detalhes da consulta o veterinário consegue cancelar uma consulta.

Dependências: RF1.9.1

Admin

ID: RF1.10

Título: Vista do Veterinário

Descrição: Quando o utilizador é do tipo “admin” consegue fazer operações exclusivas.

Dependências: RF1.1

ID: RF1.10.1

Título: Criar uma clínica

Descrição: Admin consegue criar uma clínica, introduzindo diversas informações, nomeadamente nome, morada, imagem e os respetivos contactos.

Dependências: RF1.10

ID: RF1.10.2

Título: Remover uma clínica

Descrição: Admin consegue remover uma clínica.

Dependências: RF1.10

ID: RF1.10.2

Título: Editar uma Clínica

Descrição: Admin consegue editar as informações da clínica.

Dependências: RF1.10

ID: RF1.10.3

Título: Associar um veterinário a uma clínica

Descrição: Admin consegue associar um veterinário a uma clínica. Tornando esse veterinário num utilizador do tipo “vet”.

Dependências: RF1.10

ID: RF1.10.4

Título: Remover um veterinário a uma clínica

Descrição: Admin consegue remover um veterinário de uma clínica Tornando esse veterinário num utilizador do tipo “normal”.

Dependências: RF1.10

3.2. Requisitos não funcionais

ID: RNF1

Título: Disponibilidade do site.

Descrição: O website deve estar funcional 100% do tempo.

ID: RNF2

Título: Disponibilidade da aplicação.

Descrição: A aplicação móvel deve estar funcional 100% do tempo.

ID: RNF3

Título: Restauração de dados.

Descrição: No caso de falha de Hardware ou outras, que irão comprometer o funcionamento da aplicação, website ou API, deverá ser possível recuperar o sistema (reconfigurar, reiniciar ou repor dados) num intervalo de tempo inferior a 24 horas.

ID: RNF4

Título: Acesso à internet.

Descrição: Qualquer tipo de utilizador necessita de estabelecer conexão com a internet de modo a usufruir das funcionalidades do website e da aplicação.

ID: RNF5

Título: Cópias de segurança.

Descrição: Deverão ser configuradas cópias de segurança regulares da base de Dados e do código (versão funcional encontra-se sempre no github) .

ID: RNF6

Título: Documentação.

Descrição: Deverá ser gerada e desenvolvida toda a documentação necessária a uma rápida e correta interpretação da arquitetura das aplicações.

ID: RNF7

Título: Performance do website.

Descrição: O Website deverá ter um tempo médio de resposta inferior a 15 segundos, em todas as páginas.

ID: RNF8

Título: Performance da aplicação móvel.

Descrição: A aplicação deverá ter um tempo médio de resposta inferior a 10 segundos

4. Planeamento

4.1 Tarefas

Nome	Duração	Início	Fim	Antecessores	Nomes dos Recursos
Elaboração de Proposta	29 dias?	11-02-2021 8:00	11-03-2021 17:00		Miguel Costa;José Rodrigues;Vasco Silva
- Brainstorm e planeamento do projeto	23 dias?	11-02-2021 8:00	05-03-2021 17:00		
- Estrutura e requisitos do projeto	5 dias?	06-03-2021 8:00	10-03-2021 17:00	2	
- Atendimento com os orientadores e confirmação da proposta de trabalho	1 dia?	11-03-2021 8:00	11-03-2021 17:00	3	
Instalação e configuração das ferramentas necessárias	8 dias?	12-03-2021 8:00	15-03-2021 17:00	4	Miguel Costa;José Rodrigues;Vasco Silva
Leitura de documentação e visualização de tutoriais online	11 dias?	16-03-2021 8:00	22-03-2021 17:00	5	Miguel Costa;José Rodrigues;Vasco Silva
Implementação de código e testes	76 dias?	23-03-2021 8:00	06-06-2021 17:00		
- Implementação da API do professor em GO	6 dias?	23-03-2021 8:00	28-03-2021 17:00	6	Miguel Costa;José Rodrigues;Vasco Silva
- Implementação de novos endpoints na API	58 dias?	29-03-2021 8:00	25-05-2021 17:00	8	Miguel Costa;José Rodrigues;Vasco Silva
- Conexão do site à API	4 dias?	30-03-2021 7:00	02-04-2021 17:00	8	José Rodrigues
- Programação do site em Angular	65 dias?	03-04-2021 8:00	06-06-2021 17:00	10	Miguel Costa;José Rodrigues
- Conexão da aplicação à API	4 dias?	20-04-2021 7:00	23-04-2021 17:00	8	Miguel Costa;Vasco Silva
- Programação da aplicação em Flutter	44 dias?	24-04-2021 8:00	06-06-2021 17:00	12	Miguel Costa;Vasco Silva
- Migração da API do Docker para a AWS	4 dias?	27-04-2021 7:00	30-04-2021 17:00	8	José Rodrigues
- Correção de erros	18 dias?	20-05-2021 7:00	06-06-2021 17:00		Miguel Costa;José Rodrigues;Vasco Silva
Apresentação do projeto	1 dia?	07-06-2021 8:00	07-06-2021 17:00	7	Miguel Costa;José Rodrigues;Vasco Silva
Realização do relatório	7 dias?	08-06-2021 8:00	14-06-2021 17:00	16	Miguel Costa;José Rodrigues;Vasco Silva

	Nome	Duração	Início	Fim	Antecessores	Nomes dos Recursos
1	Elaboração de Proposta	29 dias?	11-02-2021 8:00	11-03-2021 17:00		Miguel Costa; José Rodrigues; Vasco Silva
2	Brainstorm e planeamento do projeto	23 dias?	11-02-2021 8:00	05-03-2021 17:00		
3	Estrutura e requisitos do projeto	5 dias?	06-03-2021 8:00	10-03-2021 17:00	2	
4	Atendimento com os orientadores e confirmação da proposta de trabalho	1 dia?	11-03-2021 8:00	11-03-2021 17:00	3	
5	Instalação e configuração das ferramentas necessárias	8 dias?	12-03-2021 8:00	15-03-2021 17:00	4	Miguel Costa; José Rodrigues; Vasco Silva
6	Leitura de documentação e visualização de tutoriais online	11 dias?	16-03-2021 8:00	22-03-2021 17:00	5	Miguel Costa; José Rodrigues; Vasco Silva
7	Implementação de código e testes	76 dias?	23-03-2021 8:00	06-06-2021 17:00		
8	Implementação da API do professor em GO	6 dias?	23-03-2021 8:00	28-03-2021 17:00	6	Miguel Costa; José Rodrigues; Vasco Silva
9	Implementação de novos endpoints na API	58 dias?	29-03-2021 8:00	25-05-2021 17:00	8	Miguel Costa; José Rodrigues; Vasco Silva
10	Conexão do site à API	4 dias?	30-03-2021 7:00	02-04-2021 17:00	8	José Rodrigues
11	Programação do site em Angular	65 dias?	03-04-2021 8:00	06-06-2021 17:00	10	Miguel Costa; José Rodrigues
12	Conexão da aplicação à API	4 dias?	20-04-2021 7:00	23-04-2021 17:00	8	Miguel Costa; Vasco Silva
13	Programação da aplicação em Flutter	44 dias?	24-04-2021 8:00	06-06-2021 17:00	12	Miguel Costa; Vasco Silva
14	Migração da API do Docker para a AWS	4 dias?	27-04-2021 7:00	30-04-2021 17:00	8	José Rodrigues
15	Correção de erros	18 dias?	20-05-2021 7:00	06-06-2021 17:00		Miguel Costa; José Rodrigues; Vasco Silva
16	Apresentação do projeto	1 dia?	07-06-2021 8:00	07-06-2021 17:00	7	Miguel Costa; José Rodrigues; Vasco Silva
17	Realização do relatório	7 dias?	08-06-2021 8:00	14-06-2021 17:00	16	Miguel Costa; José Rodrigues; Vasco Silva

Figura 66. Organização de Tarefas

4.2 Gráfico de Gantt

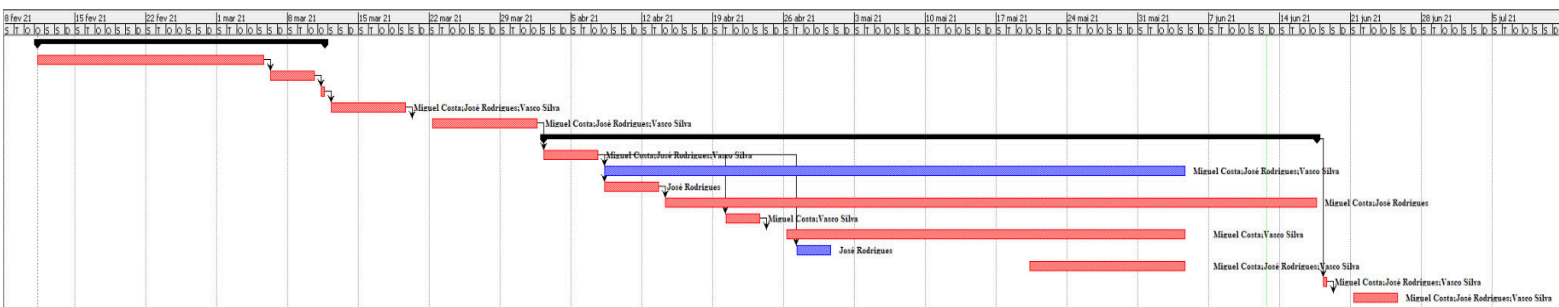


Figura 67. Gráfico Gantt

O trabalho total foi dividido por todos os membros do grupo, sendo assim cada um realizou aproximadamente um terço do projeto em si.

4.3 GIT

Toda a gestão de código foi feita com recurso ao git, criando um repositório onde existiam 3 branches:

- **master** - Código da API.
- **angularmaster** - Código do website
- **fluttermaster** - Código da aplicação

4.4 Tecnologias

Na realização deste projeto foram utilizadas diversas tecnologias, bem como diversas ferramentas. Pode-se dividir este tópico em 2 partes:

- Principais tecnologias:

Neste projeto foram utilizadas 3 tecnologias predominantes, tecnologias essas nunca utilizadas no passado por nenhum dos membros

1. **GoLang**

Linguagem de programação criada pela Google em 2009, que foi utilizada no desenvolvimento da **API** que comunica com o site e **aplicação**

2. **Flutter**

UI open-source criado pelo Google que foi utilizado no desenvolvimento da **aplicação mobile** para android.

3. **Angular**

Framework criada pela Google que foi utilizada para desenvolvimento do **website**.

- Outras tecnologias/ferramentas:

1. **PostgreSQL** - Base de dados utilizada na qual era guardada toda a informação.

2. IDE's:

- a. Visual Studio Code
- b. Android Studio

3. Plataformas de Serviço:

- a. **Docker** - Utilizado numa primeira fase, onde cada elemento do grupo possuía um próprio "host" na sua própria máquina (utilizado aquando o desenvolvimento da API).
- b. **AWS IC2** - Utilizado numa segunda fase, onde foi colocado toda a API num servidor da amazon, para que todos os elementos do grupo tivessem acesso aos mesmos dados (utilizado no desenvolvimento da aplicação/website).

4. API Google Maps

Utilizada para uma funcionalidade da aplicação Mobile.

5. Swagger e Postman

Utilizados na comunicação com a API, auxiliando em diversos testes quando ainda não estavam desenvolvidos a aplicação/website.

6. Livrarias fundamentais:

- a. **GORM** - Biblioteca que faz a comunicação entre a API e a Base de Dados.
- b. **GIN** - Mini framework de alta performance que auxilia nos pedidos efetuados à API.

7. Github

5 Modelização

5.1 Modelos de Base de Dados

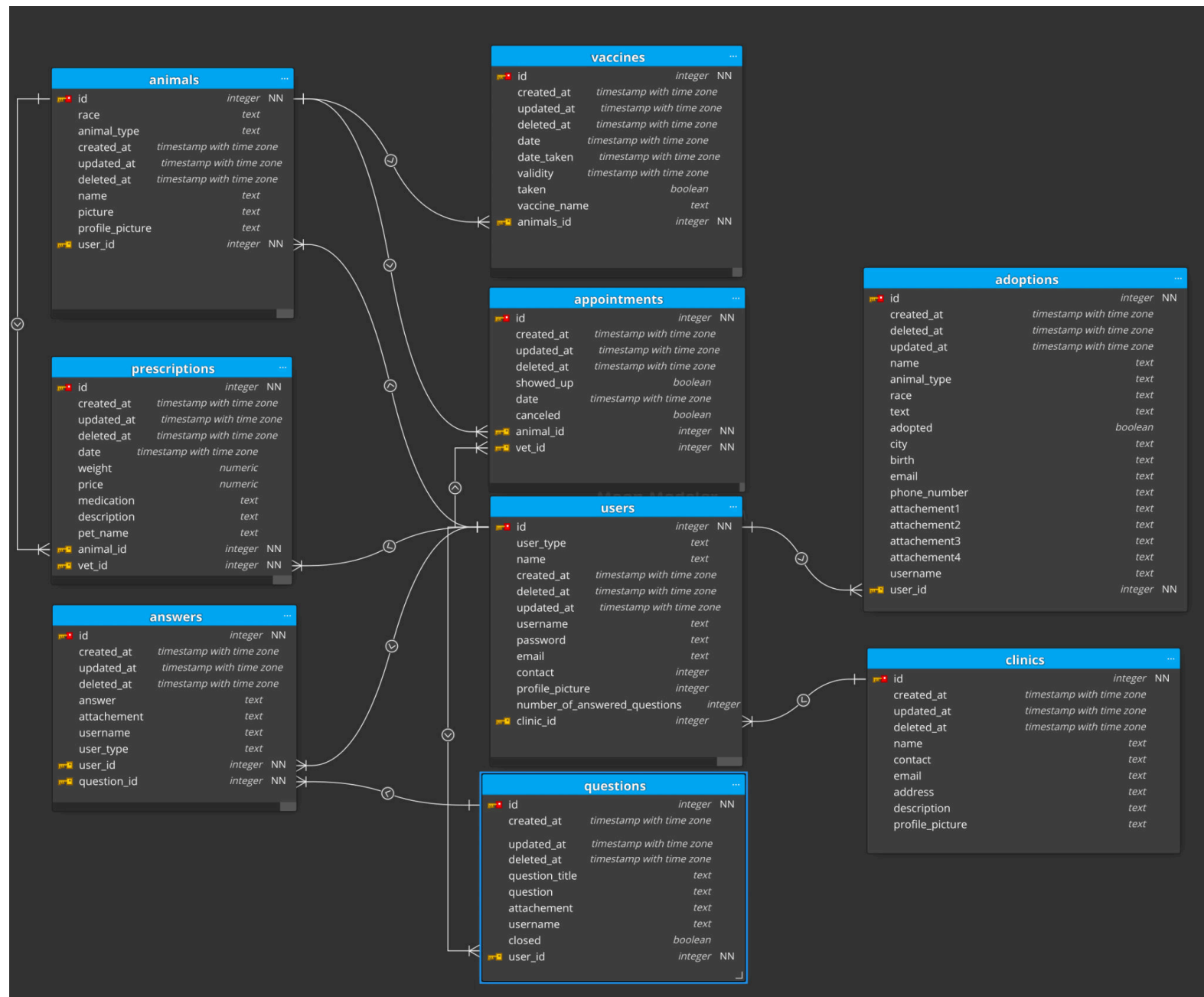


Figura 68. Modelo de Base de Dados

“**adoptions**”, contém toda a informação relativa a uma adoção

- **id**: chave primária incremental
- **created_at**: timestamp com data e hora de criação da adoção
- **updated_at**: timestamp com data e hora de alteração de informações da adoção
- **deleted_at**: timestamp com data e hora de remoção da adoção

- **name:** nome do animal
- **animal_type:** tipo de animal
- **race:** raça do animal
- **user_id:** chave primária do dono do animal
- **text:** descrição do animal
- **adopted:** estado da adoção (adotado, ou por adotar)
- **city:** cidade onde se localiza o animal
- **birth:** data de nascimento do animal
- **email:** email da pessoa responsável pelo animal
- **phone_number:** número de telemóvel da pessoa responsável pelo animal
- **attachement1:** imagem do animal
- **attachement2:** imagem do animal
- **attachement3:** imagem do animal
- **attachement4:** imagem do animal
- **username:** username do utilizador responsável pelo animal

“**animals**”, contém toda a informação pessoal relativa a um animal

- **id:** chave primária incremental
- **created_at:** timestamp com data e hora de criação do animal
- **updated_at:** timestamp com data e hora de alteração de informações do animal
- **deleted_at:** timestamp com data e hora de remoção do animal
- **name:** nome do animal
- **animal_type:** tipo de animal
- **race:** raça do animal
- **profile_picture:** imagem principal
- **picture:** imagem secundária
- **user_id:** chave primária do dono do animal

“**answers**”, armazena as informações relativas às respostas do fórum

- **id:** chave primária incremental
- **created_at:** timestamp com data e hora de criação da resposta
- **updated_at:** timestamp com data e hora de alteração de informações da resposta

- **deleted_at**: timestamp com data e hora de remoção da resposta
- **answer**: texto da resposta
- **user_id**: chave primária do utilizador que escreveu a resposta
- **question_id**: chave primária da pergunta relativa à resposta
- **attachement**: imagem opcional anexada à resposta
- **username**: username do utilizador que escreveu a resposta
- **user_type**: tipo de utilizador (para distinguir veterinários de outros)

“**questions**”, armazena as informações relativas às questões do fórum

- **id**: chave primária incremental
- **created_at**: timestamp com data e hora de criação da questão
- **updated_at**: timestamp com data e hora de alteração de informações da questão
- **deleted_at**: timestamp com data e hora de remoção da questão
- **user_id**: chave primária do utilizador que escreveu a questão
- **question_title**: título da questão
- **question**: texto da questão
- **attachement**: imagem opcional anexada à questão
- **closed**: caso questão resolvida, user consegue fechar questão
- **answers**: número de respostas na questão

“**appointments**”, contém toda a informação relativa a uma consulta

- **id**: chave primária incremental
- **created_at**: timestamp com data e hora de criação da consulta
- **updated_at**: timestamp com data e hora de alteração de informações da consulta
- **deleted_at**: timestamp com data e hora de remoção da consulta
- **showed_up**: presença do animal
- **animal_id**: chave primária do animal a ser consultado
- **date**: timestamp com data e hora da consulta
- **vet_id**: chave primária do veterinário encarregado da consulta
- **canceled**: estado da consulta (se foi cancelada ou não)

“**clinics**”, contém toda a informação relativa a uma clínica

- **id:** chave primária incremental
- **created_at:** timestamp com data e hora de criação da clínica
- **updated_at:** timestamp com data e hora de alteração de informações da clínica
- **deleted_at:** timestamp com data e hora de remoção da clínica
- **name:** nome da clínica
- **contact:** número de telefone da clínica
- **email:** email da clínica
- **address:** endereço da clínica
- **latitude:** latitude geográfica da clínica
- **longitude:** longitude geográfica da clínica
- **description:** descrição da clínica
- **profile_picture:** imagem de perfil da clínica

“**prescriptions**”, armazena as informações de uma prescrição

- **id:** chave primária incremental
- **created_at:** timestamp com data e hora de criação da prescrição
- **updated_at:** timestamp com data e hora de alteração de informações da prescrição
- **deleted_at:** timestamp com data e hora de remoção da prescrição
- **weight:** peso do animal consultado
- **price:** preço da consulta
- **medication:** string com informações acerca dos medicamentos prescritos ao animal, atua como uma string protocolo onde cada campo é uma unidade de informação
- **date:** timestamp com data e hora da prescrição
- **animal_id:** chave primária do animal consultado
- **description:** resumo da consulta
- **pet_name:** nome do animal consultado
- **vet_id:** chave primária do veterinário encarregado da consulta

“vaccines”, armazena as informações de uma vacina

- **id**: chave primária incremental
- **created_at**: timestamp com data e hora de criação da vacina
- **updated_at**: timestamp com data e hora de alteração de informações da vacina
- **deleted_at**: timestamp com data e hora de remoção da vacina
- **animal_id**: chave primária do animal que tomou vacina
- **vaccine_name**: nome da vacina
- **date_taken**: data de toma da vacina
- **validity**: validade da vacina

“users”, armazena as informações de um user

- **id**: chave primária incremental
- **created_at**: timestamp com data e hora de criação do user
- **updated_at**: timestamp com data e hora de alteração de informações do user
- **deleted_at**: timestamp com data e hora de remoção do user
- **username**: username unico do user
- **password**: password do user, de notar que password está hashed
- **email**: email do user
- **name**: nome do user
- **user_type**: tipo de user (vet,norma,admin)
- **contact**: contacto do user
- **clinic_id**: id da clínica onde user trabalha, se for veterinário
- **profile_picture**: imagem de perfil user
- **number_of_answered_questions**: número de perguntas que user respondeu

5.2 Casos de Uso

5.2.1 Administrador e Veterinário

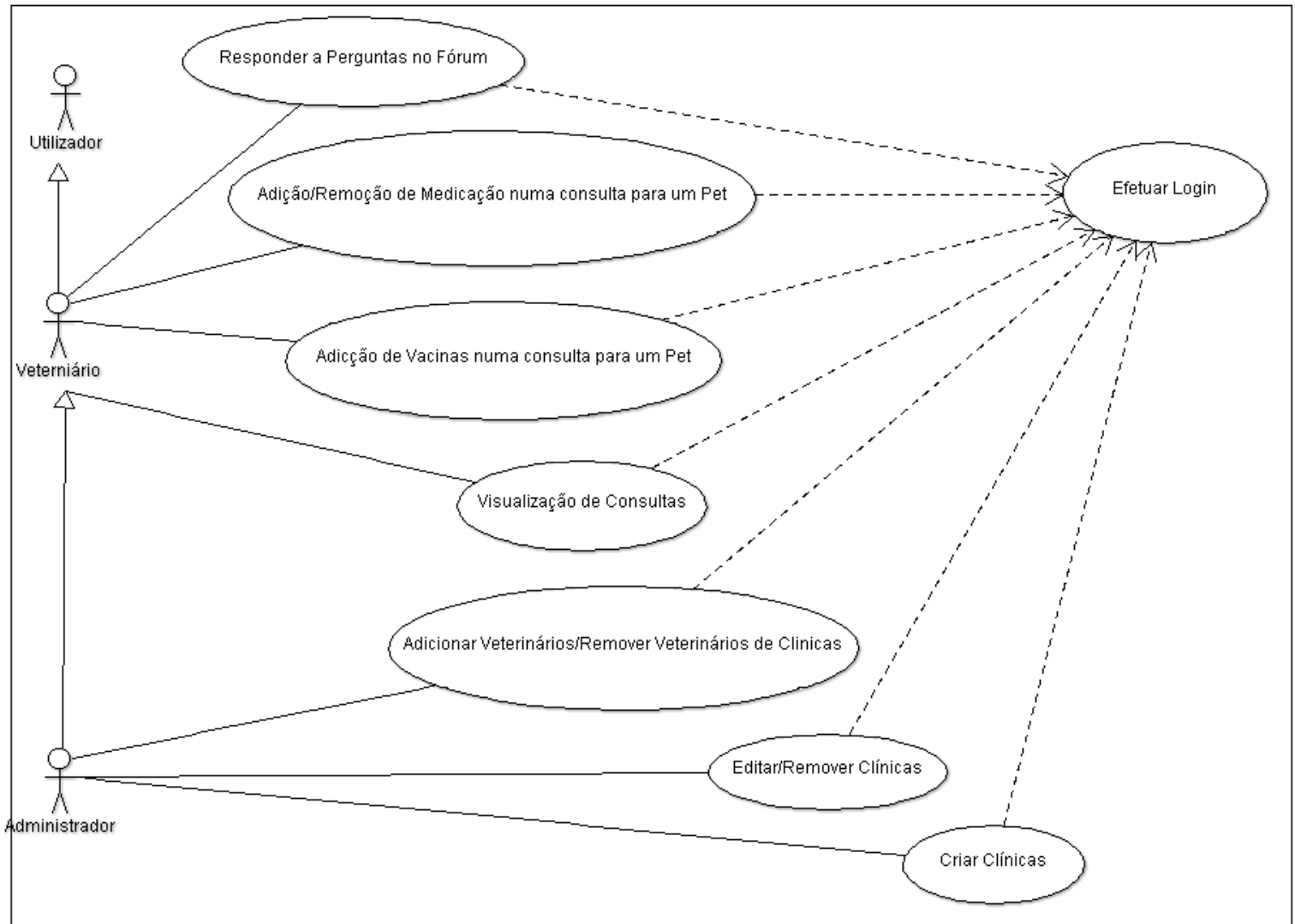


Figura 69 . Casos de Uso para o Veterinário e para o Administrador

5.2.2 Utilizador

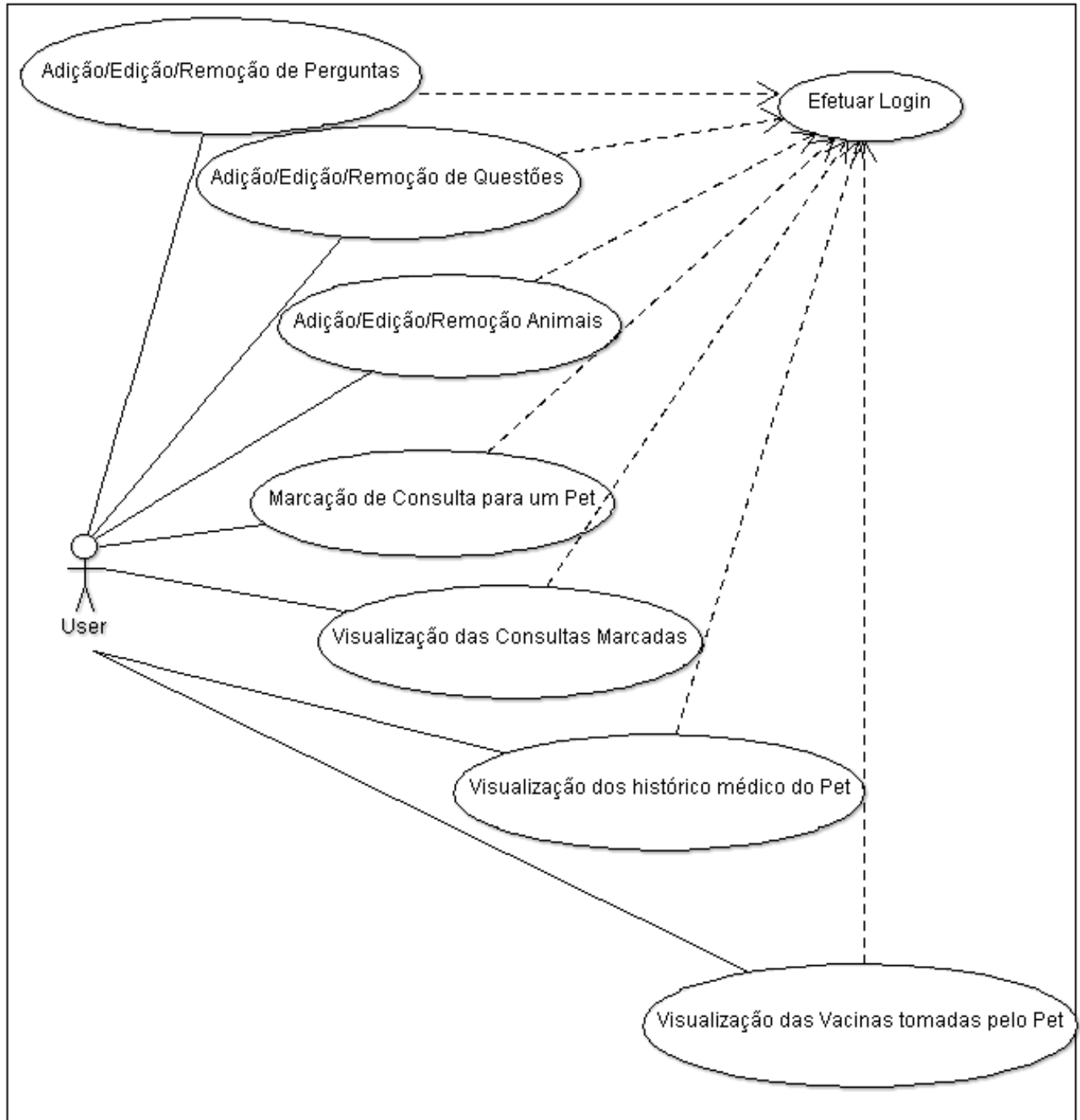


Figura 70. Casos de Uso para o utilizador

5.3 Diagrama de Classes

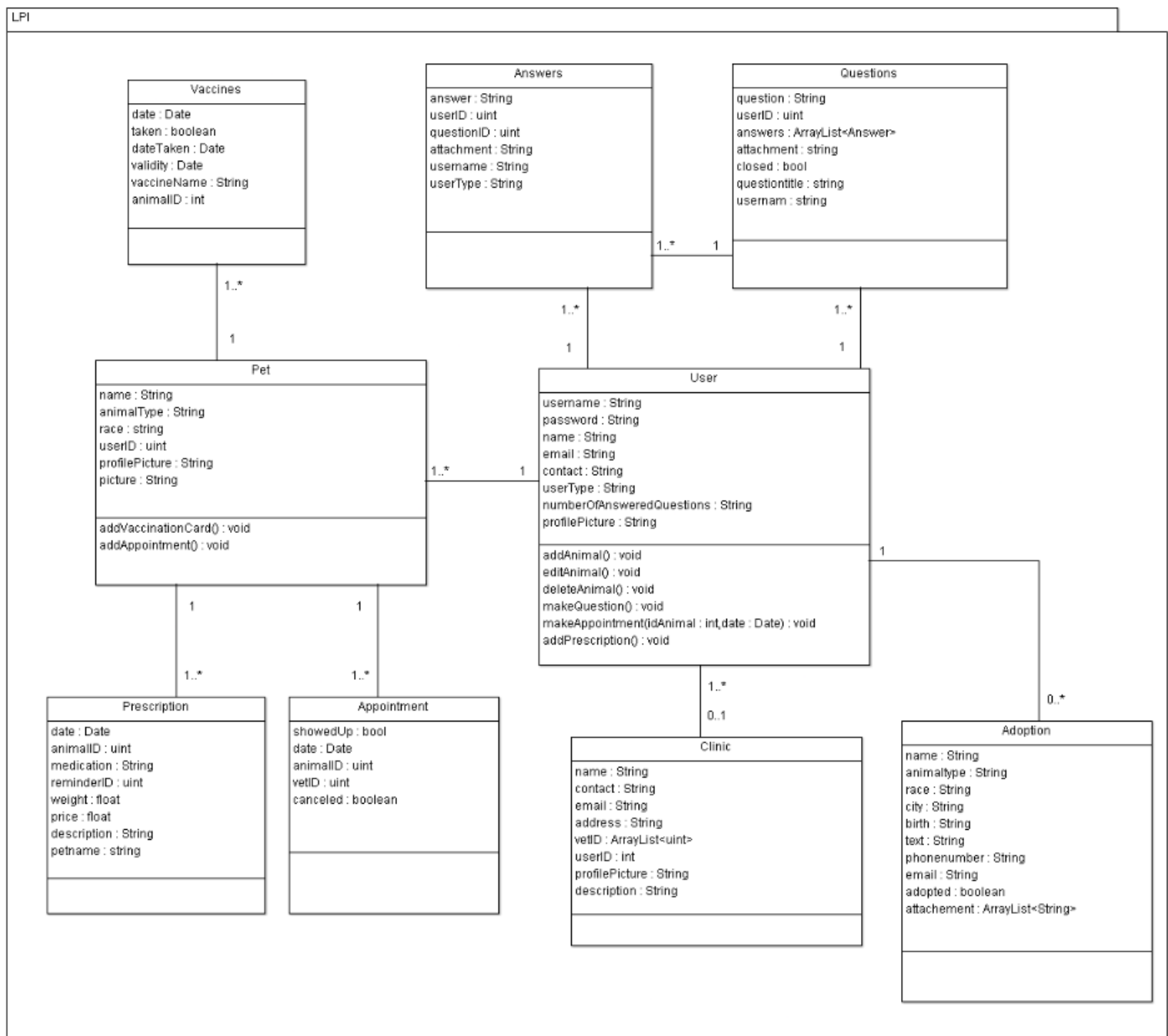


Figura 71. Diagrama de Classes relativo à aplicação

A classe 'User' é onde são feitas todas as operações possíveis pelo Utilizador. Como já foi dito anteriormente existem 3 tipos de utilizadores diferentes que são distinguidos pelo atributo 'userType' na classe 'User'. Um User normal pode ter

vários 'Pet', que por sua vez podem ter várias Vacinas, Prescrições e Consultas. Este pode também adicionar adopções à aplicação e também perguntar e responder a perguntas feitas por outros utilizadores, tendo assim uma ligação para as Answers e Questions. Um User veterinário tem operações que o User normal não possui, e o mesmo para o User admin.

A ligação entre o User e a Clínica é apenas utilizada se o 'userType' for vet, pois o veterinário trabalha numa Clínica Veterinária.

6. Demos e testes

6.1 Aplicação móvel

Para uma forma de avaliação mais interativa, foi criado um QR code que direciona para um link de download da aplicação, pois, devido ao tempo reduzido nas semanas finais, não foi possível colocar a aplicação na Play Store dos dispositivos Android, devido ao método de aprovação, pois é um processo lento que pode demorar até mesmo semanas.

Segue um QR code onde apenas basta fazer scan com o seu dispositivo móvel, fazer download do arquivo **app-release.apk** e instalar no seu dispositivo.



Figura 72. Código QR para instalar a aplicação

6.1.1 Dados teste

Relativamente a dados teste, irá ser fornecido a conta usada neste trabalho para demonstração de funcionalidades.

Email:miguel@email.com

Palavra-passe:123456

6.1.2 Demos

Demo App: https://www.youtube.com/watch?v=6ZvB-hu_640&t=113s

Demo Website User UI : <https://www.youtube.com/watch?v=kx15xzsGqMQ>

Demo Website Vet UI: <https://www.youtube.com/watch?v=6vjoQ5SnYhg>

Demo Website Admin UI: https://www.youtube.com/watch?v=X0TCx_plro

7.Conclusão

Em suma, foi uma experiência bastante enriquecedora, onde foi possível potencializar os conhecimentos adquiridos ao longo da licenciatura. Com a realização deste projeto cada membro do grupo conseguiu melhorar as suas capacidades como engenheiro informático, adicionando conhecimento de novas ferramentas do mundo moderno da programação para o seu futuro.

Foi também uma experiência bastante desafiadora, por se tornar num projeto complexo e demorado, onde no início nenhum dos membros possuía conhecimentos das três principais tecnologias utilizadas, nomeadamente GoLang, Flutter e Angular e em cerca de três meses, conseguimos com sucesso, criar uma aplicação de telemóvel e um website, estando ambos interligados por uma API utilizando essas três tecnologias .

Todas as ideias e requisitos iniciais foram implementados, à excepção de um. O único requisito que não cumprimos foram as notificações para os medicamentos, que iriam atuar como uma espécie de lembrete para que um dado Utilizador não se esquecesse de dar a medicação ao animal.

Surgiram também ideias que gostaríamos de aplicar neste projeto futuramente:

- Implementar os lembretes para os medicamentos.
- Implementar Forecast - identificando anomalias em animais de raça características semelhantes (e.g. variações de peso).
- Implementar Integração de documentos em Papel c/ OCR (e.g., faturas / recibos do veterinário adicionados com base numa fotografia).

Por fim, gostaríamos de agradecer aos nossos orientadores, o professor Christophe Soares e o professor Alessandro Moreira por nos terem ajudado ao longo deste percurso.

8.Logbook

Planeamento)

Antes 11/03 - Pensar em ideias para projeto

11/03 - Designação do projeto

12/03 - Estrutura e requisitos do projeto

13/03 até 15/3 - Instalações necessárias para começar a trabalhar com Go, Angular e Flutter

16/03 até 20/03 - Leitura de documentação e visualização de tutoriais no youtube

API)

26/3 - 28/3 = Implementação da api do professor e realização de algumas modificações

31/03 - Implementação de novos endpoints, relativos a registo, login, consultas, animais e lembretes

5/04 até 18/04 - Implementação de novos endpoints, acerca de clínicas, respostas, adoções, animais, prescrições e consultas. Alteração de middlewares e no token JWT

19/04 até 21/04 - Implementação de novos endpoints relacionados com clínicas veterinárias

23/04 - Correção de erros nos endpoints relativos a consultas

25/04 - Implementação de novos endpoints e alterações a outros existentes relacionados com os fóruns e clínicas

26/04 até 4/05 - Criação de novos endpoints relacionados a pesquisas de informações de animais, users, consultas, e outros endpoints relacionados com autenticação e alteração de password

5/05 até 13/05 - Implementação de novos endpoints relativos a clínicas, consultas, prescrições e respostas de fóruns

15/05 até 25/05 - Alteração de endpoints existentes e implementação de outros novos, relacionados a perguntas de fóruns, animais, token JWT, envio de email e autenticação

Angular)

3/04 - Inicialização do website, foi desenvolvido o formulário de login e o de registo.

4/04 até 6/04 - Ligação entre website e API.

7/04 até 10/04 - Construção da aba “Pets”, adição, remover, editar e listagem

11/04 até 15/04 - Construção da aba “Clinics” e Clinic Profile, com todas as operações possíveis.

15/04 até 20/04 - Construção da aba “Adoções” com todas as operações

20/04 até 27/04 - Implementação da marcação de consultas

27/04 até 3/05 - Implementação do UI do veterinário com o calendário e vista de detalhada de consultas

4/05 - Implementação do UI do Admin

5/05 - Criação do Perfil do Utilizador

6/05 - Implementação do reset password

7/05 - Permitir alterar fotografias dos animais e do utilizador

8/05 - Melhoramento da marcação de consulta e resolução de problemas com o JWT token.

9/05 até 13/05 - Construção da aba “Fórum” com todas as operações de fazer pergunta, responder, adicionar imagem

13/05 até 20/05 - Correção de Bugs

21/05 até 23/05 - Implementação da adição de vacinas, adição de prescrições e cancelamento da consulta do lado do veterinário

24/05 - 25/04 - Adição da visualização de vacinas e prescrições no perfil do animal

26/05 - Construção do gráfico de variação do peso do animal.

30/05 até 6/06 - Correção de Bugs

Flutter)

30/04 - Atendimento com o professor Christophe, instalação da API na Amazon AWS

1/05 - Fazer a ligação entre a aplicação e a API e construir o Login/Registo

2/05 - Realização da decodificação do JWT e início da construção da HomePage e da NavBar

3/05 - Construção da aba “Pets”, apenas listagem de valores estáticos

4/05 - Operações de adicionar, remover e editar animal e listagem de valores dinâmicos

6/05 - Construção da aba adoções e tentativas de designs.

7/05 - Operações de adicionar, remover e editar adoções e listagem de forma estática.

9/05 - Melhoramento no design da aplicação (pop ups e embelezamentos)

10/05 - Aba de display clínicas e perfil clínica feitos de forma estática.

11/05 - Aba de display clínicas dinâmica

12/05 - Implementação de base64 e troca de todas as fotos estáticas para imagens dinamicamente adicionadas.

12/05 - Design estático para a marcação de consultas

13/05 - Correção de alguns bugs

15/05 até 18/05 - Construção da aba Fórum, assim como as operações de adicionar/remover perguntas/respostas. Correção de erros e alterações na aplicação em geral.

20/05 até 28/05 - Construção das consultas com um stepper e da implementação da escolha de horário por slots disponíveis. Também foi implementado as packages de calendários

28/05 até 30/05 - Correção de bugs, aprimoração do design, e algumas implementações como fotos no fórum, e novas funcionalidades.

1/05 - Adição da API do GoogleMaps

2/05 até 4/05 - Todo tipo de validações

5/05 - Correção de erros e bugs