



UNIVERSIDADE  
LUSÓFONA

# VetOn

## Trabalho Final de curso

Relatório Final

Nome do Aluno: Rodrigo Simões a22001628

Nome do Aluno: Ruben Silva a22003218

Nome dos Orientadores: Professor João Pedro Leal Abalada de Matos Carvalho

João Pedro Calado Barradas Branco Pavia

Trabalho Final de Curso | LEI | 30/06/23

[www.lusofona.pt](http://www.lusofona.pt)

## **Direitos de cópia**

(VetOn), Copyright de (Rodrigo Simões e Ruben Silva), ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

## Resumo

Este documento foi realizado no âmbito do trabalho final de curso dos autores, que tem como título “VetOn”. Este projeto tem como objetivo o desenvolvimento de uma aplicação *mobile* e *web* que permite agendar, de uma forma organizada e simples, qualquer serviço prestado pelo Hospital Veterinário da Universidade Lusófona.

Os animais de estimação, mais que uma companhia, são destaque em muitos lares portugueses, fazendo parte das famílias que habitam nos mesmos. Com o começo da pandemia de Covid-19, a adoção destes “amigos de 4 patas” aumentou em 32%, de acordo com um estudo feito pela plataforma Fixando.

Com isto, e sabendo que é importante garantir a saúde dos animais de estimação de qualquer indivíduo, a nossa intenção é desenvolver uma aplicação que seja acessível e interativa a todo e qualquer tipo de cliente que usufrua dos serviços veterinários do Hospital Veterinário da Universidade Lusófona. A nossa aplicação *mobile* e *web* deverá colmatar dificuldades identificadas no que toca ao agendamento de consultas e divulgação de serviços prestados pelo Hospital Veterinário da Universidade Lusófona, fazendo com que os clientes tenham toda a informação necessária, seja referente ao seu animal de estimação seja aos próprios serviços mencionados anteriormente, à distância de um clique no seu smartphone.

Por fim, a pandemia de Covid-19 não trouxe somente dados de que fiquemos orgulhosos, visto que o abandono dos animais continua a ser um problema preocupante que põe em causa o bem-estar animal. Sabendo que o Hospital Veterinário da Universidade Lusófona apresenta vontade de combater essa mesma causa, pretendemos implementar na aplicação uma parte de cariz solidário onde será possível adotar animais que se encontrem nas instalações à espera de uma família para os acolher.

**Palavras-chave:** Aplicação *Mobile*, Aplicação *Web*, Veterinário, Consultas, Hospital Veterinário da Universidade Lusófona, Animais, JavaScript, Node.js, MongoDB, Express

## Abstract

This report was written as part of the final course work of the authors, which is entitled “VetOn”. This project aims to develop a mobile and web application that allows scheduling, in an organized and simple way of any service provided by the Hospital Veterinário da Universidade Lusófona.

Pets, more than a company, are featured in many Portuguese homes, being part of the families that live in them. With the onset of the Covid-19 pandemic, the adoption of these “four-legged friends” increased 32%, according to a study conducted by the Fixando’s platform.

With this in mind and knowing that it is important to ensure the health of any individual's pets, our intention is to develop an application that is accessible and interactive to any and all types of customers who benefit from the veterinary services of the Hospital Veterinário da Universidade Lusófona. Our mobile and web application should overcome difficulties identified when it comes to scheduling appointments and publicizing the services provided by the Hospital Veterinário da Universidade Lusófona, ensuring that customers have all the necessary information, whether referring to their pet or their own previously mentioned services, just a click away on your smartphone.

Finally, the Covid-19 pandemic has not only brought data that we are proud of, given that the abandonment of animals continues to be a worrying problem that jeopardizes animal welfare. Knowing that the Hospital Veterinário da Universidade Lusófona is willing to fight this same cause, we intend to implement a solidarity part in the application where it will be possible to adopt animals that are in the premises waiting for a family to welcome them.

**Keywords:** Mobile application, Web application, Vet, Appointments, Hospital Veterinário da Universidade Lusófona, Animals, JavaScript, Node.js, MongoDB, Express

# Índice

Resumo .....	iii
Abstract.....	v
Lista de Figuras .....	3
Lista de Tabelas .....	4
1 Identificação do Problema.....	5
2 Viabilidade e Pertinência .....	6
3 Benchmarking .....	7
4 Engenharia.....	10
4.1 Levantamento e Análise de Requisitos .....	10
4.2 Diagramas de Casos de Uso.....	13
4.3 Diagramas de Atividades .....	13
4.4 Modelos relevantes .....	15
4.5 Estrutura.....	16
4.6 Mockups e Story Board .....	17
5 Solução Proposta.....	21
5.1 Introdução .....	21
5.2 Arquitetura .....	21
5.3 Tecnologias e Ferramentas utilizadas .....	22
5.4 Implementação.....	25
5.5 Abrangência.....	26
6 Plano de testes e validação .....	28
7 Método e Planeamento .....	32
8 Conclusão e trabalhos futuros.....	33
Bibliografia.....	34
Anexo 1 – Questionário.....	36
Anexo 2 – Resultados do questionário .....	39
Anexo 3 – Guião de tarefas .....	45
Anexo 4 – Resultados dos testes .....	49
Progresso de Trabalho .....	53
Update 26 janeiro:.....	56

Update 23 abril: .....	58
Retificação dos Requisitos.....	59
Atualização de Diagramas .....	60
Atualização do Tópico referente à Arquitetura.....	60
Atualização do Tópico referente as Tecnologias e Ferramentas utilizadas .....	61
Atualização do Tópico Implementação .....	61
Atualização do Tópico Abrangência.....	61
Glossário.....	68

## Lista de Figuras

Figura 1 – Secção do Hospital Veterinário da Universidade Lusófona	5
Figura 2 – Solução “ZU”	7
Figura 3 – Solução “Vetecare”	8
Figura 4 – Solução “Petappoint”	8
Figura 5 – Solução “Petable”	9
Figura 6 – Diagrama de Casos de Uso	13
Figura 7 – Diagrama de Atividades (Marcação de consulta)	13
Figura 8 - Diagrama de Atividades (Desmarcação de consulta)	14
Figura 9 - Diagrama de Atividades (Registo)	14
Figura 10 - Diagrama de Atividades (Login)	15
Figura 11 - Modelo Entidade-Relação (Conceptual)	15
Figura 12 – Diagrama de classes	16
Figura 13 – Mapa aplicacional	16
Figura 14 – <i>Landing Page</i>	17
Figura 15 - <i>NavBar</i>	18
Figura 16 – Secção “Serviços”	18
Figura 17 – Secção “Médicos Afiliados”	19
Figura 18 – Secção “Contactos”	19
Figura 19 – Login / Registo	20
Figura 20 – Arquitetura da aplicação	22
Figura 21 – Planeamento em formato GANTT	32
Figura 22 – Marcação e desmarcação de consultas	56
Figura 23 – Formulários de autenticação	57
Figura 24 – Código encriptação de password	57
Figura 25 – Código para a conexão à base de dados MongoDB	58
Figura 26 – Mongo Atlas	58
Figura 27 – Atualização do Mapa Aplicacional	60
Figura 28 – Atualização do Diagrama Arquitetura	61
Figura 29 – Live-Chat	62
Figura 30 – Marcação e desmarcação de Consultas	62
Figura 31 - Atualização área pessoal	63
Figura 32 – Visualização das Consultas	63
Figura 33 – Complexidade das Passwords	64
Figura 34 – Email para redefinir a password	65
Figura 35 – Alterações na Landing Page	65
Figura 36 – Página do admin (adição médicos)	66
Figura 37 – Página admin (adição de serviços)	66
Figura 38 – Página support (Live-Chat)	67

## Lista de Tabelas

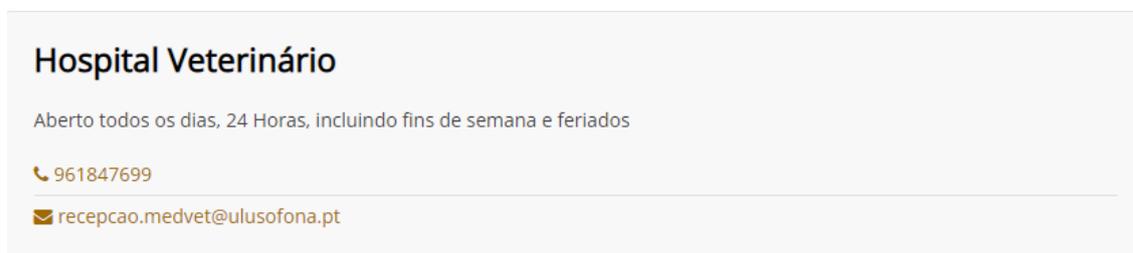
Tabela 1 - Benchmarking	9
Tabela 2 - Tabela de testes	30
Tabela 3 - Tabela de Requisitos Não Funcionais	53
Tabela 4 - Tabela de Requisitos Funcionais	55
Tabela 5 – Tabela de Requisitos Técnicos	55

# 1 Identificação do Problema

Verificámos que o Hospital Veterinário da Universidade Lusófona possui alguma carência em aspetos de logística e gerenciamento nas suas consultas médicas bem como na divulgação de informação e de serviços prestados pelos mesmos.

Depois de uma análise à gestão de consultas realizada por parte do Hospital Veterinário da Universidade Lusófona, verificamos que esta é realizada por meio telefónico e por e-mail, consideramos que nos dias de hoje é extremamente necessário e fiável o desenvolvimento de uma aplicação que agregue tanto essa vertente de marcação e respetiva gestão de consultas bem como o total acompanhamento dos animais de estimação de cada indivíduo, tornando assim todo o processo mais simples e rápido para os utilizadores.

Ao fazer uma pesquisa básica no Google sobre o Hospital Veterinário da Universidade Lusófona, encontramos somente uma página referente aos contactos da Faculdade de Medicina Veterinária da Universidade Lusófona, onde se encontrava uma secção intitulada de “Hospital Veterinário”, como se pode observar na Figura 1 abaixo apresentada.



**Figura 1 – Secção do Hospital Veterinário da Universidade Lusófona**

No nosso ponto de vista, isto é um problema de elevada preocupação, visto que se não houver contacto prévio com o Hospital Veterinário da Universidade Lusófona, não conseguimos obter nenhuma informação útil no que toca a serviços prestados pelo mesmo, bem como qualquer tipo de informação sobre o agendamento de consultas.

## 2 Viabilidade e Pertinência

Este projeto tem as capacidades necessárias para garantir a viabilidade do seu desenvolvimento para além do âmbito deste trabalho académico, visto que pode ser adaptado e implementado para outras entidades (clínicas veterinárias).

Numa primeira fase o projeto é exclusivamente direcionado ao Hospital Veterinário da Universidade Lusófona, de forma a gerir as suas consultas e a dar a conhecer aos utilizadores todos os serviços disponíveis. Contudo, a metodologia aplicada no desenvolvimento do projeto permite que este seja facilmente adaptado uma vez que o *design* e funcionalidades deste não se encontram única e exclusivamente centrados no Hospital Veterinário da Universidade Lusófona.

No desenvolvimento do projeto teremos em consideração todos as aplicações e *websites* já existentes no mercado com o mesmo propósito da nossa solução, visto que estas poderão afetar a viabilidade da mesma.

Posto isto, após uma breve pesquisa a potenciais concorrentes à nossa aplicação, identificámos algumas falhas que pretendemos colmatar tais como: o envio de tickets com imagens/vídeos para possível análise de um médico disponível, o uso de notificações para a marcação de consultas.

Realizámos um questionário online que se encontra presente no Anexo 1 (implementado através da plataforma *Google Forms*), de forma a suportar a decisão na inclusão de funcionalidades e/ou aspetos importantes no desenvolvimento do *website*/aplicação bem como adição de novas funcionalidades sugeridas ao analisar os resultados do questionário que se encontram presentes no Anexo 2.

### 3 Benchmarking

Após uma breve pesquisa a potenciais concorrentes à nossa aplicação verificámos, que atualmente existem algumas aplicações com o mesmo intuito da nossa. Posto isto procurámos analisar os aspetos mais diferenciadores de alguns aplicativos e refletir melhorias na nossa solução.

Vamos tomar como exemplo soluções como a “Zu” [1], a “VetCare” [2], a “Petappoint” [3] e “Petable” [4].

Zu é uma aplicação *mobile* desenvolvida pela Sonae para as suas lojas Zu com o intuito de colaborar na gestão / agendamento de consultas, banhos e tosquiagens de animais de estimação.

A solução oferecida é direcionada apenas às lojas Zu, esta oferece aos seus utilizadores: o acesso ao perfil do animal, histórico clínico, marcação de serviços, consulta das lojas ZU e da agenda.

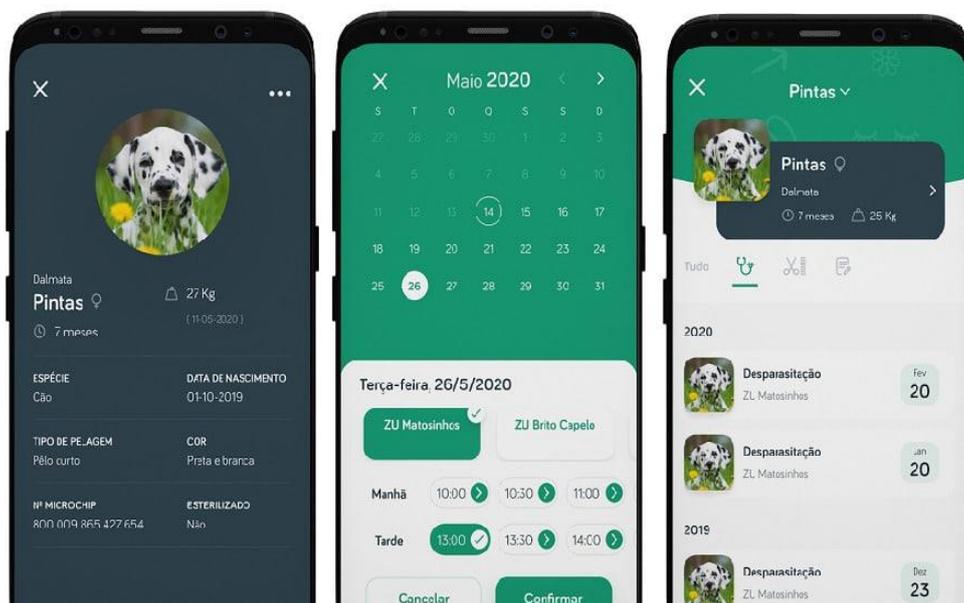
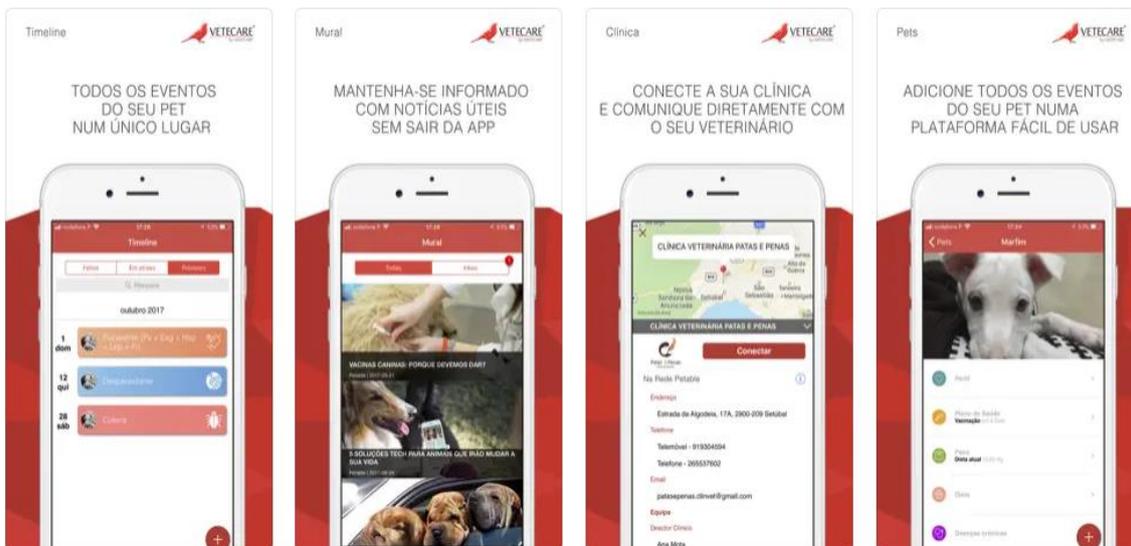


Figura 2 – Solução “ZU”

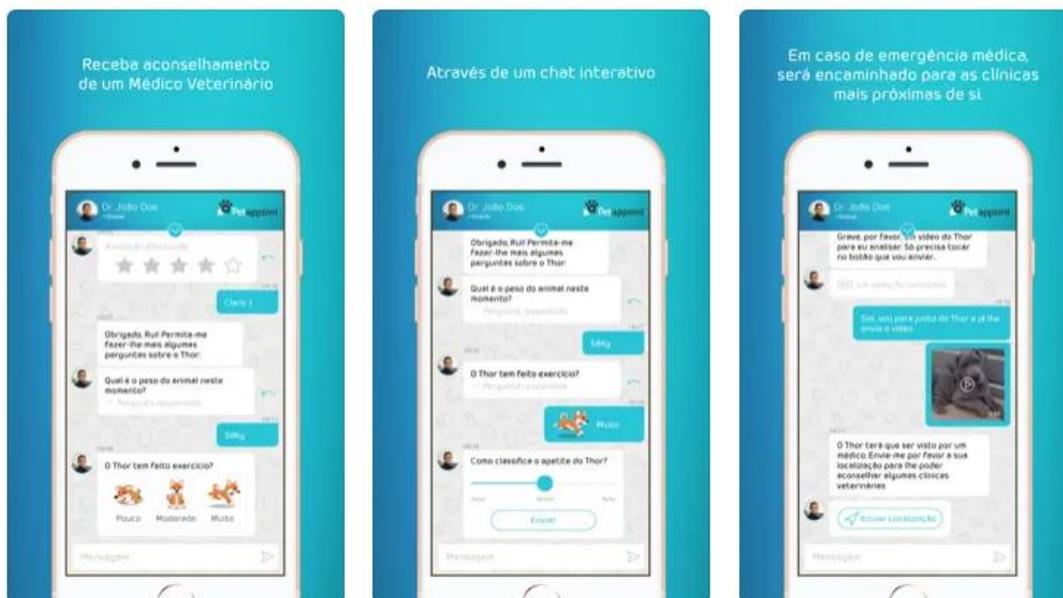
*Vetecare* é uma aplicação para ajudar os donos de animais de estimação a manter a saúde e o bem-estar dos mesmos em dia.

A solução oferece aos seus utilizadores o acesso a: uma área de perfil, lembretes para todos os eventos, uma *timeline* da vida dos seus *pets*, a possibilidade de se conectar à sua clínica veterinária através da aplicação e um mural de notícias.



**Figura 3 – Solução “Vetecare”**

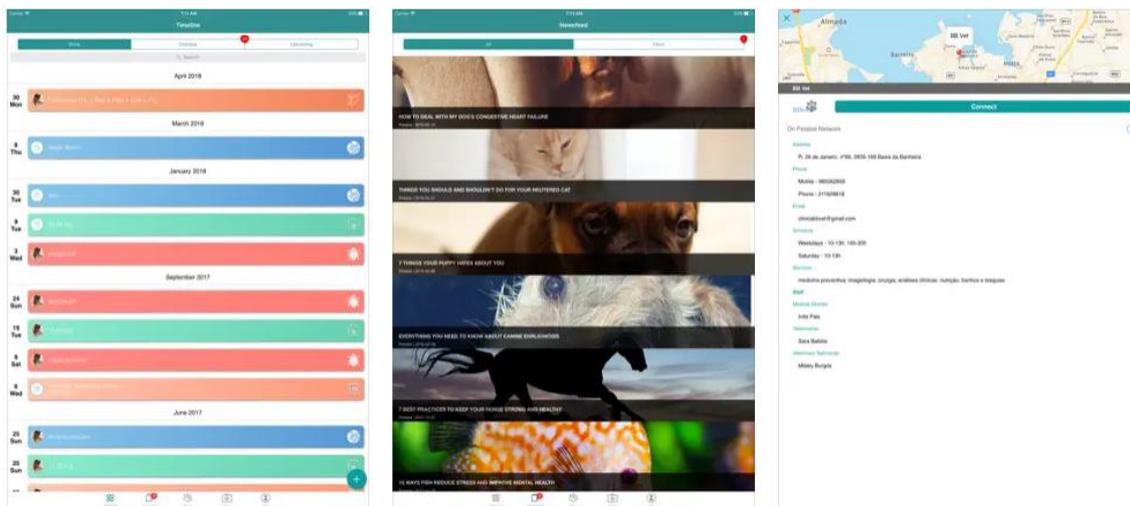
*Petappoint* é uma aplicação *mobile* que permite a comunicação com um médico veterinário através de um *chat* em tempo real, onde é possível enviar vídeos, fotografias e outro tipo de informações tornando a conversa ainda mais interativa.



**Figura 4 – Solução “Petappoint”**

*Petable* é uma aplicação desenvolvida por uma equipa de veterinários para ajudar os donos de animais de estimação a manter a saúde e o bem-estar dos seus animais em dia.

A solução oferece aos seus utilizadores o acesso a: uma área de perfil, lembretes, uma *timeline*, a possibilidade de se conectar à sua clínica veterinária através da aplicação e um mural de notícias.



**Figura 5 – Solução “Petable”**

É necessário ter em consideração que existem mais soluções no mercado, mas na nossa pesquisa consideramos apenas as que tiveram updates recentes e que não apresentam custos adicionais.

Com a solução apresentada por nós, tentamos responder às necessidades do Hospital Veterinário da Universidade Lusófona, dando uso do que de melhor estes tipos de aplicações oferecem, permitindo assim com que o produto final seja uma aplicação simples, mas completa.

	Gestao de Consultas	Agenda	Historico do animal	Notificacoes (SMS e Email)	Mural de noticias	Prestacao de servicos	Live Chat	Perfil do animal
Zu	✓	✓	✓	✓	✗	✓	✗	✓
Vetcare	✗	✓	✓	✗	✓	✗	✗	✓
Petappoint	✗	✗	✗	✓	✗	✗	✓	✗
Petable	✗	✗	✓	✓	✓	✗	✗	✓
VetOn	✓	✓	✓	✓	✗	✓	✓	✓

**Tabela 1 - Benchmarking**

## 4 Engenharia

### 4.1 Levantamento e Análise de Requisitos

Para uma criação bem-sucedida de uma aplicação, é necessário realizar o levantamento e a análise de requisitos inerentes ao dimensionamento da mesma. Estes estabelecem as bases para a inicialização do projeto e ainda especificam os objetivos e parâmetros do mesmo.

Sem requisitos bem definidos, a equipa de desenvolvimento poderá ter problemas em perceber o que é necessário cumprir, podendo assim criar uma aplicação que fique aquém dos objetivos idealizados, tendo como exemplo os seguintes problemas:

1. Difícil usabilidade;
2. Erros na execução;
3. Desorganização;
4. Requisitos desnecessários;
5. Falta de funções importantes;
6. Perda de tempo no desenvolvimento.

Tal facto, poderá gerar frustração por parte do cliente e dos desenvolvedores, prejudicando o desenvolvimento do projeto (atrasando o prazo de entrega e/ou ocorrência de possíveis mudanças no orçamento). Sendo assim, quanto mais coesa e clara for a especificação de requisitos de software, maior será a possibilidade de o projeto ser bem sucedido.

Enumera-se de seguida, os Requisitos Não Funcionais (**RNF**), Requisitos Funcionais (**RF**) e Requisitos Técnicos (**RT**) que cumpram tanto os objetivos que estipulamos como as possíveis necessidades provenientes dos utilizadores, assim como a escala 'have' e o esforço esperado para cada requisito:

#### Definição das escalas:

**Escala 'have':** nice to have, could have, should have, must have

**Esforço esperado:** XS, S, M, L, XL

- **Requisitos Não Funcionais (RNF)**

**RNF1.** A aplicação deve ser capaz de lidar com um número significativo de utilizadores e de tarefas sem sofrer quaisquer problemas de lentidão; (**must have**)

**RNF2.** O processo de marcação e cancelamento de consultas deve ser fácil e intuitivo para os utilizadores; (**should have**)

**RNF3.** A aplicação deve ser capaz de se recuperar de falhas e retornar à operação normal com o mínimo de interrupções para os utilizadores; (**must have**)

**RNF4.** O processo de registo e login deve ser fácil e intuitivo para os utilizadores; **(should have)**

**RNF5.** O sistema de autenticação deve ter um tempo de resposta rápido e não diminuir o desempenho geral da aplicação; **(must have)**

**RNF6.** Tanto os utilizadores quanto os médicos devem ser capazes de compreender e usar facilmente a interface do Live-Chat; **(should have)**

**RNF7.** O Live-Chat deve ser capaz de suportar vários chats em execução ao mesmo tempo, sem a existência de quaisquer problemas; **(must have)**

**RNF8.** Os utilizadores autenticados devem sempre ter acesso ao Live-Chat; **(must have)**

- **Requisitos Funcionais (RF)**

**RF1.** Os utilizadores devem poder marcar consultas com médicos disponíveis em datas e horários específicos; **(must have / XL)**

**RF2.** Os utilizadores devem poder visualizar as suas próximas consultas e os respetivos detalhes; **(should have / XL)**

**RF3.** Os utilizadores devem poder cancelar as suas consultas previamente agendadas, que devem ser removidas da área pessoal e da base dados; **(must have / XL)**

**RF4.** Os utilizadores devem receber uma confirmação tanto para o agendamento como para o cancelamento das consultas; **(must have / L)**

**RF5.** A aplicação deverá enviar notificações aos utilizadores aquando da consulta **(should have / XL)**

**RF6.** Os utilizadores deverão conseguir realizar o registo na aplicação preenchendo um formulário com os seus dados pessoais (email, nome de utilizador e password); **(must have / M)**

**RF7.** Os utilizadores deverão conseguir realizar o login usando um nome de utilizador e uma password; **(must have / M)**

**RF8.** Os utilizadores devem poder recuperar a sua password usando o endereço de e-mail que definiram no ato do registo; **(should have / M)**

**RF9.** Após feito o login, os utilizadores devem poder aceder e editar os seus dados pessoais; **(must have / L)**

**RF10.** A aplicação deverá ter diferentes níveis de acesso com base na função do utilizador; **(must have / XL)**

**RF11.** O sistema deve ter um mecanismo para bloquear uma conta após várias tentativas de login malsucedidas; **(nice to have / L)**

**RF12.** A aplicação deve ser capaz de gravar as conversas do Live-Chat e armazená-las para referência futura. **(nice to have / L)**

**RF13.** A aplicação deve permitir o envio de fotos e mensagens através do Live-Chat entre o utilizador e o customer support (**nice to have / L**)

**RF14.** Ao fazer login, o utilizador deverá ter acesso ao Live-Chat sem necessidade de mais verificações (**should have / S**)

**RF15.** O utilizador deve conseguir registar os seus animais e estes aparecerem na sua área pessoal. (**must have / S**)

**RF16.** O utilizador deve ser capaz de editar e eliminar os seus animais (**must have / M**)

**RF17.** O admin deverá poder marcar, editar e desmarcar consultas dos utilizadores a pedido dos mesmos ou do médico envolvido. (**must have / M**)

**RF18.** O admin deverá poder editar certas secções da *Landing Page* (**must have / S**)

**RF19.** O utilizador deverá conseguir entrar em contacto com o Hospital Veterinário através de um formulário sem necessitar de realizar o registo / login no website (**nice to have / S**)

**RF20.** O médico deverá poder visualizar na sua conta as consultas que possui agendadas nos próximos 7 dias. (**nice to have / L**)

- **Requisitos Técnicos (RT)**

**RT1.** A aplicação deverá conseguir aceder e recolher dados de uma Base de Dados MongoDB; (**must have**)

**RT2.** A aplicação deve ser compatível com navegadores da web populares; (**must have**)

**RT3.** A aplicação deve ter um sistema de API para permitir integração e automação de terceiros. (**must have**)

**RT4.** O sistema de autenticação deve proteger contra o acesso não autorizado à aplicação e aos dados dos utilizadores; (**must have**)

**RT5.** Deve ser usada criptografia forte para proteger as passwords dos utilizadores e outras informações confidenciais; (**must have**)

**RT6.** O sistema de autenticação deve estar sempre disponível para os utilizadores; (**must have**)

**RT7.** O sistema deve ser capaz de garantir a complexidade das passwords dos utilizadores; (**should have**)

**RT8.** O JWT associado ao login fica armazenado numa cookie até o utilizador realizar o logout (**must have**)

## 4.2 Diagramas de Casos de Uso

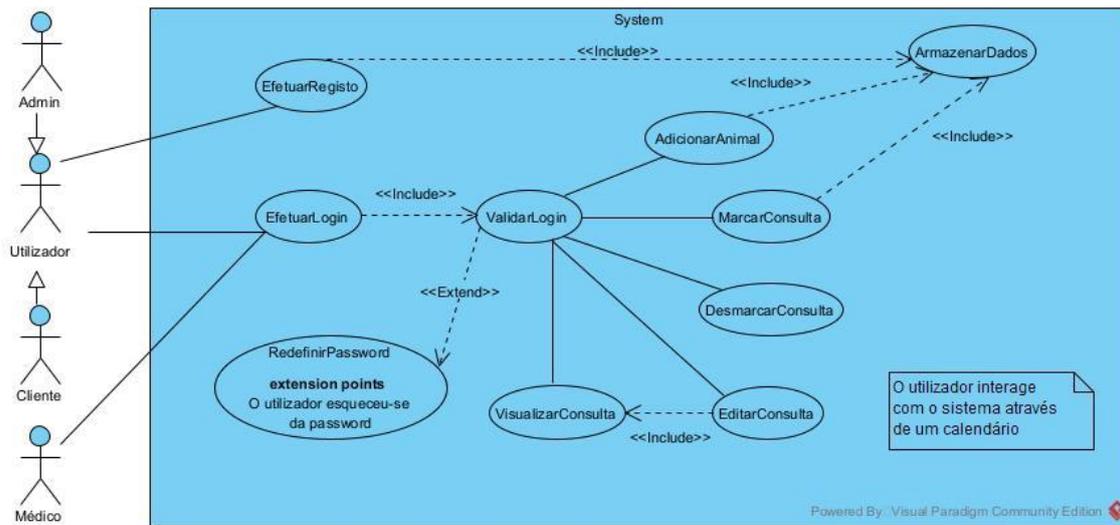


Figura 6 – Diagrama de Casos de Uso

## 4.3 Diagramas de Atividades

- Marcação de consulta

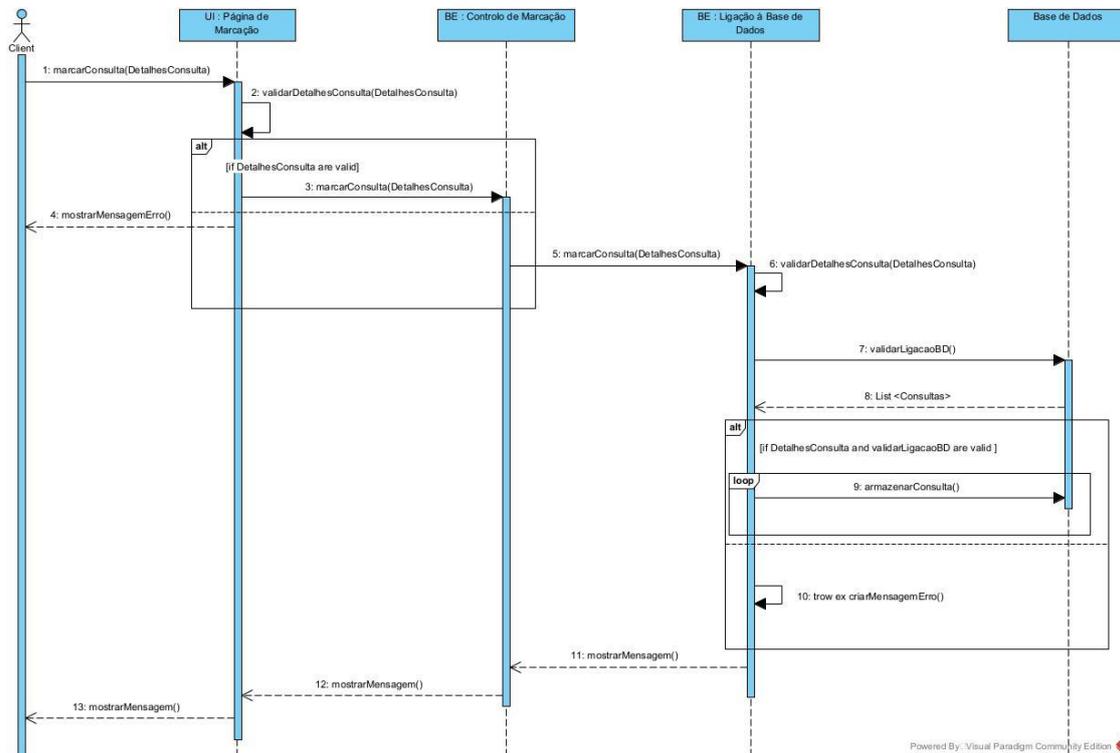


Figura 7 – Diagrama de Atividades (Marcação de consulta)

- Desmarcação de consulta

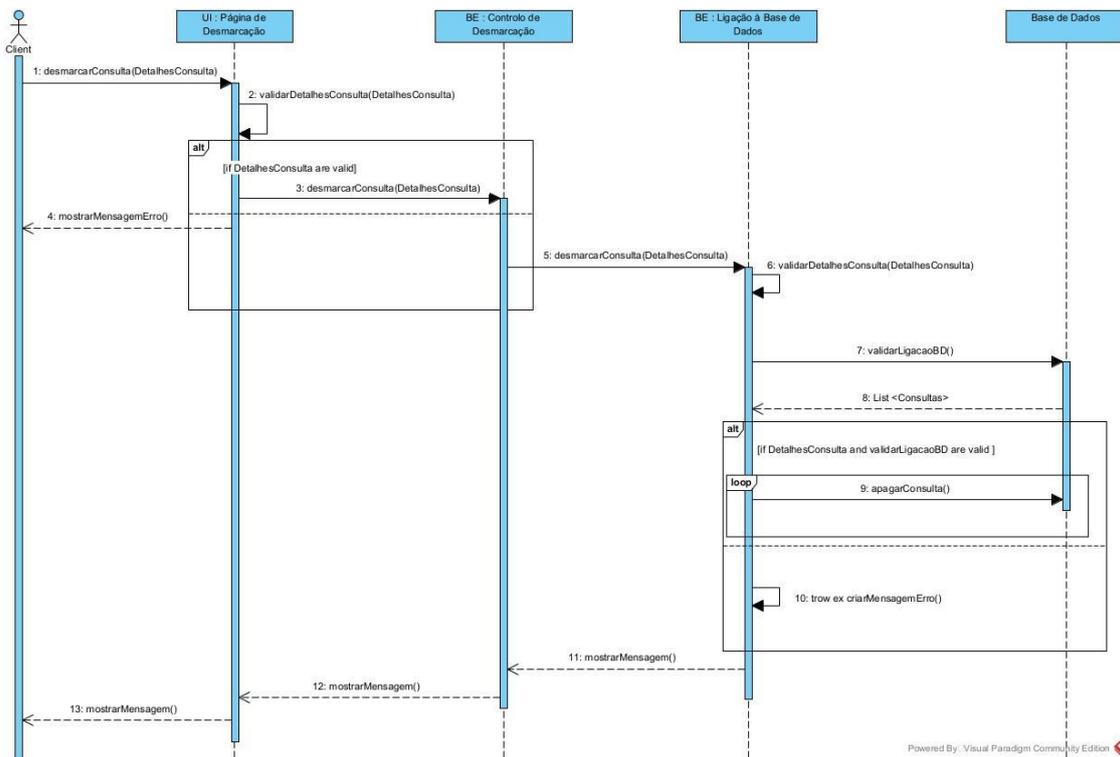


Figura 8 - Diagrama de Atividades (Desmarcação de consulta)

- Registo

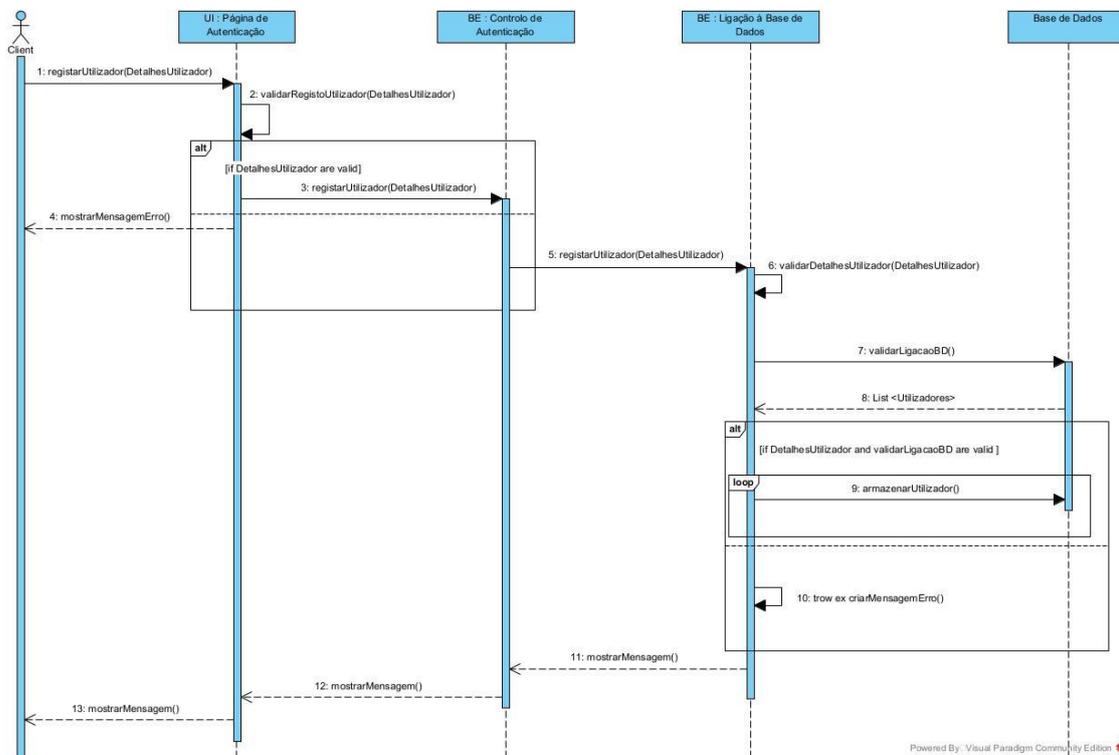


Figura 9 - Diagrama de Atividades (Registo)

- Login

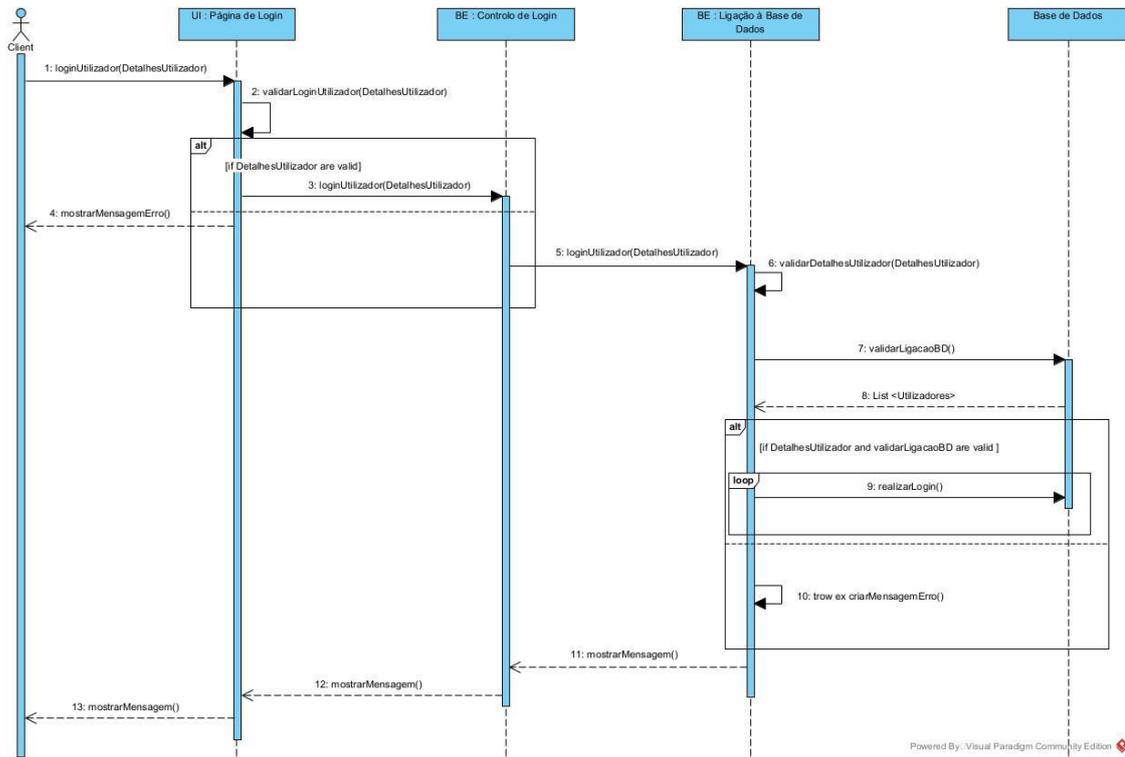


Figura 10 - Diagrama de Atividades (Login)

#### 4.4 Modelos relevantes

- Modelo Entidade-Relação (Diagrama Conceptual)

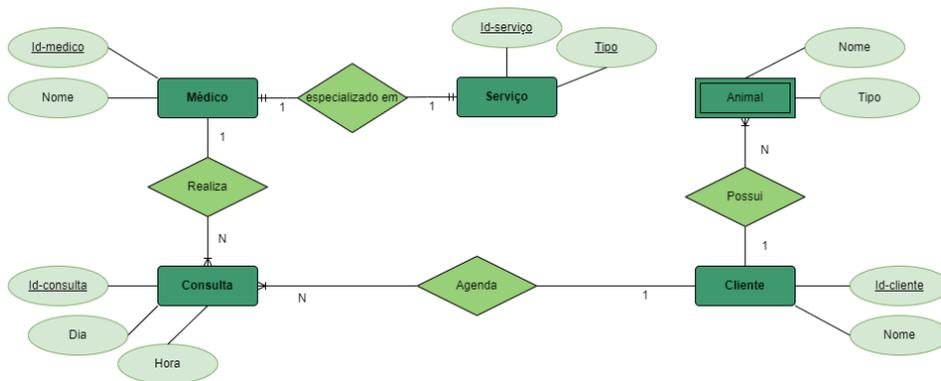


Figura 11 - Modelo Entidade-Relação (Conceptual)

● Diagrama de Classes

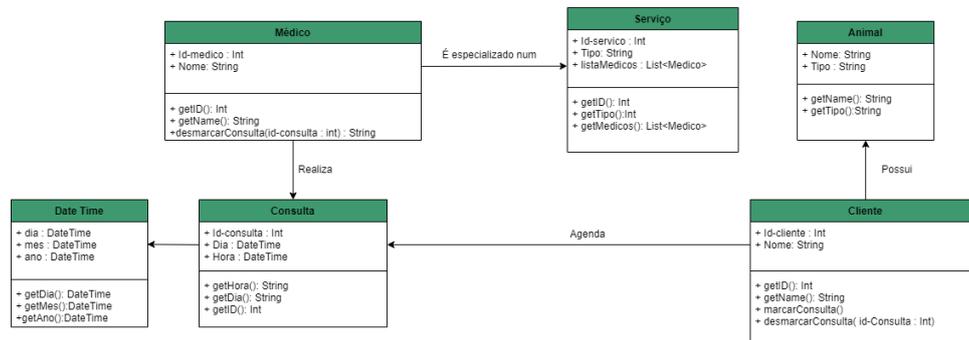


Figura 12 – Diagrama de classes

4.5 Estrutura

A nossa aplicação está estruturada em 4 níveis:

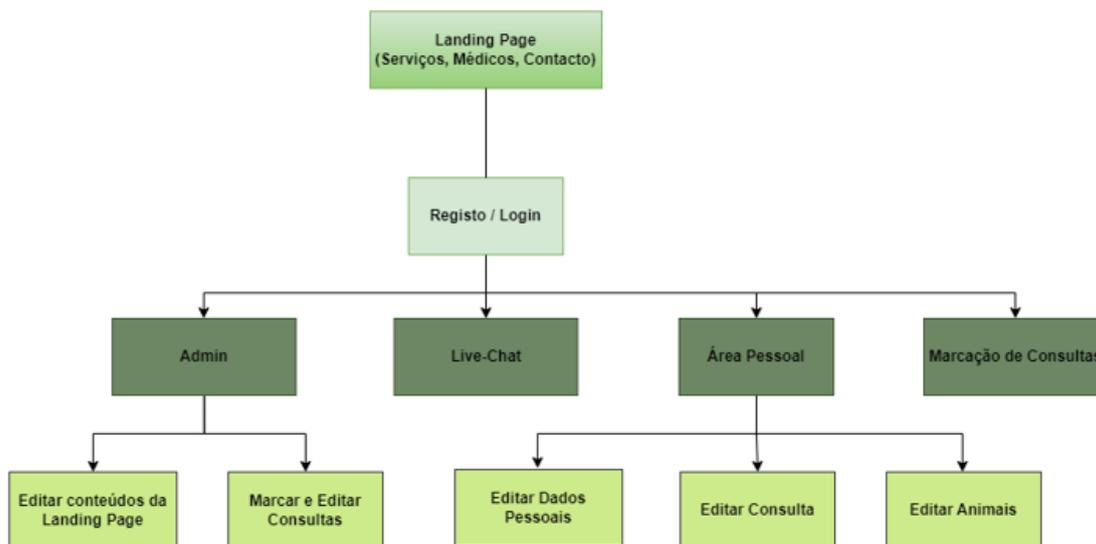
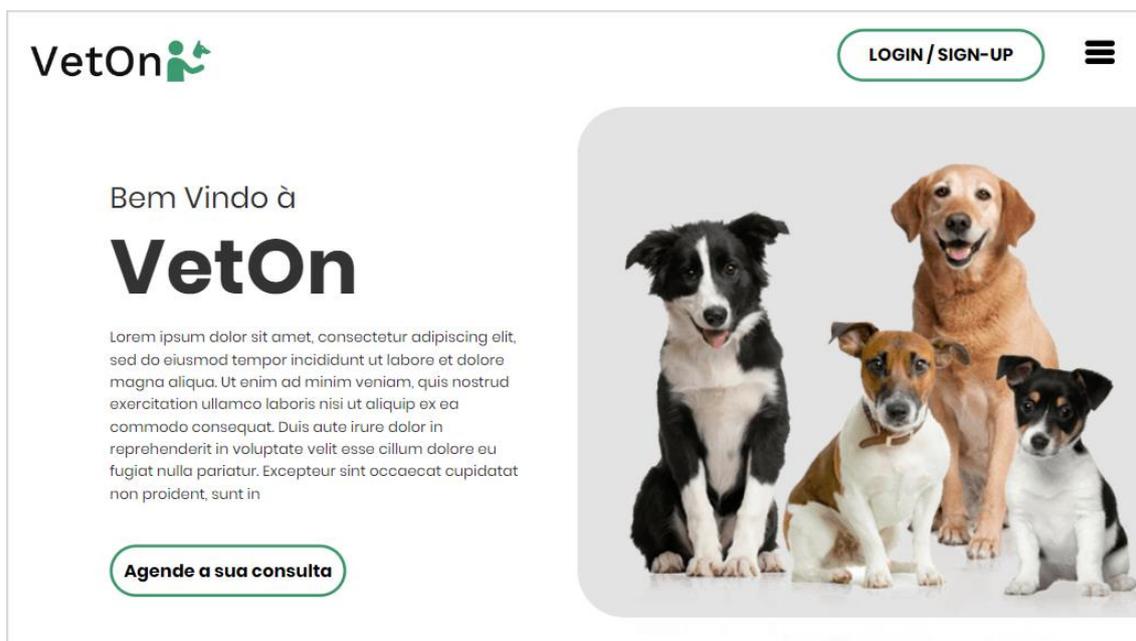


Figura 13 – Mapa aplicacional

## 4.6 Mockups e Story Board

De seguida apresentamos os *mockups* iniciais do nosso *website*. Ao aceder o *website* o utilizador tem acesso à *Landing Page* que se observa na Figura 2, na qual lhe é apresentado um botão através do qual este pode realizar a sua autenticação e posteriormente caso pretenda realizar a marcação de uma consulta. É também apresentado um pequeno texto descritivo acerca da *VetOn* para que este adquira alguma familiarização com o serviço.



**Figura 14 – Landing Page**

Ao aceder o *website* na *Landing Page* encontra-se um *hamburger button* que ao ser premido abre a *NavBar*, tal como se pode observar na Figura 3 e permite ao utilizador o acesso a todas as secções do *website* nomeadamente: *Home*, *Serviços*, *Médicos* e *Contactos*.

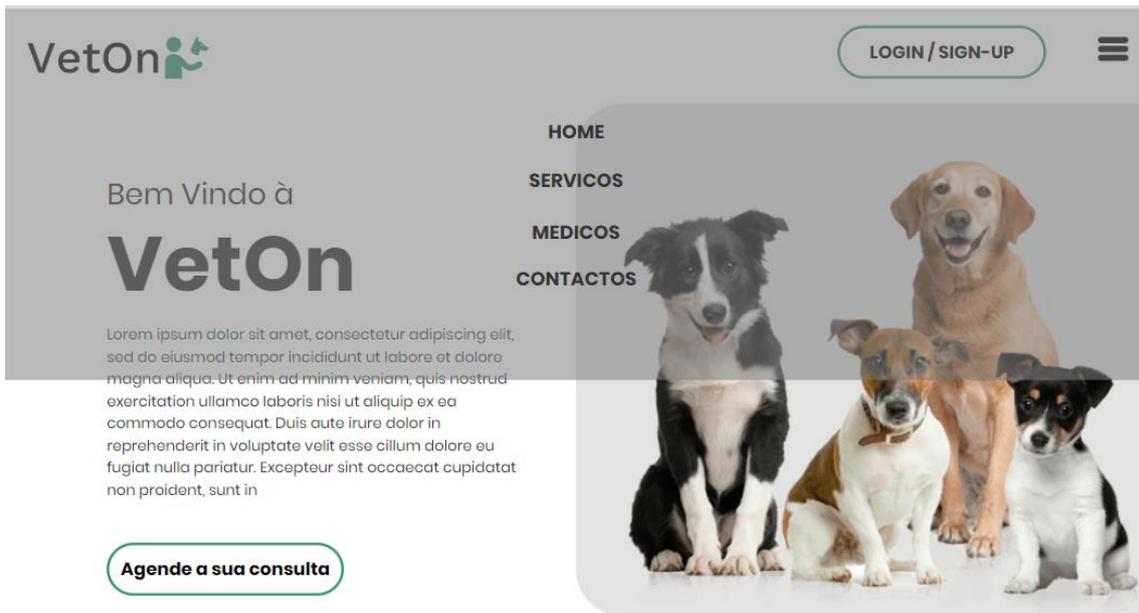


Figura 15 - NavBar

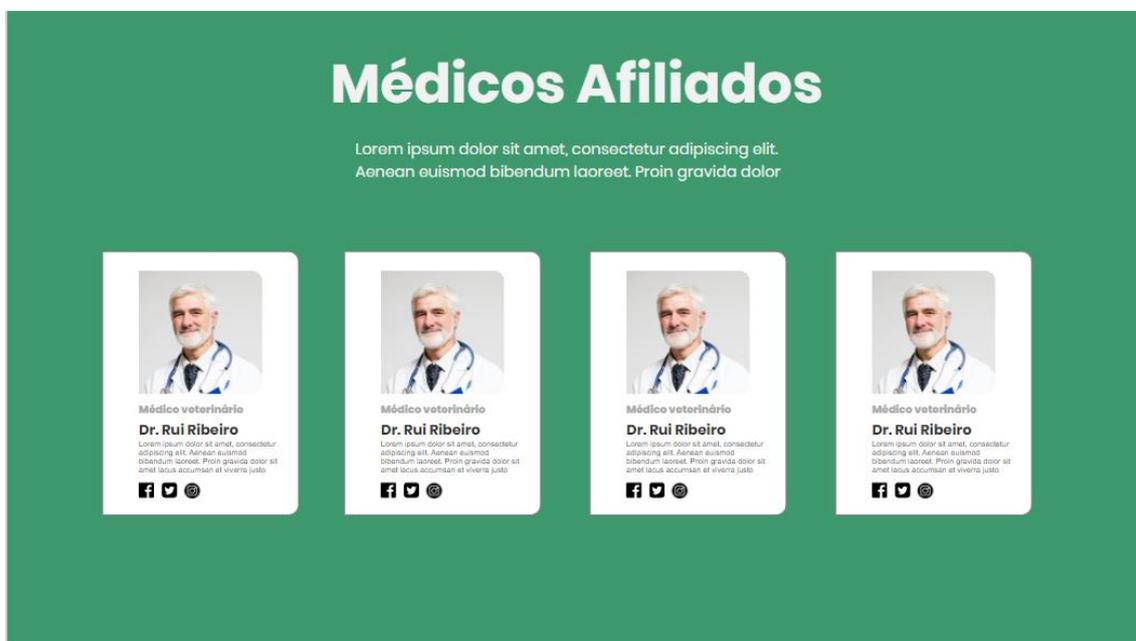
Uma das secções apresentadas no *website* são os Serviços, que se podem observar na Figura 4. Na secção abaixo são apresentados os serviços que o Hospital Veterinário da Universidade Lusófona tem para oferecer aos seus clientes.



Figura 16 – Secção “Serviços”

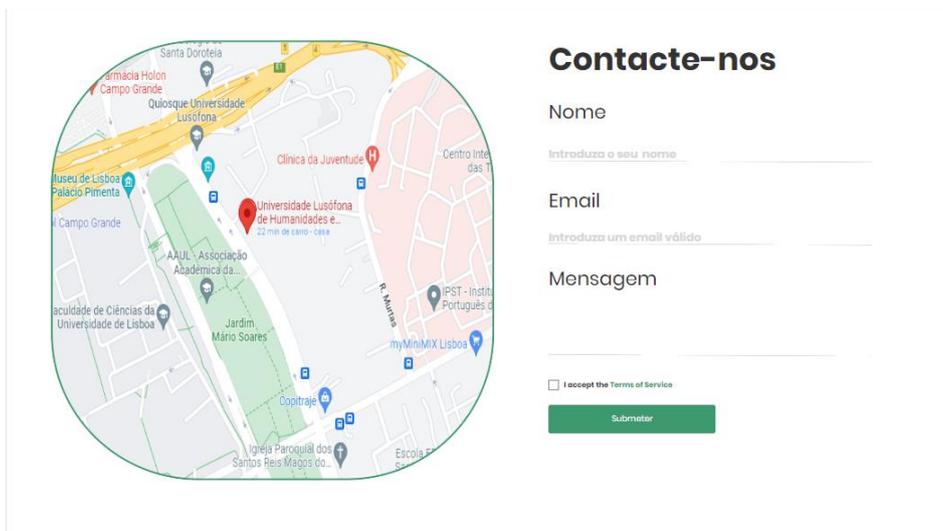
A secção seguinte apresentada no *website* é a dos Médicos Afiliados, tal como se pode observar na Figura 5.

Nesta secção são apresentados os médicos afiliados que o Hospital Veterinário da Universidade Lusófona possui.



**Figura 17 – Secção “Médicos Afiliados”**

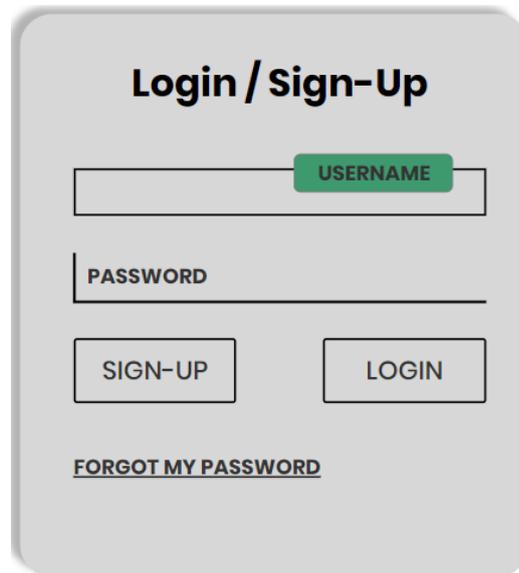
Por último é apresentada no *website* a secção de Contacto, conforme ilustrado na Figura 6. Os utilizadores podem verificar onde fica localizado o Hospital Veterinário da Universidade Lusófona, e caso pretendam, entrar em contacto com a instituição podem preencher o respetivo formulário.



**Figura 18 – Secção “Contactos”**

O utilizador ao aceder à página de Autenticação, tal como se observa na Figura 7, pode realizar o *login* se já possuir uma conta, ou caso não tenha, proceder ao registo.

Caso o utilizador se esqueça da sua *password* é também possível realizar a sua redefinição.



The image shows a login and sign-up form titled "Login / Sign-Up". It features a text input field for "USERNAME" with a green label, a text input field for "PASSWORD", a "SIGN-UP" button, a "LOGIN" button, and a link for "FORGOT MY PASSWORD".

**Figura 19 – Login / Registo**

## 5 Solução Proposta

### 5.1 Introdução

A solução que se propõem consiste no desenvolvimento de um *website* e de uma aplicação *mobile* para gerenciar consultas e serviços do Hospital Veterinário da Universidade Lusófona.

A solução será dividida em:

- 1- Desenvolvimento do *Front-end* do *website*;
- 2- Incorporação de uma base de dados capaz de relacionar as entidades necessárias;
- 3- Desenvolvimento da API (Application Programming Interface).
- 4- Comunicação entre a API e a base de dados
- 5- Realização de Media Queries para implementar a *Web View* (aplicação mobile)

- Link do vídeo da solução proposta:  
<https://www.youtube.com/watch?v=1on7JUKjQ8A>
- Link do repositório Git: <https://github.com/rbnvsilva/VetOn>

### 5.2 Arquitetura

A arquitetura da aplicação é baseada na arquitetura de 3 camadas, sendo esta constituída pelas seguintes camadas:

- **Camada de apresentação**  
A interface do utilizador e as camadas de comunicação da aplicação são encontradas na camada de apresentação, que também é onde o utilizador interage com o sistema. O principal objetivo desta camada é exibir e recolher dados do utilizador.
- **Camada Lógica**  
O núcleo da aplicação é a camada lógica. Nesta camada, os dados recolhidos na camada de apresentação são processados. Os dados da camada de dados também podem ser adicionados, removidos ou modificados pela camada lógica.
- **Camada de Dados**  
As informações processadas pela camada lógica são armazenadas e tratadas na camada de dados.

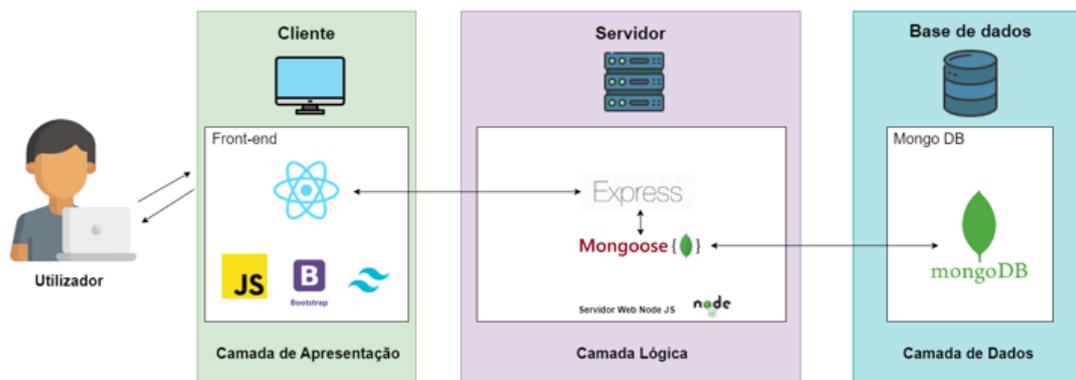


Figura 20 – Arquitetura da aplicação

### 5.3 Tecnologias e Ferramentas utilizadas

Para o desenvolvimento do nosso *website*, foi utilizada a MERN stack.

A MERN stack é uma stack de tecnologia de desenvolvimento Web muito apreciada, onde a sua abreviação ("MERN") indica as quatro tecnologias essenciais desta stack, sendo estas:

- **(M)ongoDB:** MongoDB é um banco de dados NoSQL que usa esquemas flexíveis e documentos que se assemelham a JSON para armazenar dados. Devido à sua excelente escalabilidade e velocidades rápidas de leitura e gravação, é uma opção popular para aplicações web. Este apresenta uma estrutura de documentos flexível, e bastante escalável o que permite o armazenamento de uma grande capacidade de dados de forma eficiente e eficaz
- **(E)xpress.js:** Express.js oferece uma solução rápida e adaptável para desenvolver aplicações web e APIs, conhecido pelo seu design simplificado e amplo suporte a middleware, este permite definir rotas para manipular as solicitações HTTP recebidas pelo website
- **(R)each:** React é uma biblioteca da linguagem JavaScript para construir *client* interfaces. Esta permite que os desenvolvedores criem componentes reutilizáveis e atualizem as interfaces de utilizador com eficiência quando os dados forem alterados. O React é amplamente usado para criar single-page applications (SPAs) e é conhecido pelo seu desempenho e escalabilidade.

- **(N)ode.js:** Node.js é um software de código aberto que permite a execução de códigos JavaScript fora de um navegador web. Este é construído no mecanismo JavaScript V8 do Chrome e fornece um modelo de E/S sem bloqueio e orientado a eventos que o torna adequado para a criação de aplicações Web escaláveis.

Para complementar a stack de desenvolvimento, ainda são usadas as seguintes tecnologias / ferramentas:

- **Bootstrap**
  - Bootstrap é uma framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web, usando HTML, CSS e JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do utilizador. Através desta framework é possível desenvolver websites responsivos que se ajustem aos diversos dispositivos que o utilizador possua de forma a manter uniforme a experiência de utilização do website
- **Tailwind CSS**
  - Tailwind CSS é uma framework de CSS do tipo utility-first que ajuda na construção de interfaces de usuário personalizadas de forma rápida e eficiente, oferece um conjunto amplo de utilitários que podem ser combinados para criar estilos únicos e interfaces altamente personalizadas.
- **Mongoose**
  - Mongoose é uma *framework* da linguagem de programação JavaScript. Foi utilizada para fazer a conexão entre a nossa base de dados MongoDB e o ambiente de desenvolvimento node.js. É amplamente utilizado pela comunidade Node.js para simplificar o acesso e a manipulação à base de dados MongoDB, fornece uma camada de abstração poderosa e flexível que facilita o desenvolvimento de aplicações, oferecendo recursos adicionais à funcionalidade básica fornecida pelo banco de dados.
- **Docker**

Docker é um conjunto de produtos de plataforma como serviço que usam virtualização de nível de sistema operativo para entregar software em pacotes chamados containers, tornou-se amplamente adotado na indústria devido à sua portabilidade, facilidade de uso e eficiência, este facilita a criação, implantação e execução de aplicativos em containers, proporcionando maior flexibilidade, isolamento e escalabilidade. Com o Docker, é possível criar ambientes de desenvolvimento consistentes, facilitar a colaboração entre equipas e agilizar o processo de implantação de aplicativos.

- **Bcrypt**
  - Método de criptografia do tipo *hash* para senhas baseado na cifra *Blowfish*. Todas as *passwords* introduzidas pelos utilizadores são encriptadas por este método e posteriormente armazenadas na nossa BD. Este método é bastante recomendado para garantir a segurança das senhas dos utilizadores ao utilizá-lo protegemos as informações confidenciais destes, reduzindo significativamente o risco de exposição em caso de ataques.
- **JWT (JSON Web Token)**
  - Padrão para a criação de dados com assinatura opcional e/ou criptografia. Na nossa aplicação, ao ser realizado o *Login*, e criada uma cookie com o respetivo token.
- **Cookie**
  - É responsável por guardar a informação do utilizador resultante do JWT (JSON Web Token) no ato do login, de forma a conseguirmos manter a conta apresentável até este realizar o logout
- **Git**
  - O Git é amplamente utilizado por desenvolvedores de software e é uma parte fundamental do fluxo de trabalho moderno de desenvolvimento colaborativo, oferece uma maneira eficiente de controlar as versões do código, trabalhar em paralelo e colaborar com outros membros da equipa. Com o Git, é possível acompanhar as alterações, restaurar versões anteriores e colaborar de forma eficaz em projetos de qualquer tamanho. É um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software. Utilizado para realizar o versionamento da nossa aplicação e uma maior facilidade de partilha do mesmo.
- **Postman**
  - Postman é uma plataforma de *API* para desenvolvedores, para construir e testar as *APIs* da aplicação, este oferece uma interface intuitiva e diversas funcionalidades úteis para facilitar o trabalho com APIs nas diferentes fases de desenvolvimento do projeto
- **Amazon S3**
  - O Amazon S3 é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade de dados, segurança e desempenho líderes do setor. Podemos facilmente armazenar e proteger qualquer quantidade de dados para casos de uso como *data lakes*, sites, aplicativos nativos de nuvem, *backups*, arquivamento, *machine learning* e análises. O Amazon S3 foi projetado para 99,999999999% (11 noves) de durabilidade e armazena dados de milhões de clientes para empresas em todo o mundo.

- **Vercel**
  - A Vercel é conhecida por ser uma das principais opções para realizar o *host* de sites estáticos e aplicações *SPA - Single Page Applications*, oferece integração com várias *frameworks* populares, como o Next.js, React, Angular e Vue.js.  
No nosso caso foi utilizada para que pudéssemos colocar o nosso projeto online de forma a ser acessível por qualquer utilizador através de um link.

## 5.4 Implementação

No que toca a implementação da nossa aplicação, esta será dividida nos seguintes tópicos:

- **Gestão de versões**

Foi criado um repositório no GitHub (referido anteriormente no ponto [5.1](#)) de modo a ser utilizada a ferramenta de versionamento Git. Ambos os autores desta aplicação realizaram “*commits*” a cada funcionalidade desenvolvida com uma mensagem específica, de modo a ser mais fácil gerir todo o trabalho realizado e não realizado ao longo do tempo.

Qualquer pessoa interessada em usufruir do código desenvolvido, poderá fazer um “*pull*” do repositório do GitHub e realizar as devidas alterações, como criar um `.env` com as variáveis de ambiente, uma base de dados MongoDB, entre outros...

- **Instalação, Manutenção e *deploy***

De modo a facilitar todo o processo de instalação, manutenção e *deploy* da nossa aplicação, decidimos optar pela ferramenta Docker.

Foram criadas duas imagens (Client / API) da nossa aplicação (*docker image*) de modo a ser possível a execução do código desenvolvido em *docker containers*. Com estas *docker images*, é possível dar *upload* das mesmas para o *DockerHub*, onde qualquer pessoa pode fazer um “*pull*” (processo similar ao GitHub) e ter toda a aplicação a funcionar na sua máquina, sem necessidade de qualquer configuração adicional.

Todos os passos para a sua correta instalação, encontram-se no [README.md](#) do repositório GitHub.

- **Estrutura do website**

Como dito anteriormente, no que toca ao website, iremos ter maioritariamente duas divisões:

- **Client**

- **Assets** – *static assets*, como imagens, vídeos, fontes e outros arquivos que são necessários para a aplicação, mas não fazem parte do código JavaScript da mesma;
- **Components** – elementos da interface de utilizador ou uma funcionalidade específica;
- **Pages** – páginas do *website* que fazem uso dos *components* criados e serão acessadas através das *routes* definidas;
- **Routes** – existe um ficheiro no nosso projeto, denominado “App.jsx”, onde são criadas as ligações entre um *URL* específico e a página a que este está associado quando é feita uma requisição.
- **API**
  - **Models** - ficheiros “.js” que implementam as entidades necessárias para o funcionamento da aplicação;
  - **Uploads** – ficheiros, maioritariamente imagens, que são enviados pelos utilizadores;
  - **Endpoints** – ficheiros “.js” na directoria “api” que implementa a lógica necessária para enviar dados da API ou receber dados por parte do *cliente-side*.
  - **.env** – um ficheiro .env e um ficheiro de configuração de variáveis específicas para o projeto que não devem ser visíveis no código desenvolvido. Exemplos destas variáveis são *passwords*, *URLs*, *connection strings*, entre outras;
- **Teste de API**

Para serem realizados testes na *API* da nossa aplicação foi utilizada a ferramenta Postman para requisitar métodos *GET* e *POST*, acedendo ao *URL* da nossa *API* para onde é enviado um *raw body* em *JSON* com um ou mais pares de chave-valor necessários.

## 5.5 Abrangência

Esta solução envolve conhecimentos já adquiridos e trabalhados em algumas áreas curriculares (UCs) da licenciatura em Engenharia Informática, tais como:

- **Programação Web**

Serão abrangidos alguns conteúdos transmitidos nesta UC, que no contexto do nosso projeto, fazemos uso dos conhecimentos relativos a SPAs (*single-page application*) e comunicação entre *client-side* e *server-side* (*API*).

- **Interação Humano-Máquina**

Para a realização de todos os *mock-ups* produzidos com o intuito de desenvolver a solução proposta, serão utilizados conhecimentos em *UI/UX* e utilização da ferramenta de prototipagem Axure RP 10, lecionados na UC respectiva.

- **Engenharia de Requisitos e Testes**

Utilização de todo o conhecimento referente a levantamento de requisitos, bem como a modelação de diagramas relativos aos mesmos.

- **Computação Móvel**

Para a realização da aplicação *mobile*, iremos utilizar os conhecimentos retidos desta UC relativamente ao uso de *WebView*, ou seja, usar o nosso *website* e *API* como estrutura para a nossa aplicação *mobile*.

- **Algoritmia e Estrutura de Dados**

Visto que entender algoritmos e estruturas de dados é crucial para um programador, serão usados métodos e teoria aprendidos nesta UC para criar código simples e eficaz. Faremos uso das estruturas de dados apropriadas para a situação, bem como o uso de algoritmos de ordenação.

- **Sistemas de Suporte à Decisão**

Será importante utilizar conhecimentos desta UC para realizar a modelação de todos os dados e estruturas referentes a aplicação.

- **Engenharia de Software**

Devido a ser necessária a modelação de diagramas de atividades bem como ter noções básicas de abordagens no que toca ao desenvolvimento de projetos, e uma mais valia fazer uso da teoria aprendida nesta UC.

## 6 Plano de testes e validação

<b>Título</b>	<b>Descrição</b>	<b>Identificação dos Requisitos</b>	<b>Resultado Esperado</b>	<b>Resultado Obtido</b>
<b>Registo do utilizador</b>	O utilizador ao aceder ao website realiza o registo preenchendo o formulário	RF7	O utilizador cria conta no website	O utilizador cria conta e esta é armazenada na base de dados
<b>Login no website</b>	O utilizador realiza o login no website utilizando as credenciais definidas no registo	RF8	O utilizador possui acesso as totais funcionalidades de um utilizador	O utilizador possui acesso à sua área pessoal, pode marcar consultas e registar os seus animais
	O utilizador realiza o login no website preenchendo as credenciais erradas		O utilizador é alertado de que as credenciais inseridas estão erradas	Ao clicar no botão de login o utilizador é alertado de que as credenciais inseridas se encontram erradas
<b>Recuperação de Password</b>	O utilizador pretende realizar a alteração da sua password, acede à sua área pessoal e clica na opção alterar password	RF9	O utilizador preenche o formulário indicando a nova password e ao submeter esta é alterada	A password é alterada e armazenada na base de dados
<b>Edição dados pessoais</b>	Após realizar o login o utilizador dirige-se para a sua área pessoal e edita os seus dados	RF10	O utilizador preenche o formulário para editar os seus dados pessoais	O utilizador edita os seus dados pessoais e estes são atualizados na base de dados

<b>Diferentes níveis de acesso</b>	O website apresenta 3 níveis de acesso, o utilizador, o <i>admin</i> e o <i>support</i>	RF11 RF18 RF19	Ao realizar login com a conta <i>admin</i> este tem acesso a funcionalidades extra	Após o login em modo <i>admin</i> , este possui acesso à edição de páginas e marcações / desmarcações de consultas
	O website apresenta 3 níveis de acesso, o utilizador, o <i>admin</i> e o <i>support</i>		Ao realizar login com a conta de <i>support</i> este tem acesso ao Live-Chat onde possui conversas com todos os utilizadores	Após o login em modo <i>support</i> este possui acesso a todas as mensagens enviadas pelos utilizadores
<b>Marcação de consultas</b>	Após realizado o login o utilizador marca uma consulta para um médico e horário disponível	RF1	A consulta é agendada para o medico e horário escolhido	A consulta é agendada, sendo armazenada na base de dados
<b>Visualização consultas agendadas</b>	Após a marcação da consulta esta é apresentada ao utilizador na sua área pessoal	RF2	É possível visualizar as consultas agendadas	São apresentadas as consultas agendadas marcadas por esse utilizador
<b>Desmarcação de consultas</b>	O utilizador desmarca a consulta que tinha previamente agendado	RF4	A consulta é desmarcada	A consulta é desmarcada sendo automaticamente removida da base de dados
<b>Alteração da consulta</b>	O utilizador altera a data da consulta que tinha agendado	RF3	A consulta é reagendada	A consulta é reagendada para o dia solicitado sendo alterada a sua data na base de dados
<b>Acesso ao Live-Chat</b>	O live-chat deve estar disponível para o utilizador assim que este realize o login	RF15	Após o login o utilizador tem acesso ao Live-Chat	Ao realizar o login o utilizador tem acesso ao Live-Chat através de um botão

<b>Envio de mensagens através do Live-Chat</b>	O utilizador acede ao Live-Chat enviando uma mensagem ao <i>customer support</i>	RF14	O <i>customer support</i> recebe a mensagem do utilizador	A mensagem é recebida e armazenada na base de dados
<b>Envio de fotografias através do Live-Chat</b>	O utilizador acede ao Live-Chat enviando uma fotografia ao <i>customer support</i>	RF14	O <i>customer support</i> recebe a fotografia enviada pelo utilizador	A fotografia é enviada para o <i>customer support</i> e armazenada na pasta “uploads”
<b>Armazenamento de mensagens do Live-Chat</b>	As mensagens enviadas entre o utilizador e o Live-Chat são armazenadas numa base de dados	RF13	Armazenamento das mensagens do Live-Chat numa base de dados	As mensagens provenientes do Live-Chat são armazenadas na base de dados
<b>Adição de animais</b>	O utilizador adiciona os seus animais e pode consultá-los na sua área pessoal	RF16	São adicionados à base de dados os animais referentes ao utilizador	Os animais adicionados pelo utilizador são armazenados na base de dados e são apresentados na área pessoal respetiva
<b>Edição e eliminação de animais</b>	O utilizador edita ou elimina os seus animais através da sua área pessoal	R17	Assim que seja editado ou eliminado o animal este é automaticamente atualizado na base de dados	Após editar ou eliminar o animal é possível verificar as alterações na base de dados bem como na área pessoal respetiva
<b>Formulário de Contacto</b>	O utilizador preenche o formulário de contacto	R20	A mensagem proveniente do formulário é enviada	Após preencher o formulário de contacto e submetê-lo a mensagem é enviada

Tabela 2 - Tabela de testes

De forma a corrigir possíveis erros na nossa solução realizamos um guião de testes disponível no [[Anexo3](#)], que os utilizadores devem seguir de forma a testar todas as funcionalidades do website.

No final do guião é disponibilizado aos utilizadores um formulário, desenvolvido através da Ferramenta *Google Forms* para que estes nos possam dar feedback de acordo com as tarefas por nós solicitadas.

Link do formulário: <https://forms.gle/V7AJA5kwuj7tfwNj9>

Após termos dado como concluído o desenvolvimento do website fornecemos o guião de testes aos utilizadores, obtivemos até à data da entrega 15 repostas ao formulário. Através de uma análise aos resultados dos testes [[Anexo4](#)], conseguimos obter um feedback bastante positivo uma vez que a avaliação geral do trabalho foi superior a 3 numa escala de 1-5, o que garante que os utilizadores conseguiram realizar a totalidade dos passos pretendidos, ainda assim obtivemos a informação de 2 que tiveram algumas dificuldades, uma vez que o host da vercel é free e por vezes o tempo de resposta do website aumenta.

De um modo geral os utilizadores consideraram que o projeto se encontrava simples e intuitivo, forneceram-nos também algumas sugestões que iremos implementar de forma a fornecermos a melhor solução possível.

## 7 Método e Planeamento

Para uma melhor organização e como forma de atingirmos os nossos objetivos para este Trabalho Final de Curso, optamos por utilizar a ferramenta *Project Libre* para realizar o planeamento. Desta forma, estão destacadas as *milestones* que achamos mais importantes no decorrer do desenvolvimento do nosso projeto.

No que toca ao desenvolvimento do *website*, temos:

*Design – Mockups* realizados no programa Axure RP 10.

*Front-end* – Desenvolvimento da *Landing page*, página para autenticação e página para marcação de consultas.

*Back-end* – Inicialização da base de dados MongoDB e criação das entidades e relações entre as mesmas;

Criação de APIs.

No que toca ao desenvolvimento da aplicação *mobile*, temos:

*Design – Mockups* realizados no programa Axure RP 10.

*Back-end* – Sincronização com a base de dados criada no *website*.

Por fim, realizar testes tanto ao *website* quanto a aplicação *mobile*.

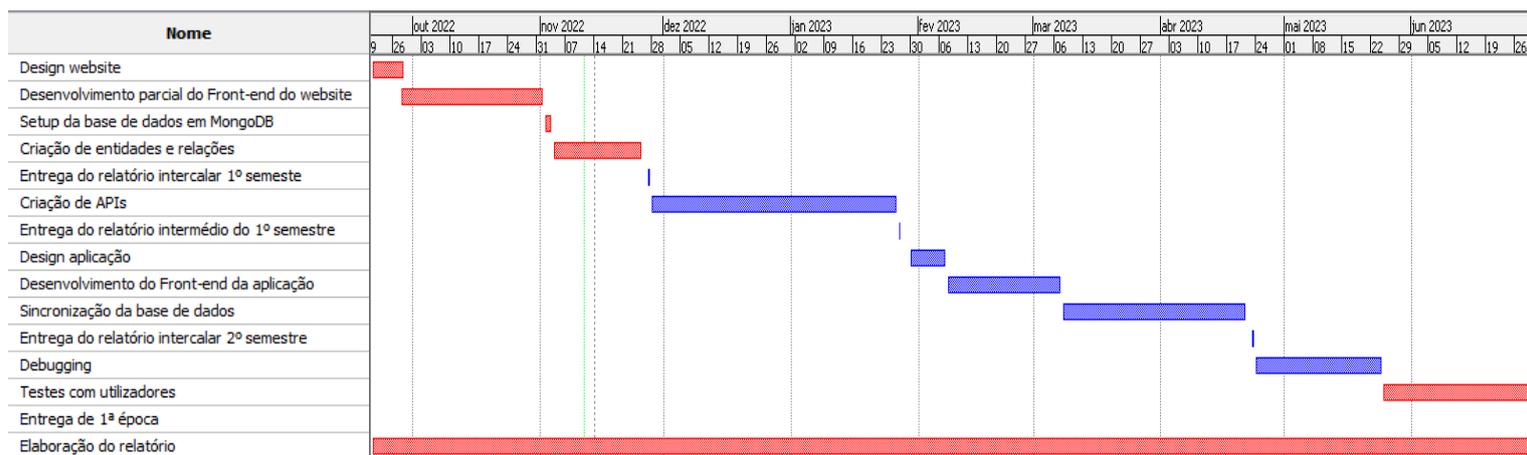


Figura 21 – Planeamento em formato GANTT

## 8 Conclusão e trabalhos futuros

Ao desenvolver o trabalho final de curso conseguimos compreender muito bem como funcionam todas as fases de um projeto bem como de que forma devemos gerir o tempo, para que no final consigamos cumprir com tudo aquilo que previamente nos comprometemos a desenvolver.

Passamos por diversas fases sendo que aquelas que possuímos maiores dificuldades foram a escolha do tema e a escolha das tecnologias a utilizar uma vez que não tínhamos muita noção de quais seriam mais vantajosas ou se adaptariam melhor tendo em conta as nossas necessidades, ainda assim cumprimos com praticamente todas as tarefas que nos propusemos a realizar no ato do Levantamento e Análise de requisitos, à exceção do desenvolvimento da aplicação devido a uma má gestão do tempo e do envio de fotografias através do Live Chat. A solução que desenvolvemos é bastante abrangente uma vez que pode ser adaptada a qualquer clínica ou hospital veterinário que desta pretenda usufruir.

Estamos muito satisfeitos com o resultado do nosso trabalho, ao longo destes meses tentamos ao máximo desenvolver um projeto que pudesse vir a ser utilizado pelo Hospital Veterinário da Universidade Lusófona e que fosse bastante simples e intuitivo de forma a melhorar todos os processos do agendamento e gestão das consultas.

Consideramos que o desenvolvimento deste projeto nos trouxe inúmeras vantagens para podermos ingressar no mercado de trabalho, uma vez que nos trouxe a noção de como funcionam os projetos e todas as fases que por estes passam para que possam ter sucesso.

Por fim não nos podíamos esquecer de agradecer aos nossos orientadores os professores João Pedro Leal Abalada de Matos Carvalho e o professor João Pedro Calado Barradas

Branco Pavia que sempre se mostraram disponíveis para ajudar e não nos deixaram perder o foco ao longo destes meses de trabalho.

## Bibliografia

[1] *Website* Notícias ao minuto, Notícia sobre o estudo da plataforma Fixando [Online].

Available: <https://www.noticiasao minuto.com/pais/1701413/adocao-de-animais-em-portugal-aumenta-32-com-pandemia>

[Acedido em: 19-Nov 2022]

[2] *Website* Faculdade Medicina Veterinaria, Secção do Hospital Veterinário da Universidade Lusófona [Online].

Available: <https://fmv.ulusofona.pt/>

[Acedido em: 19-Nov 2022]

[3] *Website* da aplicação “Zu”, Design da aplicação [Online].

Available: <https://www.zuonline.pt/appzu>

[Acedido em: 23-Out 2022].

[4] *Website* da Apple Store referente à aplicação “Vetecare”, Design da aplicação [Online].

Available: <https://apps.apple.com/pt/app/vetecare/id1245689386>

[Acedido em: 24-Out 2022].

[5] *Website* da Apple Store referente à aplicação “Petappoint”, Design da aplicação [Online].

Available: <https://apps.apple.com/pt/app/petappoint/id1403496721>

[Acedido em: 24-Out 2022].

[6] *Website* da aplicação “Petable”, Design da aplicação [Online].

Available: <https://petable.care/pt/home-2/>

[Acedido em: 24-Out 2022].

[7] *Figura 7; Figura 8; Figura 9; Figura 10*, Diagrama de Caso de Uso e Diagramas de Atividades (elaborados pelos autores através da ferramenta “Visual Paradigm”).

[8] *Figura 11; Figura 12; Figura 13; Figura 20*, Diagrama Entidade-Relacao (conceitual), Diagrama de classes, Mapa Aplicacional e Arquitetura da aplicacao (elaborados pelos autores através do Website “[draw.io](https://draw.io)”).

[9] *Figura 14; Figura 15; Figura 16; Figura 17; Figura 18; Figura 19*, Mockups Website (elaboradas pelos autores através da ferramenta “Axure RP 10”).

[10] *Figura 21*, Planeamento em Gantt (elaborada pelos autores através da ferramenta “Project Libre”).

[11] *Tabela 1*, Benchmarking (elaborada pelos autores através da ferramenta “Microsoft Excel”).

[12] *Tabela 2; Tabela 3; Tabela 4*, Tabela de Requisitos Não Funcionais, Tabela de Requisitos Funcionais e Tabela de Requisitos Tecnicos (elaborada pelos autores através da ferramenta “Microsoft Word”).

## Anexo 1 – Questionário

# VetOn

### TFC - VetOn

No âmbito da unidade curricular Trabalho Final de Curso de LEI (Licenciatura em Engenharia Informática) da Universidade Lusófona, sob orientação do Professor Dr. João Pedro Leal Abalada De Matos Carvalho e do Professor Dr. João Pedro Calado Barradas Branco Pavia, gostaríamos da sua colaboração para preencher o formulário abaixo. Este formulário tem como objetivo recolher informações para ajudar no desenvolvimento de um website/app para o Hospital Veterinário da Universidade Lusófona.

Este questionário foi realizado pelos alunos Rodrigo Simões e Ruben Silva.

**Todos os dados recolhidos são confidenciais, anónimos e utilizados unicamente para fins de investigação.**

ruben.s8101@gmail.com [Mudar de conta](#) 

\*Obrigatório

Email \*



Sabia que a Universidade Lusófona possui um Hospital Veterinário onde pode marcar consultas para o seu animal de estimação? \*

Sim

Não

Já realizou alguma consulta no Hospital Veterinário da Universidade Lusófona? \*

Sim

Não

Se sim, teve alguma dificuldade em marcar a consulta ou em encontrar informações acerca do Hospital Veterinário da Universidade Lusófona?

Sim

Não

Considera necessário o desenvolvimento de um *website/app* para gerenciamento \* de consultas e acompanhamento de um animal de estimação?

não é necessário      0   1   2   3   4   5   6   7   8   9   10      seria bastante útil

Considera que um *LiveChat* com possibilidade de enviar uma foto seria uma boa \* adição à *app*?

Sim

Não

Considera necessária a apresentação dos serviços prestados pelo Hospital \* Veterinário da Universidade Lusófona?

Sim

Não

Considera necessária a apresentação dos médicos afiliados ao Hospital \* Veterinário da Universidade Lusófona?

Sim

Não

Gostaria de ter sincronização do calendário do seu *smartphone* com o da *app*? \*

Sim

Não

Gostaria de receber notificações aquando da consulta marcada? \*

Sim

Não

Que informações pertinentes deveriam ser apresentadas no perfil de animal de estimação? Ex: foto, peso, etc... \*

A sua resposta \_\_\_\_\_

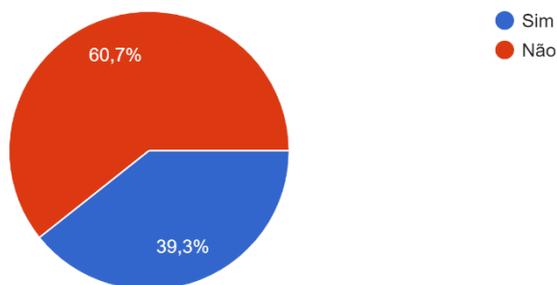
Além do que já foi questionado, o que adicionava num *website/app* para gerenciamento de consultas e acompanhamento de um animal de estimação? \*

A sua resposta \_\_\_\_\_

## Anexo 2 – Resultados do questionário

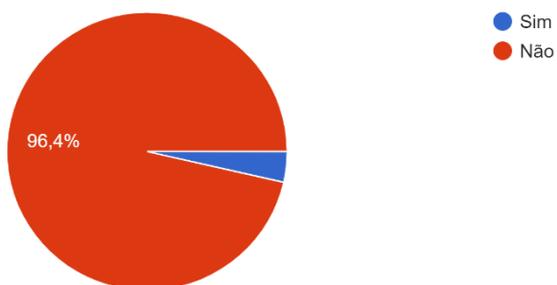
Sabia que a Universidade Lusófona possui um Hospital Veterinário onde pode marcar consultas para o seu animal de estimação?

28 respostas



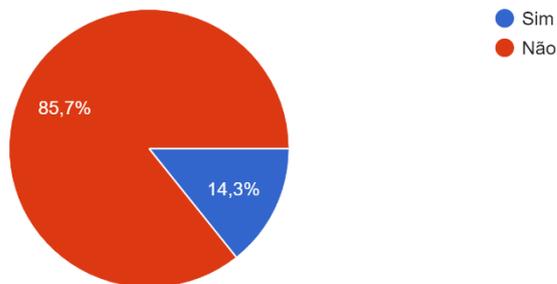
Já realizou alguma consulta no Hospital Veterinário da Universidade Lusófona?

28 respostas



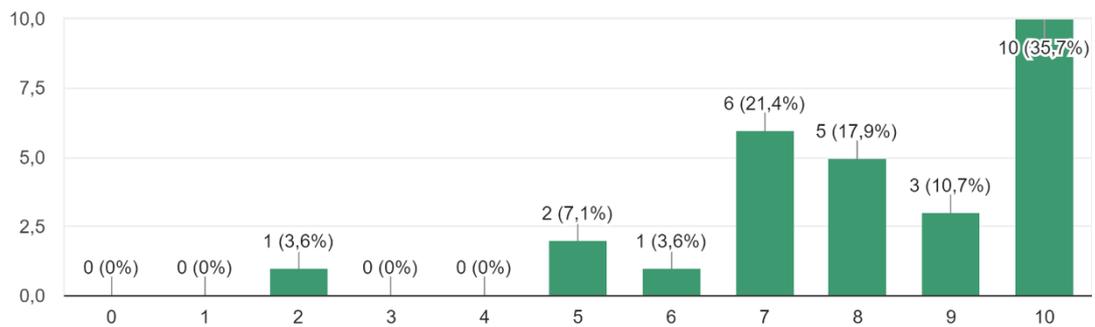
Se sim, teve alguma dificuldade em marcar a consulta ou em encontrar informações acerca do Hospital Veterinário da Universidade Lusófona?

7 respostas



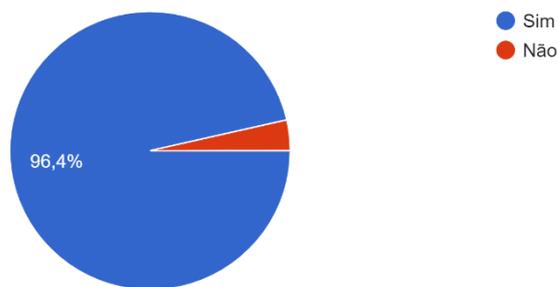
Considera necessário o desenvolvimento de um website/app para gerenciamento de consultas e acompanhamento de um animal de estimação?

28 respostas



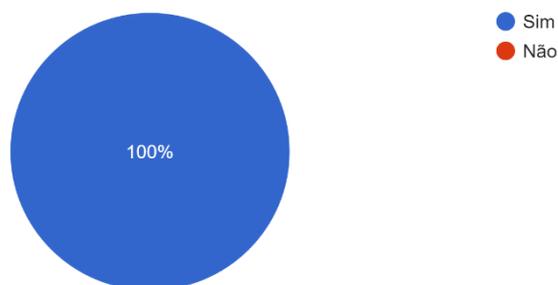
Considera que um LiveChat com possibilidade de enviar uma foto seria uma boa adição à app?

28 respostas



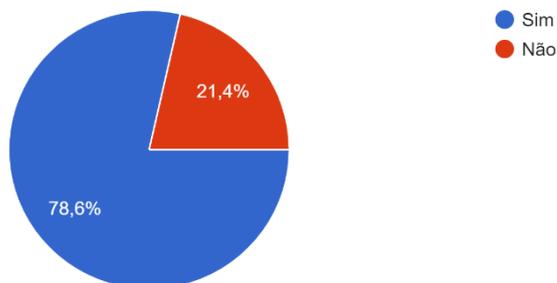
Considera necessária a apresentação dos serviços prestados pelo Hospital Veterinário da Universidade Lusófona?

28 respostas



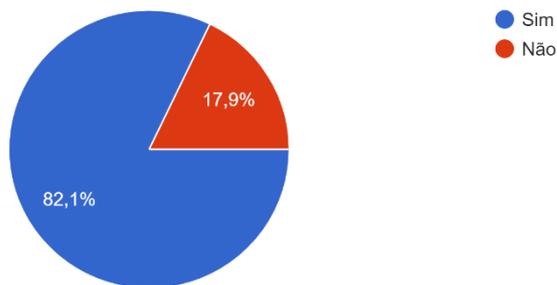
Considera necessária a apresentação dos médicos afiliados ao Hospital Veterinário da Universidade Lusófona?

28 respostas



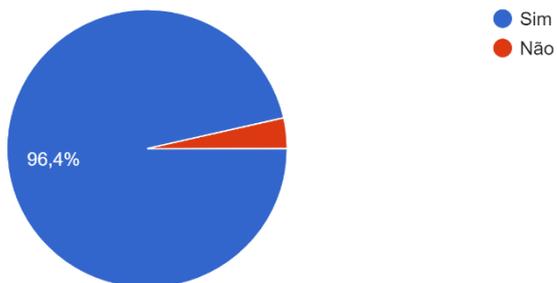
Gostaria de ter sincronização do calendário do seu smartphone com o da app?

28 respostas



Gostaria de receber notificações aquando da consulta marcada?

28 respostas



Que informações pertinentes deveriam ser apresentadas no perfil de animal de estimação? Ex: foto, peso, etc...

28 respostas

Nome, idade, peso, raça, vacinas já administradas, problemas de saúde já diagnosticados, alergias a medicação

Datas de Vacinação

Tipo de animal, raça

Foto, peso, idade, histórico médico

Foto , peso , raça, observações adicionais , problemas saúde ( cego de um olho , sem uma pata..)

Tipo de animal (ex. Cão, gato, peixe), nome, foto, peso, raça, doenças associadas, vacinas, data de nascimento, possivelmente o comportamento do animal (ex. Dócil, agressivo)

Foto, peso, idade, raça, doenças diagnosticadas, alergias, nome, vacinas tomadas (Vacina da raiva p exemplo)

Idade, validade das vacinas, sugestões de higiene (desparasitação, etc...), ter acesso a exames anteriores

Altura, idade, alimentação

Jwjwjsjw

Idade

Foto, peso, raça, porte

Foto do animal e todo o que seja dados do animal como o peso, a idade etc

raça, foto, peso, sexo.

Raça, peso, boletim vacinação

Foto, peso, etc...

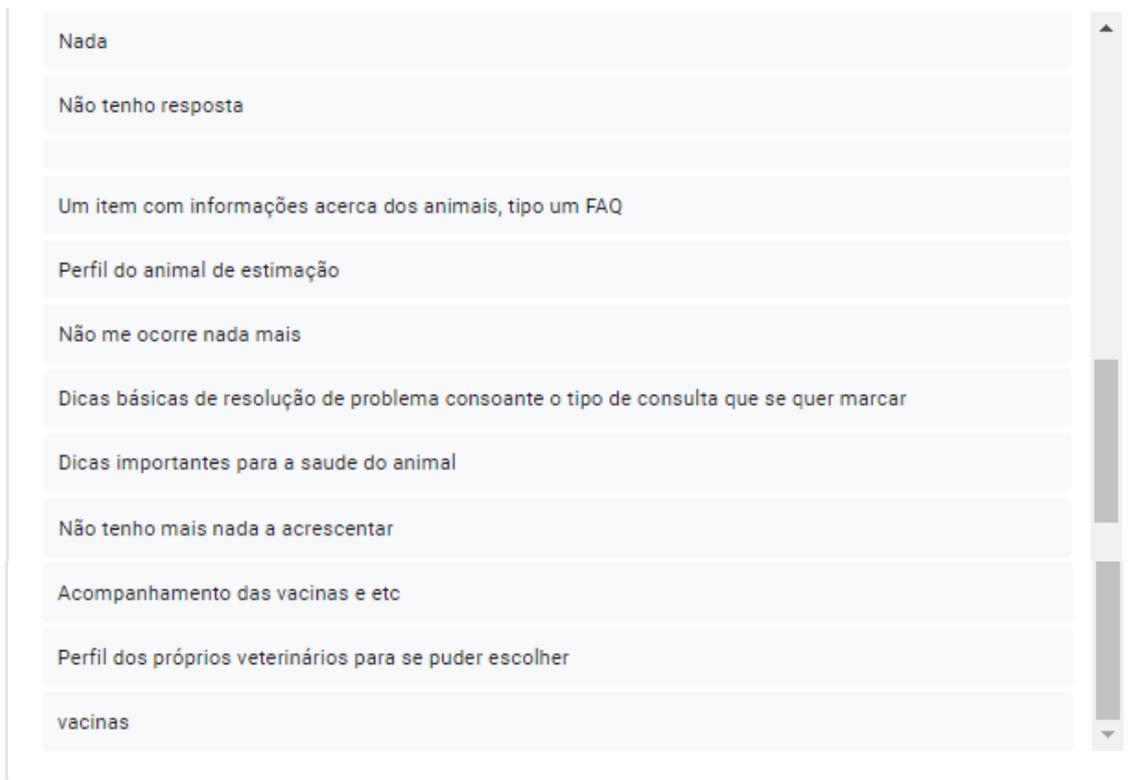
Peso

Foto, peso, idade
Tudo acerca do animal e informação relacionada com o dono
Foto e peso
Fotografia, tipo de raça, nome, peso
Raça, local de origem (de onde foi adotado ou comprado), outros problemas de saúde cujo conhecimento pode ser importante
Registo de doenças e porção de alimentação adequada ao peso do animal
peso, foto, problemas de saúde, vacinas tomadas, procedimentos que foram realizados anteriormente
Nome, peso, raça, foto, tamanho
Ficha médica
nome, raça, foto, peso, altura, vacinas

Além do que já foi questionado, o que adicionava num *website/app* para gerenciamento de consultas e acompanhamento de um animal de estimação?

28 respostas

Pagamento da consulta via mbway ou paypal
Receber notificações sobre as datas de vacinação (por exemplo: se estiver perto da data)
Já apresenta tudo o que considero necessário.
nada
Sim
-Chat privado com o médico responsável pelo meu animal. -Poder encomendar rações específicas de veterinários.
Informações sobre o faturamento das consultas, relatórios/exames do animal.
Wsss
Possibilidade de um chat com o doutor em questão, para o cliente poder dar um feedback ao longo do acompanhamento do animal.
.
Histórico de consultas
um resumo da consulta, que pode ser acedido na app após a consulta acabar.
N



## Anexo 3 – Guião de tarefas

# TESTES DE USABILIDADE

### *Instruções*



O objetivo desta bateria de testes é avaliar o nosso TFC denominado de “VetOn”, para que nos possa ajudar a detetar possíveis erros que este apresente.

*Relembramos que não está a ser testado(a), mas sim a ajudar-nos a melhorar o nosso projeto.*

Quando der por concluída a bateria de testes ser-lhe-á fornecido um breve questionário para que possamos obter o seu feedback.

Relembramos que o teste é anónimo e confidencial.

O objetivo do presente teste é verificar funcionalidades apresentadas pelo projeto.

Para tal siga os passos que lhe fornecemos .

**Nota : Visto que o site é hospedado com o plano free da Vercel as funcionalidades poderão demorar um pouco mais do que o habitual.**

**Por favor, aguarde ao longo dos passos**

1. Abra a seguinte ligação no browser: <https://vet-on.vercel.app/>
2. Passos a seguir:
  - a) Após abrir a ligação ser-lhe-á apresentada a página inicial do nosso projeto.
  - b) Dirija-se para a secção de “Contactos”
    - i) Envie uma mensagem preenchendo o formulário
    - ii) Certifique-se que foi apresentado no ecrã um pop-up com a mensagem “Email enviado com sucesso” e que os campos do formulário foram limpos

**c) Realize o registo no website:**

- i)** Clique no botão **“Login/ Sign up”**:
- ii)** De seguida, clique no link **“Registe-se aqui”** e preencha o formulário que lhe é fornecido
- iii)** Ao indicar a password tem de cumprir os requisitos que lhe são impostos, sendo que à medida que vai escrevendo irá sendo informado se está a cumprir ou não com o que lhe é solicitado
- iv)** Ao clicar no botão **“Registar”** a sua conta é automaticamente criada e é redirecionado para a página principal do website tendo acesso a todas as funcionalidades

**d) Alteração do username:**

- i)** No canto superior direito do seu ecrã tem uma aba onde diz **“Bem vindo, (nome do utilizador)”** clique nessa aba e de seguida **“Definições da conta”**
- ii)** Preencha o formulário relativo ao username e submeta clicando no botão **“Alterar username”**, certifique-se que é emitida a seguinte mensagem **“Username alterado com sucesso”**
- iii)** Clique no logo da **“VetOn”** de modo a ser redirecionado para o seu **Perfil**. Verifique que o seu username já foi atualizado.

**e) Alteração da Password:**

- i)** No seu Perfil clique no ícone de editar ao lado do seu username.
- ii)** Preencha o formulário relativo à alteração da password preenchendo todos os campos e introduzindo a **password atual incorretamente**. Por fim clique no botão **“Alterar password”**. É alertado que a password atual está incorreta
- iii)** Preencha o formulário relativo à alteração da password preenchendo todos os campos e introduzindo a password atual corretamente. Por fim clique no botão **“Alterar password”**. Recebe um alerta a dizer que a password foi alterada com sucesso
- iv)** Clique no logo da **“VetOn”** de modo a ser redirecionado para o seu **Perfil**.

**f) Adicionar animal:**

- i) No **Perfil**, clique no botão “**Adicionar animal**”
- ii) Preencha o formulário que lhe é apresentado
- iii) Pode verificar que ao submeter o formulário é redirecionado para a página de perfil e já lhe aparecem todas as informações relativas ao animal

**g) Altere peso do animal:**

- i) No seu **Perfil** clique no botão editar relativo ao animal que registou
- ii) Altere o campo relativo ao peso do animal
- iii) Ao submeter as alterações pode verificar que os dados relativos ao animal que são apresentados já se encontram atualizados

**h) Marcação de uma consulta:**

- i) No **Perfil** clique no botão “**Marcar Consulta**”
- ii) Preencha o formulário, agendando uma consulta para um dia à sua escolha
- iii) Como pode verificar já lhe aparecem todas as informações relativas a esta, bem como a opção de a desmarcar, caso surja algum imprevisto
- iv) Verifique no seu email se recebeu a confirmação do agendamento da consulta (o email pode levar algum tempo a ser enviado. Continue os passos seguintes)

**i) Desmarcar consulta:**

- i) No **Perfil** desmarque a consulta que tinha previamente agendado
- ii) Pode verificar que a consulta deixa de ser apresentada na página de perfil uma vez que foi desmarcada, e também é enviado um email alertando a desmarcação da mesma (o email pode levar algum tempo a ser enviado. Continue os passos seguintes)

**j) Logout**

- i) No **Perfil** clique no botão “**Logout**”
- ii) É redirecionado para a página principal do website sendo a sua sessão automaticamente encerrada

**k) Reset da password:**

- i) Clique no botão **“Login/ Sign up”**
- ii) De seguida clique no link **“Redefina aqui”**
- iii) Preencha o formulário introduzindo o email utilizado no ato do registo
- iv) É enviado um email para o email introduzido, abra o email e siga as instruções que lhe são fornecidas
- v) Após clicar no link fornecido é redirecionado para uma página de reset da password onde deve preencher todos os campos solicitados
- vi) Ao submeter o formulário é informado que a sua password foi alterada com sucesso
- vii) Clique no logo da **“VetOn”** de modo a ser redirecionado para a página principal do website

**l) Bloqueio da conta:**

- i) Clique no botão **“Login/ Sign up”**
- ii) Preencha o formulário introduzindo a password errada **3 vezes**
- iii) É alertado que a sua conta foi bloqueada

**Obrigada pela sua Colaboração!**

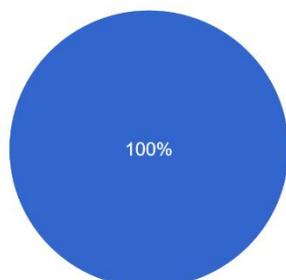
**De seguida, ser-lhe-á fornecido um questionário relativo ao projeto para que nos possa fornecer o seu feedback ao deste usufruir.**

**Link do questionário :** <https://forms.gle/V7AIA5kwuj7tfwNj9>

## Anexo 4 – Resultados dos testes

Envio de mensagem na secção Contactos

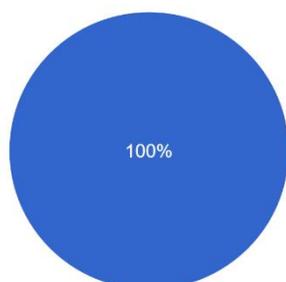
15 respostas



- Consegui enviar a mensagem
- Não consegui enviar a mensagem

Registo do Utilizador:

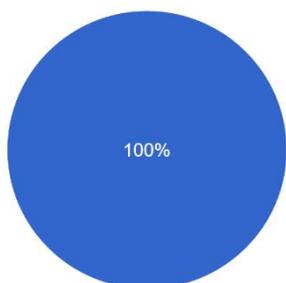
15 respostas



- Consegui registar-me
- Não consegui registar-me

Edição dados pessoais:

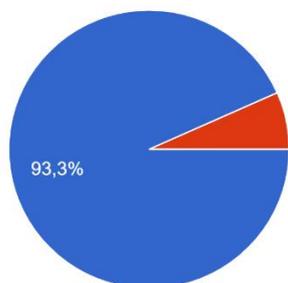
15 respostas



- Consegui editar os meus dados (username e password)
- Não consegui editar os meus dados

### Adição do animal:

15 respostas



- Consegui adicionar o animal
- Não consegui adicionar o animal

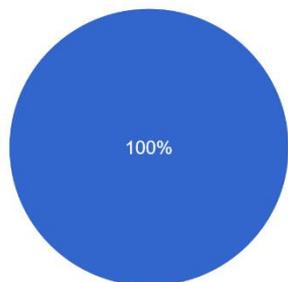
### Se respondeu que não, porque é que não conseguiu adicionar o animal ?

1 resposta

A primeira vez que tentei nao deu mas depois meteu dois

### Alteração peso do animal:

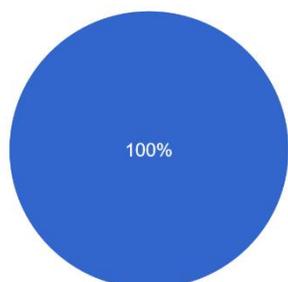
15 respostas



- Consegui alterar o peso do animal
- Não consegui alterar o peso animal

### Marcação de consultas:

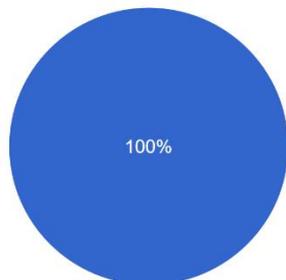
15 respostas



- Consegui agendar a consulta
- Não consegui agendar a consulta

### Reset da password

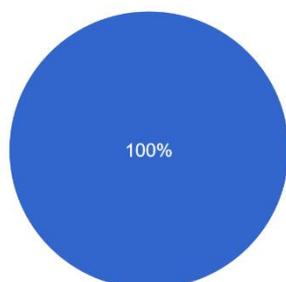
15 respostas



- Consegui realizar o reset da password
- Não consegui realizar o reset da password

### Desmarcação da Consulta

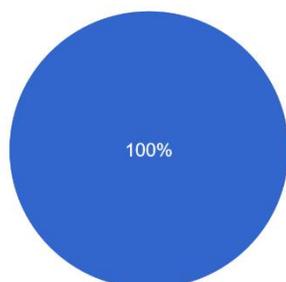
15 respostas



- Consegui desmarcar a consulta
- Não consegui desmarcar a consulta

### Realização do Logout:

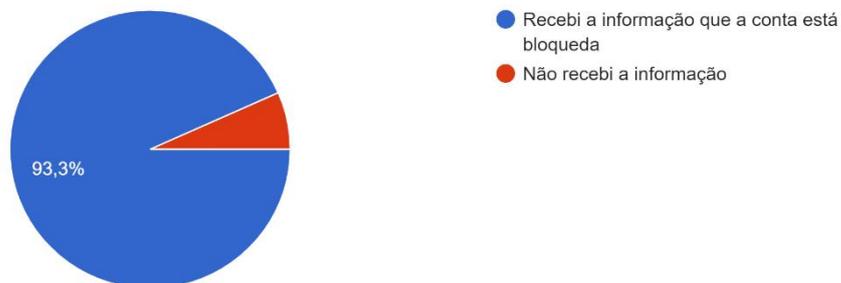
15 respostas



- Consegui realizar o logout
- Não consegui realizar o logout

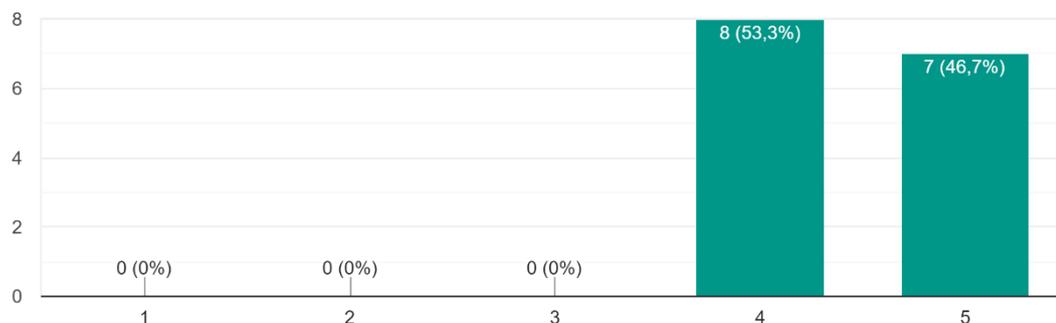
### Bloqueio da conta

15 respostas



### Feedback geral do website

15 respostas



### Sugestões / Opiniões:

7 respostas

Website bastante completo e intuitivo

Na reset da password, impedir que seja introduzida a mesma password

Site interessante. Ter atenção as mensagens que aparecem porque as vezes saem muito rápido.

A mensagem da conta bloqueada desaparece muito rápido e está tudo muito junto

Podiam meter detalhes com o histórico do animal (consultas feitas, medicação, etc)

Aplicação funcional com bastante utilidade.

Nem todos os animais possuem pelagem logo essa pergunta na criação do seu animal poderia ser opcional, a não ser que o veterinário seja só para cães e gatos. Dependendo do animal escolhido as seguintes informações poderiam ser específicas. Por exemplo, ninguém sabe a raça da tartaruga que tem. De resto está um excelente site com bastante utilidade e potencial.

## Progresso de Trabalho

	Entrega Intercalar 1º Semestre	Entrega Intermédia 1º Semestre	Entrega Intercalar 2º Semestre	Entrega Final (1º época)
RNF1		✓		
RNF2		✓		
RNF3			✓	
RNF4			✓	
RNF5		✓		
RNF6		✓		
RNF7			✓	
RNF8			✓	

**Tabela 3 - Tabela de Requisitos Não Funcionais**

	Entrega Intercalar 1º Semestre	Entrega Intermédia 1º Semestre	Entrega Intercalar 2º Semestre	Entrega Final (1º época)
RF1			✓	
RF2			✓	
RF3			✓	
RF4			✓	
RF5				✓
RF6				✓
RF7		✓		
RF8		✓		
RF9			✓	
RF10			✓	
RF11				✓
RF12			✓	
RF13				
RF14			✓	
RF15			✓	
RF16			✓	
RF17			✓	
RF18			✓	
RF19			✓	
RF20				✓

**Tabela 4 - Tabela de Requisitos Funcionais**

	Entrega Intercalar 1º Semestre	Entrega Intermédia 1º Semestre	Entrega Intercalar 2º Semestre	Entrega Final (1º época)
RT1		✓		
RT2		✓		
RT3		✓		
RT4		✓		
RT5		✓		
RT6		✓		
RT7		✓		
RT8				✓

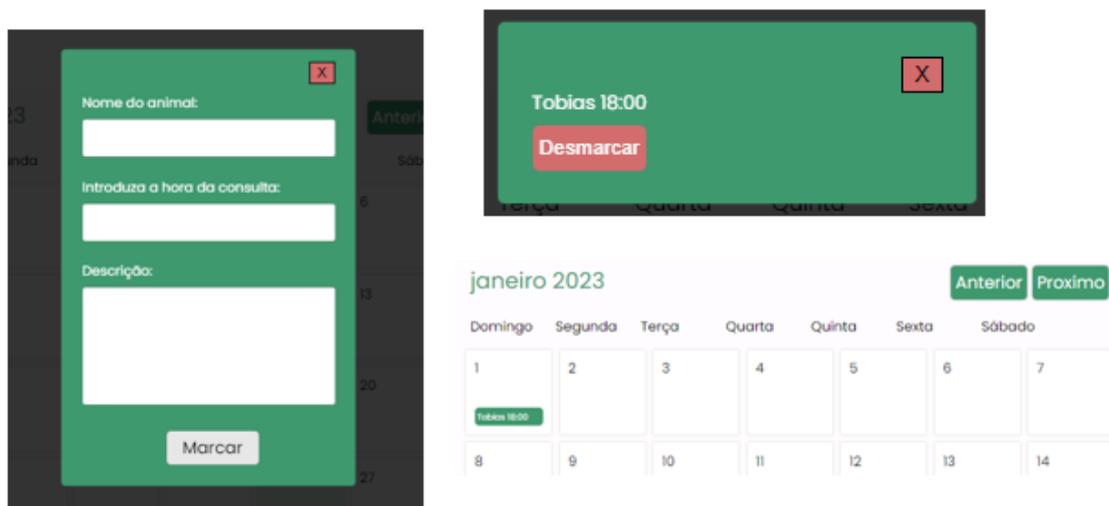
**Tabela 5 – Tabela de Requisitos Técnicos**

## Update 26 janeiro:

À data de 26 janeiro de 2023 a solução encontra-se maioritariamente desenvolvida no que toca ao *Front-end* bem como já com algum *Back-end* desenvolvido.

Em relação aos requisitos que projetamos inicialmente, foram desenvolvidos os seguintes:

- RNF3



**Figura 22 – Marcação e desmarcação de consultas**

O processo de marcação de consulta é realizado por meio de um formulário ao qual o utilizador tem acesso após clicar no calendário no dia pretendido. Caso pretenda desmarcar a consulta basta clicar no dia que a agendou e ser-lhe-á fornecido um pop-up onde ao clicar no botão desmarcar esta é automaticamente cancelada.

- RT6, RNF5, RT8, RF7, RF8



Figura 23 – Formulários de autenticação

Ao aceder ao website o utilizador tem acesso a formulários de registo e login.

- RT7

```
router.post("/", async (req, res) => {
  const email = req.body.email
  const username = req.body.username
  const password = req.body.password
  const hashedPassword = await bcrypt.hash(password, 10)

  const user = {
    email: email,
    username: username,
    password: hashedPassword
  }

  try {
    await User.create(user)
    res.render("login")
  } catch (error) {
    res.status(500).json({ error: error })
  }
})
```

Figura 24 – Código encriptação de password

Como dito anteriormente, o método de criptografia do tipo *hash* para senhas baseado na cifra *Blowfish*. Todas as *passwords* introduzidas pelos utilizadores são encriptadas por este método e posteriormente armazenadas na nossa BD.

- RT2

```
const mongoose = require("mongoose")
mongoose.connect(uri, { useNewUrlParser: true })
const db = mongoose.connection
db.on("error", error => console.error(error))
db.once("open", () => console.log("Connected"))
```

Figura 25 – Código para a conexão à base de dados MongoDB

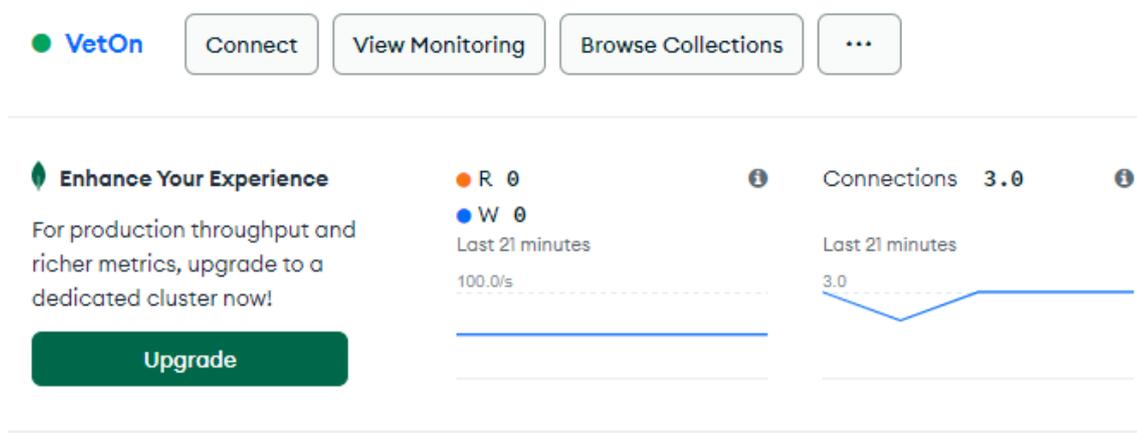


Figura 26 – Mongo Atlas

Para fazer a ligação à base dados MongoDB utilizamos a framework mongoose. Para podermos visualizar as coleções da nossa BD (visto que esta é não relacional), utilizamos o Mongo Atlas.

#### Feedback geral:

Numa primeira fase tivemos dificuldades na escolha das tecnologias a serem utilizadas para o desenvolvimento da aplicação.

No decorrer do desenvolvimento do projeto apenas sentimos dificuldades na conciliação do tempo necessário para a realização do TFC e o resto das UCs, porque em termos de capacidades técnicas (desenvolvimento da aplicação e conhecimentos de arquitetura) não tivemos dificuldades.

#### Update 23 abril:

Tendo em conta o feedback relativo à avaliação intermédia do 1º semestre, realizamos as seguintes alterações:

- Retificação dos [Requisitos](#)

- Alterações de alguns [diagramas](#) face ao desenvolvimento do projeto
- Atualização do tópico referente à [Arquitetura](#)
- Atualização do tópico [Tecnologias e ferramentas utilizadas](#)
- Maior detalhe no que toca à [Implementação](#)
- Atualização do tópico [Abrangência](#)

### **Retificação dos Requisitos**

Atualização dos requisitos, devido ao feedback proveniente da avaliação intermédia

Começamos por verificar que alguns requisitos se encontravam mal definidos, deste modo realizamos atualizações em todos os requisitos de forma a obtermos uma melhor descrição das funcionalidades que a aplicação apresenta, mas também para não existir repetição de informações.

#### **Adicionamos 8 RF :**

**RF12.** A aplicação deve ser capaz de gravar as conversas do Live-Chat e armazená-las para referência futura. (**nice to have / L**)

**RF13.** A aplicação deve permitir o envio de fotos e mensagens através do Live-Chat entre o utilizador e o customer support (**nice to have / L**)

**RF14.** Ao fazer login, o utilizador deverá ter acesso ao Live-Chat sem necessidade de mais verificações (**should have / S**)

**RF15.** O utilizador deve conseguir registar os seus animais e estes aparecerem na sua área pessoal. (**must have / S**)

**RF16.** O utilizador deve ser capaz de editar e eliminar os seus animais (**must have / M**)

**RF17.** O admin deverá poder marcar, editar e desmarcar consultas dos utilizadores a pedido dos mesmos ou do médico envolvido. (**must have / M**)

**RF18.** O admin deverá poder editar certas secções da *Landing Page* (**must have / S**)

**RF19.** O utilizador deverá conseguir entrar em contacto com o Hospital veterinário através de um formulário sem necessitar de realizar o registo / login no website (**nice to have / S**)

Englobamos os requisitos **RT4, RT5, RT6, RT7** que tínhamos anteriormente definido como RNF como RT

**RT4.** O sistema de autenticação deve proteger contra o acesso não autorizado à aplicação e aos dados dos utilizadores; (**must have**)

**RT5.** Deve ser usada criptografia forte para proteger as passwords dos utilizadores e outras informações confidenciais; (**must have**)

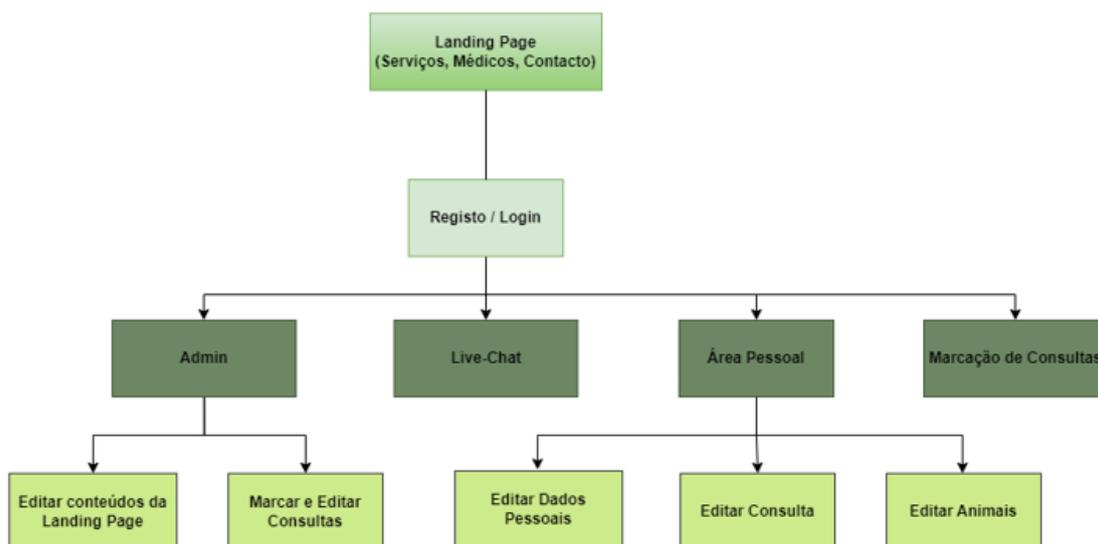
**RT6.** O sistema de autenticação deve estar sempre disponível para os utilizadores; (**must have**)

**RT7.** O sistema deve ser capaz de garantir a complexidade das passwords dos utilizadores; (**should have**)

### Atualização de Diagramas

Face ao desenvolvimento do projeto realizamos a atualização do Mapa aplicacional de forma a cumprir com a estrutura final da aplicação.

Mantivemos os 4 níveis que anteriormente havíamos definido, mas realizamos alterações nos níveis 2 e 3, incluído a página referente ao admin e ao live chat.



**Figura 27 – Atualização do Mapa Aplicacional**

### Atualização do Tópico referente à Arquitetura

No que toca à Arquitetura mantivemos o modelo baseado na arquitetura de 3 camadas, realizamos apenas alterações na Camada de Apresentação uma vez que alteramos as frameworks/tecnologias utilizadas no desenvolvimento do projeto.

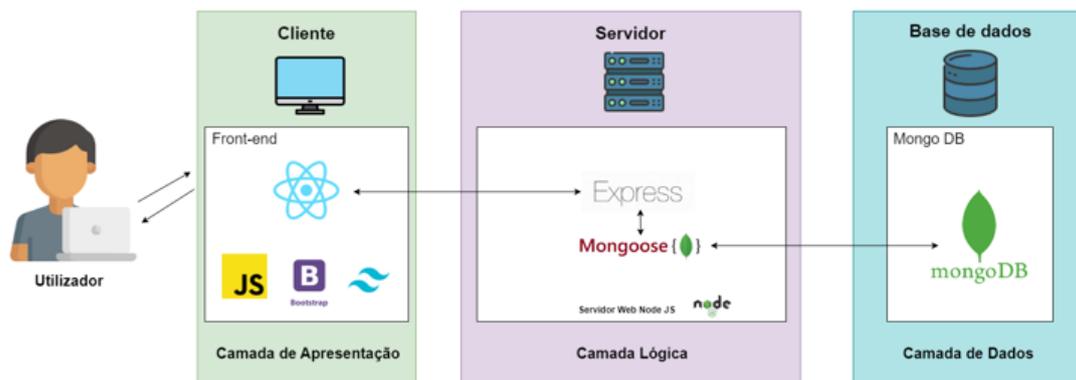


Figura 28 – Atualização do Diagrama Arquitetura

### Atualização do Tópico referente as Tecnologias e Ferramentas utilizadas

Face à entrega anterior e ao desenvolvimento do projeto realizamos algumas alterações no que toca as Tecnologias e Ferramentas utilizadas no desenvolvimento do *Front-end*.

No desenvolvimento do *Front-end* passamos a utilizar as seguintes tecnologias:

- **React**
- **Bootstrap**
- **Tailwind CSS**

### Atualização do Tópico Implementação

No que diz respeito à Implementação adicionamos o tópico referente à estrutura do website, onde realizamos a descrição de como é que este se encontra dividido (**Client e API**) e quais os ficheiros que possui em cada uma das partes.

### Atualização do Tópico Abrangência

Após o *feedback* proveniente da avaliação intermédia no que toca à abrangência, realizamos uma descrição com maior detalhe, especificando em cada caso os conhecimentos utilizados que foram abordados nas UCs.

### Feedback geral:

Em relação à entrega anterior foram realizadas diversas alterações tanto na estrutura como nas Tecnologias e Ferramentas utilizadas.

As alterações principais debruçam-se sobre:

## 1. Live-Chat

Incorporação de um chat no website de forma a permitir o contacto entre os utilizadores os hospital veterinário.

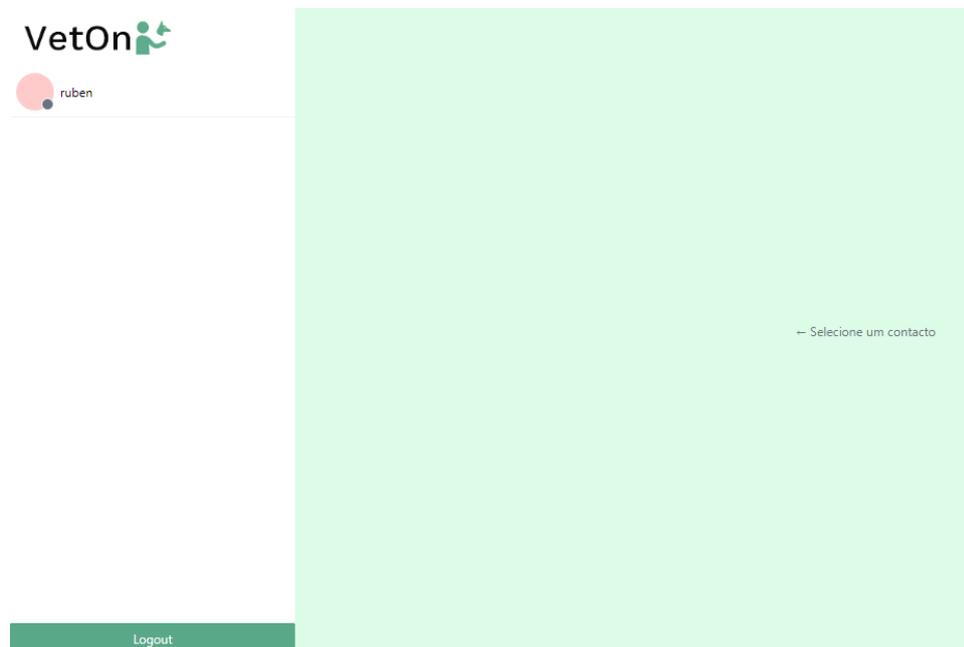


Figura 29 – Live-Chat

## 2. Marcação, desmarcação de consultas

Realizamos o processo de marcação e desmarcação de consultas, onde o utilizador deve seleccionar o animal, especialidade, o médico e o horário pretendido.

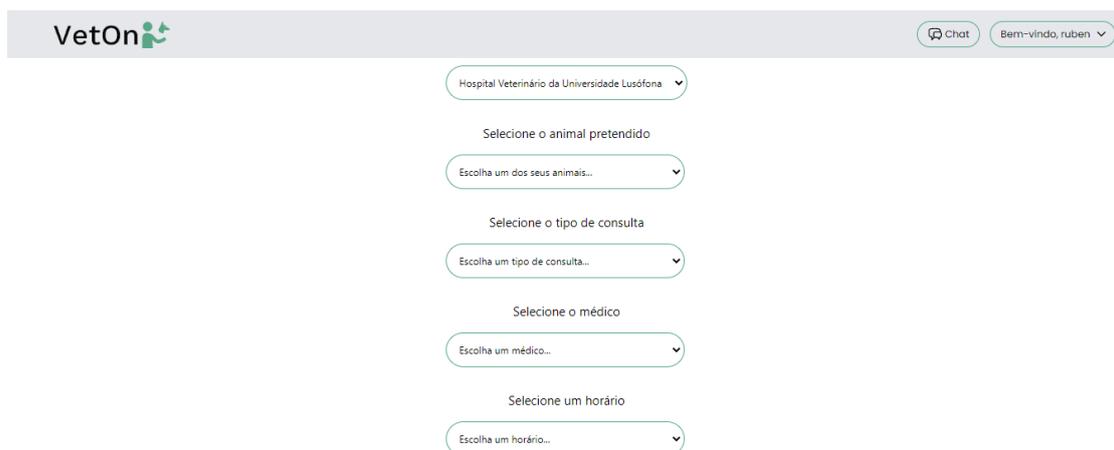


Figura 30 – Marcação e desmarcação de Consultas

### 3. Update da área pessoal

Em relação à entrega anterior realizamos uma reestruturação na página área pessoal, onde o utilizador tem acesso aos seus dados, pode adicionar e eliminar animais e verificar as consultas que possui agendadas.

Através desta página é possível realizar a edição dos dados pessoais (email, nome de utilizador e password).



**Figura 31 - Atualização área pessoal**



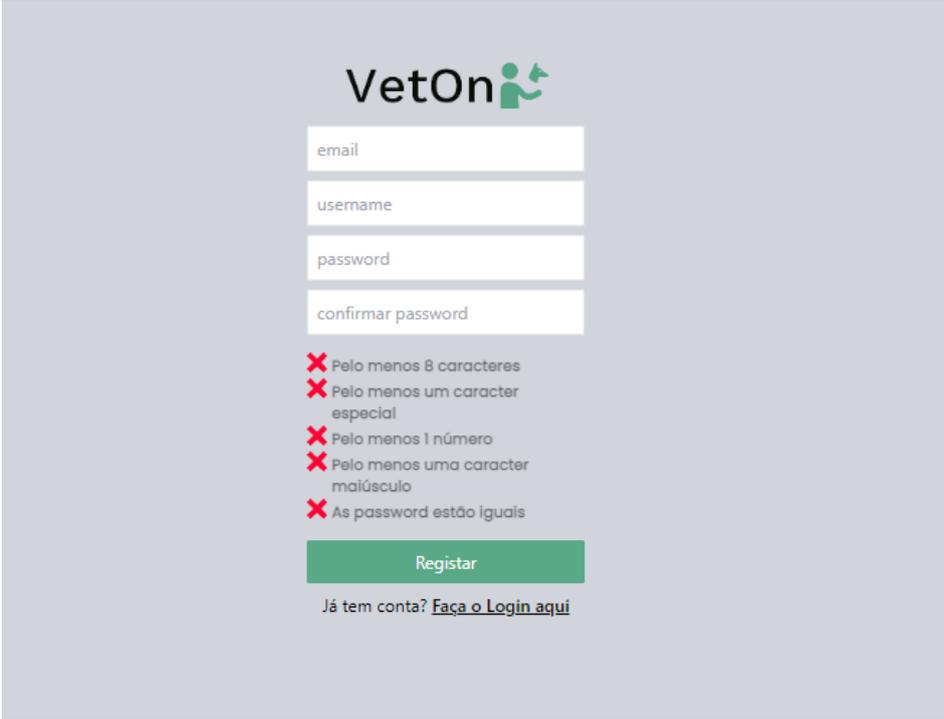
**Figura 32 – Visualização das Consultas**

#### 4. Verificação da complexidade das passwords

Uma vez que as password representam um fator crítico implementamos regras que estas devem respeitar, tanto no ato do registo como no do login

Devem respeitar as seguintes regras:

- Pelo menos 8 caracteres
- Pelo menos um caracter especial
- Pelo menos 1 número
- Pelo menos uma letra maiúscula



The screenshot shows the VetOn registration interface. At the top, the VetOn logo is displayed. Below it are four input fields: 'email', 'username', 'password', and 'confirmar password'. Under the 'password' field, there are five error messages, each preceded by a red 'X' icon:

- ✗ Pelo menos 8 caracteres
- ✗ Pelo menos um caracter especial
- ✗ Pelo menos 1 número
- ✗ Pelo menos uma caracter maiúsculo
- ✗ As password estão iguais

Below the error messages is a green 'Registrar' button. At the bottom, there is a link: 'Já tem conta? [Faça o Login aqui](#)'.

**Figura 33 – Complexidade das Passwords**

#### 5. Envio de um email ao redefinir a password

Ao realizar a redefinição da password é enviado um email para o email indicado de forma a garantir que esta está a ser redefinida pelo dono da conta

## Reset Password Caixa de entrada x



veton.verify.users@gmail.com

para mim ▾

Pediste para redefinir a password da tua conta no site da VetOn

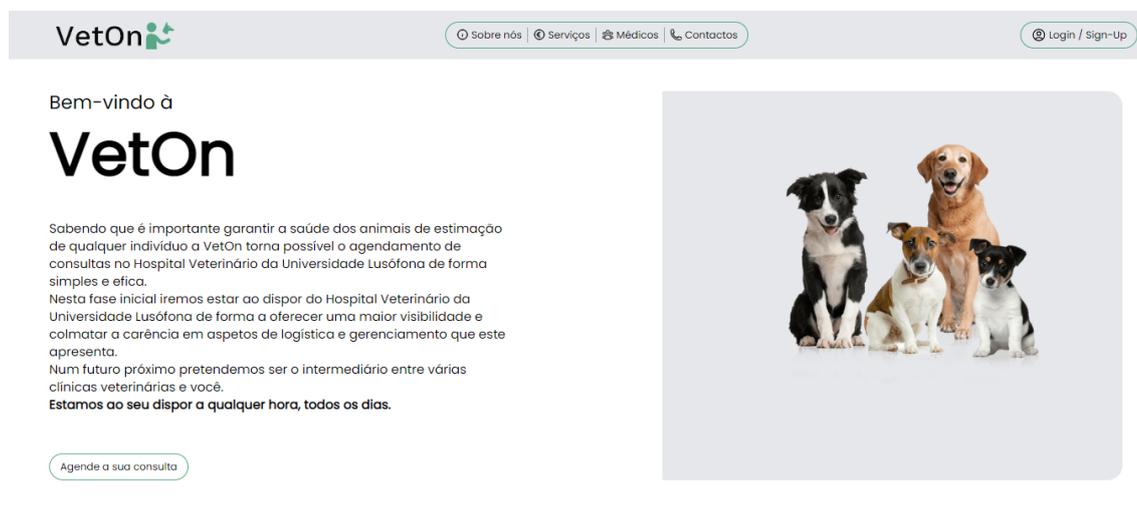
Este link expira em 6 horas

Clica aqui [link](#) para a redefinires

**Figura 34 – Email para redefinir a password**

## 6. Update da Lading Page

Alteração da nav bar e dos seus componentes bem como pequenos ajustes nas secções



**Figura 35 – Alterações na Landing Page**

## 7. Adição página para admin e support

De forma a realizar a marcação e desmarcação de consultas a pedido dos médicos, adição e eliminação de médicos e de serviços na Landing Page foi desenvolvida uma página para utilizadores do tipo admin.

Para que os utilizadores do website possam entrar em contacto com o Hospital Veterinário através do Live-Chat, foi desenvolvida uma página para utilizadores do tipo support.

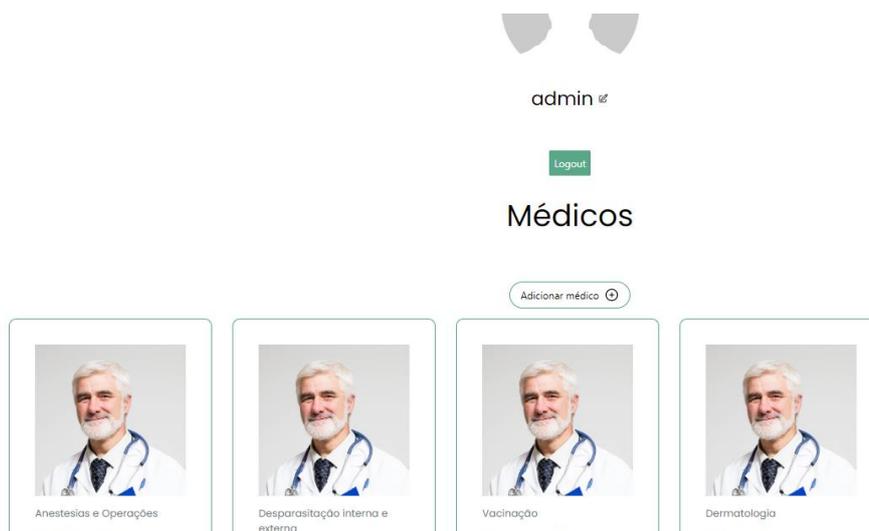
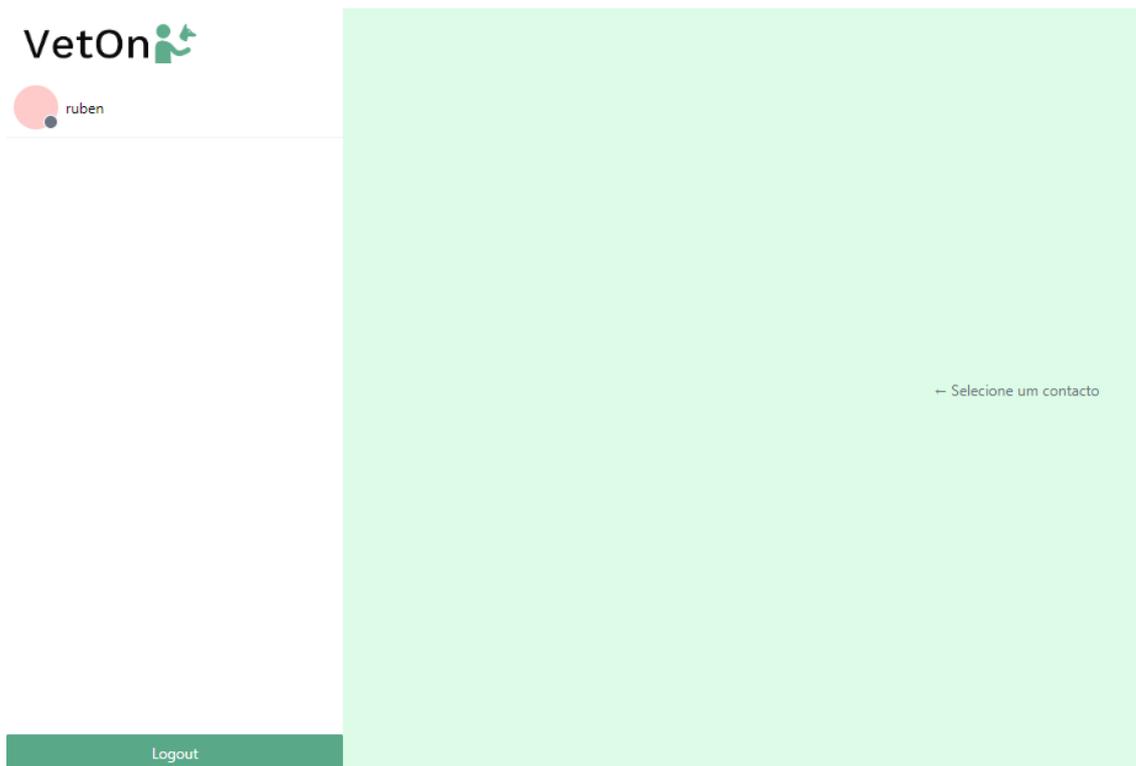


Figura 36 – Página do admin (adição médicos)



Figura 37 – Página admin (adição de serviços)



**Figura 38 – Página support (Live-Chat)**

## **8. Desenvolvimento de um formulário**

Desenvolvimento de um formulário cujo objetivo é recolher uma bateria de testes realizada ao nosso website para nos ajudar a detetar possíveis erros que este apresenta.

## **Glossário**

LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso
BD	Base de Dados
API	Application Programming Interface
RNF	Requisitos Não Funcionais
RF	Requisitos Funcionais
RT	Requisitos Técnicos
UI/UX	User Interface/User Experience