

# RELATÓRIO – CONSTRUCTION

## **Desenho e Implementação**

André Patacas, 93357  
Denis Yamunaque, 101513  
Luísa Amaral, 93001  
Pedro Loureiro, 92953  
Grupo 303

Janeiro 2020

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>5</b>
1.1	Sumário executivo . . . . .	5
1.2	Controlo de versões . . . . .	5
<b>2</b>	<b>Arquitetura do sistema</b>	<b>6</b>
2.1	Objetivos gerais . . . . .	6
2.2	Requisitos com impacto na arquitetura . . . . .	7
2.3	Decisões tomadas e fundamentação . . . . .	7
2.4	Arquitetura proposta . . . . .	8
2.4.1	Arquitetura lógica da solução . . . . .	8
2.4.2	Interação entre módulos . . . . .	9
2.4.3	Integrações com sistemas externos . . . . .	10
<b>3</b>	<b>Ambiente de desenvolvimento e incrementos</b>	<b>11</b>
3.1	Arquitetura de instalação . . . . .	11
3.2	Tecnologias de desenvolvimento . . . . .	11
3.3	Incremento desenvolvido . . . . .	11
<b>4</b>	<b>Histórias e critérios de aceitação</b>	<b>14</b>
4.1	Caracterização das Personas representativas . . . . .	14
4.2	Histórias para a 1 <sup>a</sup> iteração da Construção (Construction #1) . . . . .	15
4.3	Automação de testes de aceitação . . . . .	17
<b>5</b>	<b>Referências e recursos</b>	<b>19</b>

## **Lista de Tabelas**

1	Controlo de versões . . . . .	5
2	Requisitos com impacto na arquitetura . . . . .	7

## **Lista de Figuras**

1	Arquitetura da solução . . . . .	8
2	Diagrama de sequência - Encomendar Ração . . . . .	9
3	Diagrama de sequência - Marcar Consulta . . . . .	9
4	Diagrama de sequência - Participar em uma Consulta . . . . .	10
5	Diagrama de instalação da solução . . . . .	11
6	Fluxo de utilização da funcionalidade de pedido de consulta urgente	12
7	Fluxo de utilização da funcionalidade de publicar anúncios . . . .	12
8	Fluxo de utilização da funcionalidade de contratar um petsitter .	13
9	Fluxo de utilização da funcionalidade de compra de produtos . .	13
10	Testes criados para verificar as funcionalidades desenvolvidas . .	18

# 1 Introdução

## 1.1 Sumário executivo

Este relatório apresenta os resultados da segunda iteração da Elaboration e da fase de Construction, adaptada do método OpenUP, em que se constrói o produto ao longo de várias iterações.

Os principais requisitos com impacto na arquitetura prendem-se com a implementação de vários fluxos interdependentes que partilham informação e a necessidade do uso de vários sistemas externos como, por exemplo, uma base de dados para guardar os dados dos utilizadores e das entidades associadas à aplicação ou um sistema de autenticação.

Nesta interação, foi considerado prioritário implementar os fluxos dos casos nucleares de utilização da aplicação. Um dos principais fluxos a ser concretizado foi a encomenda de ração, visto que é essencial fornecer a possibilidade de compra de produtos vendidos pela stakeholder Royal Canine aos utilizadores, de forma a revelar as vantagens económicas que este produto traz ao stakeholder.

## 1.2 Controlo de versões

Quando?	Responsável	Alterações significativas
4/01	Todos	Conceptualização da ideia e escrita da primeira versão da construção do produto
—	André	Elaboração do produto ( <a href="#">Link para Github</a> )
13/01	Pedro	Elaboração do Capítulo 2 do relatório - Arquitetura do Sistema
16/01	Luísa	Adição de personas e user stories
16/01	Denis	Elaboração do Capítulo 3 do relatório - Ambiente de desenvolvimento e incrementos
18/01	Luísa	Adição da secção de testes de automação e finalização da introdução

Tabela 1: Controlo de versões

## 2 Arquitetura do sistema

### 2.1 Objetivos gerais

Os objetivos da arquitetura do nosso projeto estão ligados à gestão de recursos humanos, como *Pet Sitters*, utilizadores e os associados (Clínicas Veterinárias), à coordenação do processo de realização de encomendas, à gestão de anúncios de animais perdidos, e por fim, à organização do processo de marcação das consultas tanto normais como urgentes. Para isto foi necessário recorrer a uma base de dados, de forma a guardar os dados tanto de produtos como de pessoas para viabilizar toda a lógica do sistema.

O sistema está integrado com plataformas externas de forma a potenciar a utilização da aplicação, tais como: base de dados externa para guardar dados da aplicação que engloba os produtos disponíveis, lista de *Pet Sitters*, anúncios disponíveis, informação sobre os utilizadores e os seus animais e ainda guarda dados relativos às clínicas veterinárias parceiras. Na parte de autenticação de um utilizador, o sistema utiliza a plataforma *Firebase* que permite o uso de contas Google, Facebook, etc para agilizar a criação de uma conta e posteriormente o login. Outro sistema externo usado em conjunto com a nossa arquitetura é uma plataforma de videochamada para a realização das consultas, que será usada pelo utilizador e pela Clínica Veterinária. A responsabilidade de gerir encomendas (por entregar) é da empresa de distribuição, a qual partilha dados com a aplicação sobre tempo estimado, número de encomenda, entre outros. Também é usado um sistema externo de notificações, *OneSignal*, que trata a parte do envio de mensagens, emails, ou notificações Push para o utilizador que incluem alertas de consultas, encomendas, *Pet Sitters* e anúncios de animais perdidos. Por último, no domínio do pagamento, foram incorporadas as atuais tecnologias mais usadas como PayPal, MBWay e transferência direta o que vai assegurar a segurança dos pagamentos reforçando a confiança dos utilizadores. O sistema é oferecido para *Smartphones* com sistemas operativos Android e iOS, de modo que os utilizadores destes dispositivos apenas tenham que descarregar a aplicação na respetiva loja de aplicativos móveis.

## 2.2 Requisitos com impacto na arquitetura

Requisitos	Descrição
RI-2	Ativar ou desativar as notificações de datas de consultas
RI-4	Acompanhar o progresso de uma encomenda
RD-1	As transações financeiras embutidas na aplicação gastam menos que 1 minuto
RD-2	As pesquisas de <i>Pet Sitter</i> , raça, médicos disponíveis e lista de animais perdidos devem retornar um resultado visivelmente instantâneo (0.1 segundos ou menos de demora).
RD-3	A conexão com o médico veterinário para consulta de emergência deve retornar resultado no máximo em 10 segundos.
RD-4	Os servidores devem poder suportar o volume de acessos projetado pela estimativa de usuários e prestadores de serviços.
RS-1	Controle de acesso por credenciais de usuário

Tabela 2: Requisitos com impacto na arquitetura

## 2.3 Decisões tomadas e fundamentação

Tendo em conta os objetivos para a arquitetura, e os requisitos levantados na Análise, foram tomadas várias decisões. Uma destas decisões foi usar JavaScript no que toca a Frontend por diversas razões. A principal razão pela qual esta linguagem foi escolhida deve-se à sua adequação ao problema, contém uma grande variedade de funcionalidades e disponibiliza várias frameworks, como o React Native com Expo, que agilizam o desenvolvimento de apps de informação.

A decisão de usar sistemas externos para complementar o nosso projeto, tornou-o mais simples e eficiente dado que as soluções existentes no mercado, como base de dados e sistema de notificações, já foram amplamente testadas e aprovadas, o que as torna numa fonte segura.

## 2.4 Arquitetura proposta

### 2.4.1 Arquitetura lógica da solução

A arquitetura lógica do sistema foi dividida em três partes: UI, Lógica do Domínio e Infraestrutura. A camada superior, a UI, corresponde à camada que os utilizadores vêem e com a qual interagem diretamente.

Na camada intermédia estão contidos diversos mecanismos necessários ao bom funcionamento da aplicação. Um destes mecanismos é a realização de encomendas: aqui é descrito o que vai acontecer quando um utilizador encomenda ração, a informação enviada para os armazéns da Royal Canin, a informação enviada para os sistemas de pagamento, a informação enviada para o sistema de notificações relativamente a quando notificar o utilizador que a encomenda vai ser entregue, a atualização do perfil do utilizador, etc. Além deste tipo de operações que são efetuadas pelo utilizador, como publicação de anúncios e contratação de *Pet Sitters*, esta camada também engloba a gestão de diversos recursos associados com a aplicação como Clínicas Veterinárias, utilizadores da aplicação e gestão de consultas.

Na camada mais inferior, a Infraestrutura é responsável por guardar os dados da aplicação, desde produtos em venda, *Pet Sitters* listados, Clínicas Veterinárias associadas, horários das consultas marcadas, entre outros listados na Figura 1. Os restantes dados, nomeadamente dados de utilizadores, estão guardados no sistema externo Firebase, que serve para reforçar a proteção de dados dos utilizadores da aplicação.

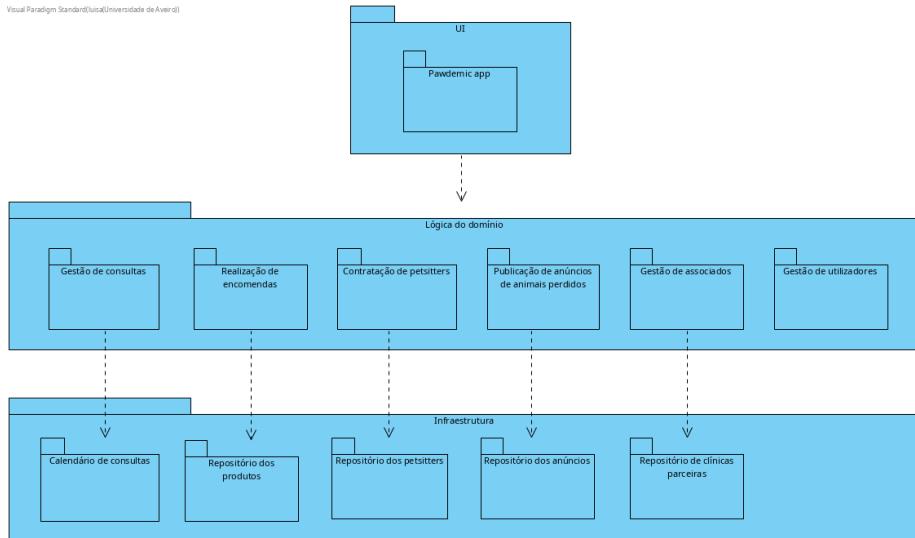


Figura 1: Arquitetura da solução

### 2.4.2 Interação entre módulos

As seguintes imagens demonstram, através de diagramas de sequência, as interações que os módulos efetuam entre si em dois casos nucleares da nossa arquitetura: Encomendar Ração e Marcar Consulta.

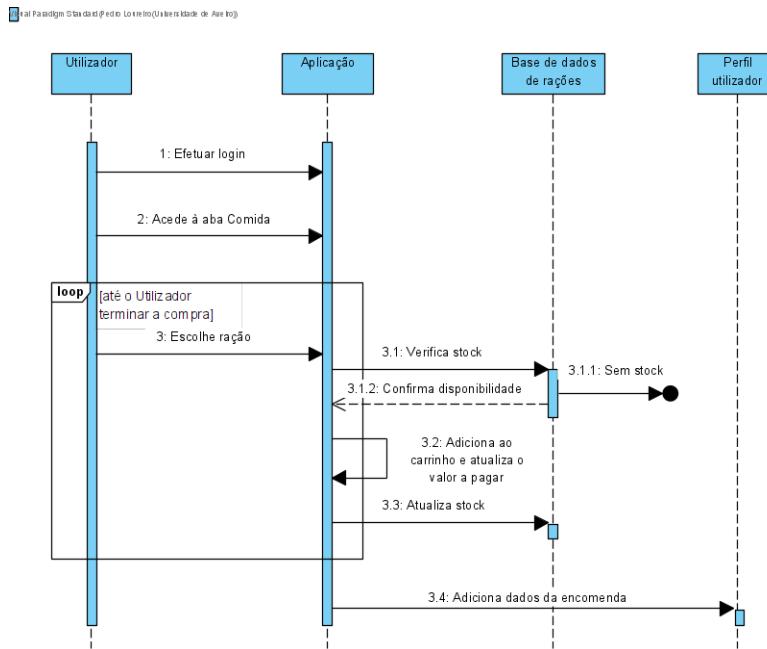


Figura 2: Diagrama de sequência - Encomendar Ração

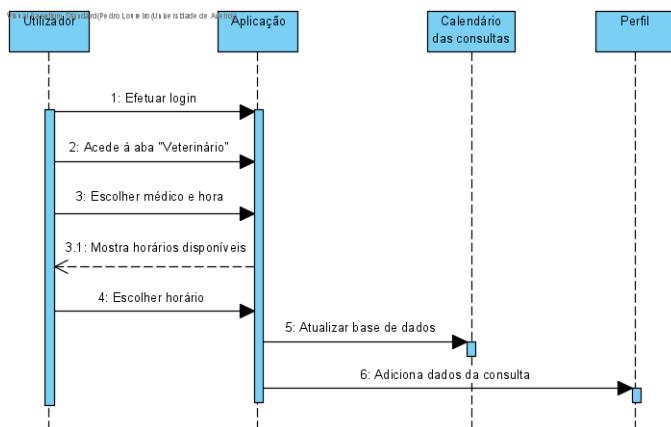


Figura 3: Diagrama de sequência - Marcar Consulta

### 2.4.3 Interações com sistemas externos

Nesta subsecção foi adicionado um diagrama de sequência para demonstrar a comunicação que a nossa aplicação tem com os diversos sistemas externos previamente apresentados. O exemplo aqui descrito é 'Participar numa Consulta' já que é possível observar o maior número de interações com sistemas externos relativamente a outros casos de utilização.

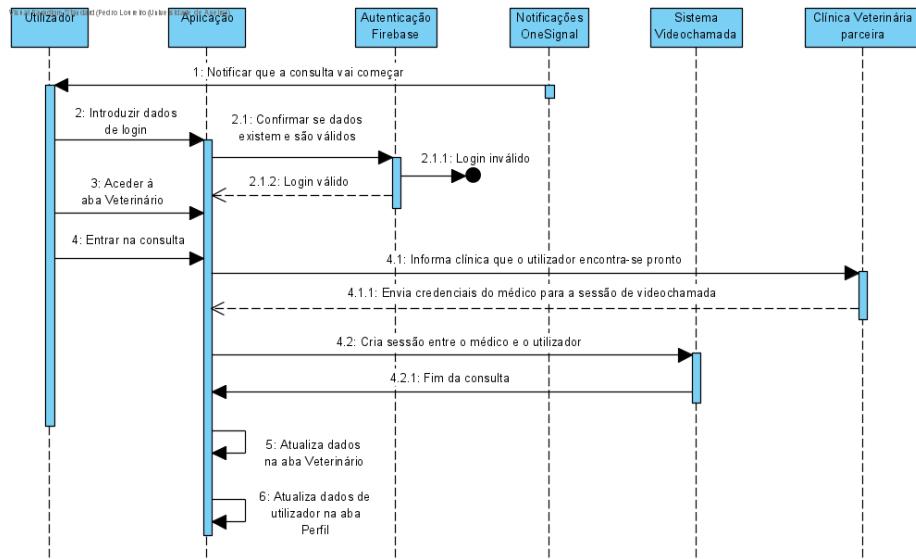


Figura 4: Diagrama de sequência - Participar em uma Consulta

### 3 Ambiente de desenvolvimento e incrementos

#### 3.1 Arquitetura de instalação

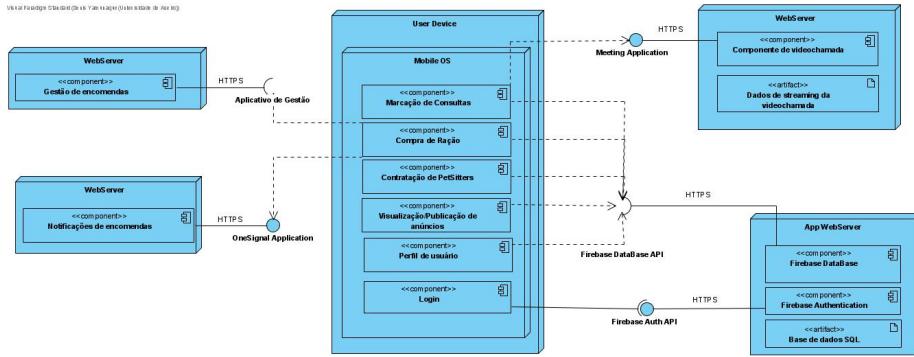


Figura 5: Diagrama de instalação da solução

Como pode ser visto no diagrama de instalação, na Figura 5, a actual iteração do produto correrá em sistema Android e iOS de equipamentos móveis. Para alcançar as funcionalidades pretendidas, a comunicação com sistemas externos torna-se essencial. Utilizaram-se na solução a aplicação OneSignal para notificação de encomendas e a plataforma Firebase como serviço de autenticação e base de dados.

#### 3.2 Tecnologias de desenvolvimento

Na versão atual da solução proposta, a linguagem escolhida foi javascript, utilizando o framework React Native com Expo, que permite uma série de funcionalidades que dão suporte ao comportamento esperado da aplicação. Foram feitos testes com o protótipo funcional da aplicação com o objectivo de verificar em diversos aspectos a usabilidade do produto e de validar a arquitectura da solução. O ficheiro de testes foi criado com a ferramenta *Selenium IDE* e encontra-se no [repositório Git](#) usado para o desenvolvimento do produto .

#### 3.3 Incremento desenvolvido

Neste capítulo são apresentados alguns screens que mostram vários passos para executar os principais fluxos de utilização da aplicação.

Está disponibilizado no micro-site um apk para instalação da app em android. Para além disso através do link <https://expo.io/@unn4m3dd/projects/pawdemic> a aplicação pode ser instalada em android ou iOS pela aplicação Expo, que atua como middleware já que ambas as lojas (play store e app store) recusaram a publicação da app por "parecer conter inApp purchases" não mediadas pelas mesmas.

O código do produto desenvolvido encontra-se no repositório Git do projeto [\(\)](#).

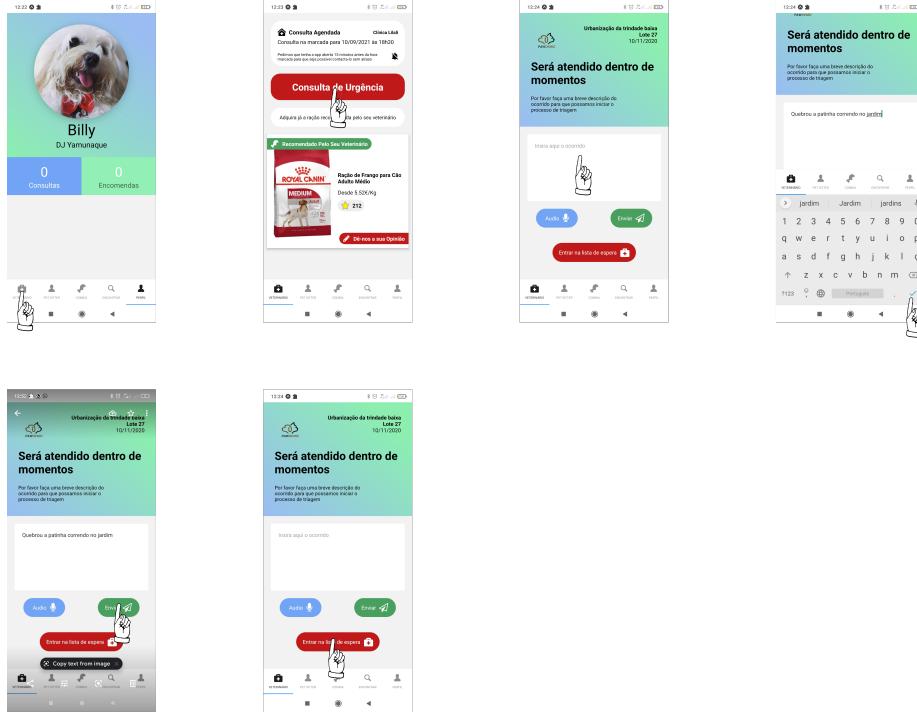


Figura 6: Fluxo de utilização da funcionalidade de pedido de consulta urgente

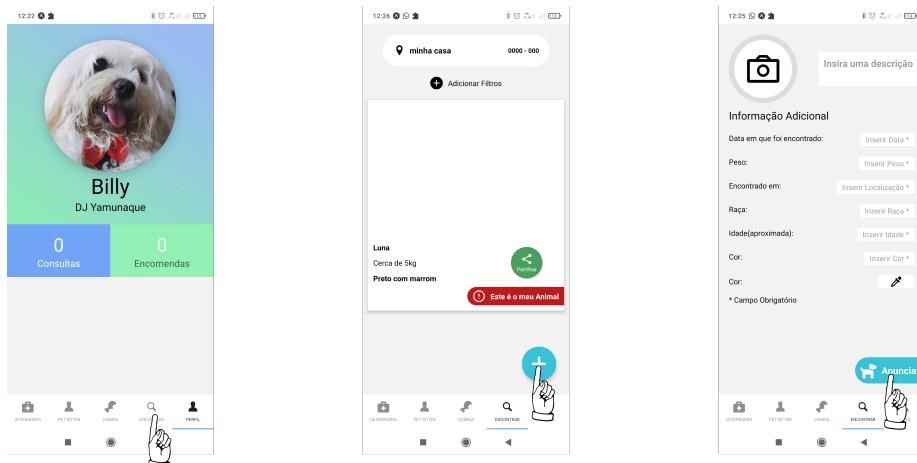


Figura 7: Fluxo de utilização da funcionalidade de publicar anúncios

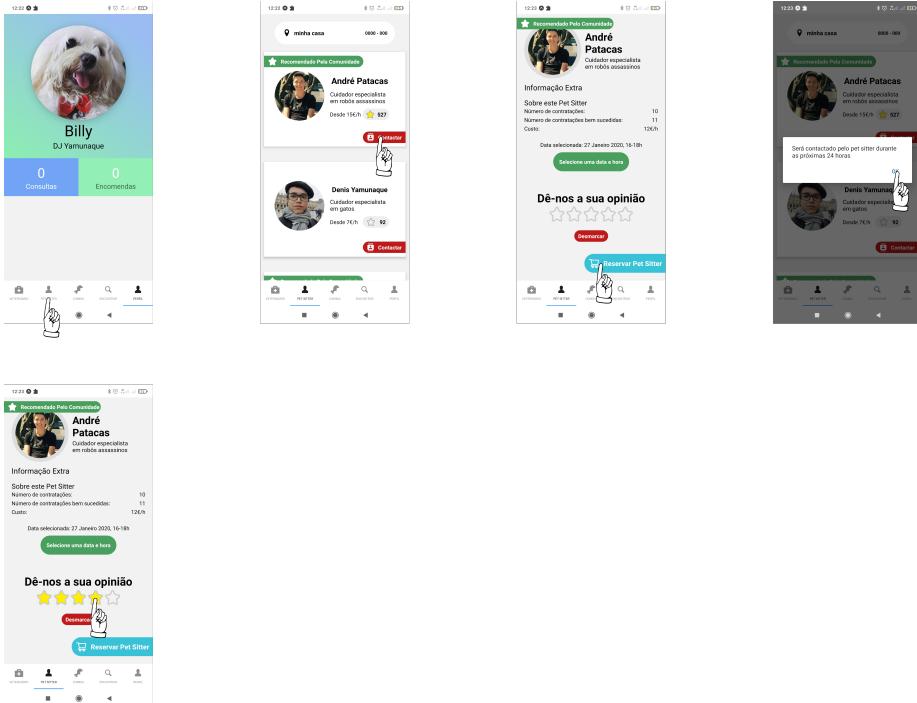


Figura 8: Fluxo de utilização da funcionalidade de contratar um petsitter

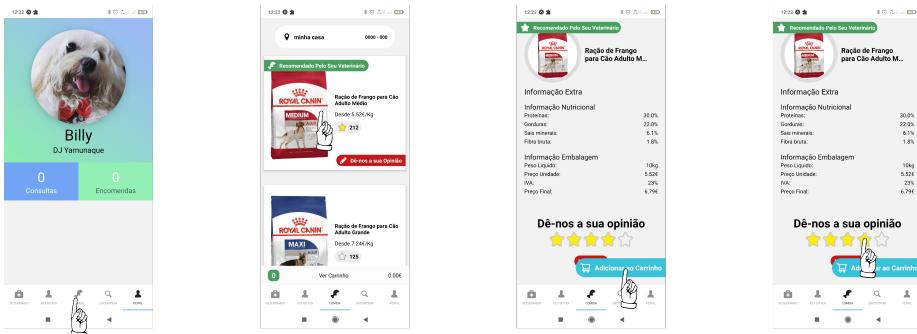


Figura 9: Fluxo de utilização da funcionalidade de compra de produtos

## **4 Histórias e critérios de aceitação**

### **4.1 Caraterização das Personas representativas**

#### **Persona 1: Artur**

O Artur trabalha como developer numa empresa, e face ao contexto da pandemia, encontra-se em trabalho remoto. É dono de uma cadela chamada Fiona e desloca-se todos os mês a uma loja de animais para comprar um pacote de ração novo para dar ao seu animal. Devido às restrições de deslocamento e a preocupação face ao contexto atual da pandemia, o Artur procura novas soluções para continuar a obter a ração para a Fiona sem sair de casa.

#### **Persona 2: Mariana**

A Mariana é dona de um cão da raça Pug chamado Oliver. O Oliver tem 10 anos e sendo uma raça com vários problemas de saúde frequentes, necessita de um acompanhamento médico constante. Devido à pandemia, a Mariana receia que seja mais difícil obter ajuda caso o estado de saúde do Oliver piorar repentinamente.

#### **Persona 3: Teresa**

A Teresa é uma médica especialista em Medicina Familiar no Centro de Saúde de Aveiro. Devido à pandemia, a carga horária da Teresa no Centro de Saúde aumentou, levando-a a que passe menos horas em casa. O animal de estimação da Teresa, um cavalier chamado Darcy, fica em casa durante as horas em que a Teresa está a trabalhar, deixando-a receosa por deixar o seu animal tantas horas sozinho sem alguém para tomar conta dele.

#### **Persona 4: Marta**

A Marta é dona de dois cães e, enquanto os passeava pelo parque de Aveiro, encontrou um gato perdido. A Marta pensou que o animal pudesse ter dono e sabendo o quanto lhe custaria perder um dos seus animais de estimação, decidiu tentar arranjar maneira de encontrar o dono do gato.

#### **Persona 5: João**

O João vive em Aveiro e é dono de um gato chamado Bigodes. Recentemente, o Bigodes fugiu de casa e tem estado desaparecido durante alguns dias. O João receia que algo tenha acontecido ao Bigodes, e gostava de ter uma maneira de ver vários anúncios de gatos perdidos, com as mesmas características do Bigodes, em Aveiro.

## 4.2 Histórias para a 1<sup>a</sup> iteração da Construção (Construction #1)

### O Artur adiciona rações ao carrinho de compras

Sendo o Artur, um dono de animal, **quero** ver as rações disponíveis para venda **de modo a** encomendar uma ou várias.

#### Cenário 1: Adição ao carrinho com sucesso

Dado que tenho o log in feito  
Quando estou na aba de "Comida"  
E escolho uma ração e clico em "adicionar ao carrinho"  
Então a quantidade de artigos no carrinho incrementa um valor  
E o preço total é atualizado

#### Cenário 2: Produto não disponível

Dado que tenho o log in feito  
Quando estou na aba de "Comida"  
E escolho uma ração  
E não há stock dessa ração  
Então nas informações do produto aparece "Sem stock"  
E a quantidade de artigos no carrinho mantém-se.

### A Mariana pede uma consulta de urgência

Sendo a Mariana, uma dona de um animal propenso a necessitar de ajuda médica urgente, **quero** ter acesso a consultas de urgências **de modo a** prestar auxílio ao meu animal

#### Cenário 1: Inserir descrição do problema

Dado que tenho o log in feito  
Quando estou na aba de Veterinário  
E clico em "Consulta de Urgência"  
E insiro uma descrição do problema  
E clico em "Entrar em fila de espera"  
Então entro numa fila de espera pela consulta

#### Cenário 2: Inserir áudio a descrever o problema

Dado que tenho o log in feito  
Quando estou na aba de "Veterinário"  
E clico em "Consulta de Urgência"  
E gravo um áudio a descrever o problema

E clico em "Entrar em fila de espera"  
Então entro numa fila de espera pela consulta

### Cenário 3: Fila de espera vazia

Dado que tenho o log in feito  
Quando estou na aba de "Veterinário"  
E clico em "Consulta de Urgência"  
E insiro um áudio ou um texto a descrever o problema  
E a lista de espera está vazia  
Então sou automaticamente entendida

## A Teresa requisita um cuidador de animais

Sendo a Teresa, uma dona de um animal, **quero** requisitar os serviços de um cuidador de animais **de modo a** poder deixar o meu animal com alguém responsável que tome conta dele.

### Cenário 1: Cuidador escolhido está disponível

Dado que tenho o log in feito  
Quando estou na aba "Pet Sitter"  
E escolho um cuidador da lista  
E escolha um dia e uma hora  
Então confirmo o pedido de contratação  
E recebo uma mensagem para aguardar pela confirmação

### Cenário 2: Cuidador escolhido não está disponível

Dado que tenho o log in feito  
Quando estou na aba "Pet Sitter"  
E escolho um cuidador da lista e um dia e uma hora  
E o cuidador não está disponível  
Então é impossível confirmar a contratação  
E continuo a pesquisar outros cuidadores

## A Marta publica um anúncio

Sendo a Marta, uma dona de animais, **quero** publicar um anúncio de um animal perdido **de modo a** poder encontrar o seu dono.

### Cenário 1:

Dado que tenho o log in feito  
Quando estou na aba "Encontrar"  
e clico no botão "+"

Então adiciono os dados do animal encontrado  
E publico um anúncio

#### **O João procura um anúncio com um animal parecido com o Bigodes**

Sendo o João, um dono de um animal perdido, **quero** visualizar anúncios de animais perdidos **de modo a** poder encontrar o meu animal.

##### **Cenário 1: Anúncio específico encontrado**

Dado que tenho o log in feito  
Quando estou na aba "Encontrar"  
e percorro a lista de anúncios  
e encontro um anúncio de um animal características similares  
Então clico em "Este é o meu Animal"  
E entro em contacto com o anunciante

##### **Cenário 2: Anúncio específico não encontrado**

Dado que tenho o log in feito  
Quando estou na aba "Encontrar"  
e percorro a lista de anúncios  
e não encontro um anúncio com um animal com as mesmas características  
que o meu  
Então a pesquisa não foi bem sucedida

### **4.3 Automação de testes de aceitação**

Para realizar os testes das histórias incluídas na secção anterior, foi utilizado o Selenium IDE para criar 5 testes que verificam se a funcionalidade pretendida cumpre os requisitos de desenvolvimento. Estes testes foram executados sobre a versão da aplicação entregue e o ficheiro com as "test suites" denomina-se *pawdemic\_tests.side* e pode ser encontrado no [repositório](#) da aplicação .

Selenium IDE - pawdemic\_tests

Project: pawdemic\_tests

Tests + REC

Search tests... 🔍

consulta\_urgencia  
contratar\_petsitter  
**encomendar\_racao**  
procurar\_anuncio  
publicar\_anuncio

Command	Target	Value
1 open	/	
2 set window size	960x1021	
3 click	css=css-view-1dbjc4n:nth-child(2) > .c ss-view-1dbjc4n > .css-textinput-11aywtz	
4 type	css=css-view-1dbjc4n:nth-child(2) > .c ss-view-1dbjc4n > .css-textinput-11aywtz	artur@gmail.com
5 click	css=css-view-1dbjc4n:nth-child(3).css -textinput-11aywtz	
6 type	css=css-view-1dbjc4n:nth-child(3).css -textinput-11aywtz	corona
7 click	css=css-view-1dbjc4n:nth-child(4) > .c ss-text-901ao	
8 assert text	css=.css-text-901ao:nth-child(4)	Artur Guimarães
9 assert text	css=r-width-13q2uu > .css-text-901oa o:nth-child(3)	Fiona
10 assert title	ProfileView	
11 click	css=r-flexWrap-tzz3ar > .css-view-1dbj c4n:nth-child(3)	
12 assert text	css=css-view-1dbjc4n:nth-child(3) > r- flexDirection-18u37iz > .css-view-1dbjc 4n:nth-child(2) > .css-text-901ao:nth..	Ração de Frango para Cão Adulto Gra nde
13 click	css=css-view-1dbjc4n:nth-child(3) > r- flexDirection-18u37iz > .css-view-1dbjc 4n:nth-child(2)	

Command: open Target: / Value: Description:

Log Reference

Figura 10: Testes criados para verificar as funcionalidades desenvolvidas

## 5 Referências e recursos

### Referências

- [1] Ilídio Oliveira. *MAS P- Demo Selenium IDE*. URL: [https://www.youtube.com/watch?v=xY\\_qzCfCYfo](https://www.youtube.com/watch?v=xY_qzCfCYfo).
- [2] *What is Sequence Diagram?* URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>.