

Projeto Temático em Desenvolvimento Web

1º RELATÓRIO

Escola Superior de Tecnologia e Gestão de Águeda

Universidade de Aveiro

1º semestre

2018-2019

Grupo 2:

António Pinho	nº74339
João Novo	nº84205
Luís Pinho	nº83926
Pedro Martinho	nº76480
Rui Duarte	nº29979

Projeto Temático em Desenvolvimento Web

1º RELATÓRIO

Escola Superior de Tecnologia e Gestão de Águeda

Universidade de Aveiro

1º semestre

2018-2019

Grupo 2:

António Pinho	nº74339
João Novo	nº84205
Luís Pinho	nº83926
Pedro Martinho	nº76480
Rui Duarte	nº 29979

Orientadores: Rita Santos; Fábio Martins

Agradecimentos

Este trabalho é o culminar de um longo processo e resulta do nosso esforço pessoal e do empenho de todos os elementos do grupo, mas o mesmo nunca teria sido realizado com sucesso sem o contributo e apoio de diversas pessoas que, de forma direta ou indireta, nele colaboraram.

Queremos agradecer, de forma muito especial, à Professora Doutora Ana Rita Costa Bonifácio Selores dos Santos e ao Professor Doutor Fábio José Reis Luís Marques, pela orientação e disponibilidade ao longo deste trabalho. A sabedoria e qualidade dos orientadores na supervisão foram cruciais para a realização deste projeto.

A toda a equipa de docentes deste curso, pelo apoio, por tudo o que nos ensinaram e sobretudo pela metodologia de trabalho que nos inculcaram.

A todos eles o nosso muito obrigado.

ÍNDICE

1. Introdução	1
2. Levantamento do estado da arte	2
3. Planeamento e execução	5
3.1 Planeamento	5
3.2 Executado	7
3.3 Requisitos	9
3.4 Modelo de casos de utilização.....	13
4. Prototipagem	14
4.1 Userflows	14
4.2 Wireframes.....	18
4.3 Protótipo de baixa fidelidade	21
4.4 Acessibilidade	25
4.5 Protótipo de alta fidelidade	26
5. Implementação	29
5.1 Identificação e Análise de Tecnologias.....	29
5.2 Soluções tecnológicas.....	29
5.3 Estrutura MVC a ser implementada	30
5.4 Base de dados	31
5.1 Testes	36
6. Produção, distribuição e lançamento da versão final.....	42
6.1 Processo de implementação	42
6.2 Arquitetura e organização da aplicação	43
7. Análise de resultados	44
8. Reflexão e conclusão	47
Bibliografia.....	48
Apêndice.....	49

Índice de tabelas

Tabela 1 - Atividades e esforço planeado	5
Tabela 2 - Atividades e esforço executado	7
Tabela 3 - Requisitos funcionais	10
Tabela 4 - Requisitos não funcionais.....	11
Tabela 5 –Comparação de tempo de descriptação dos diferentes algoritmos	35

Índice de figuras

Figura 1 - Caraterísticas do sistema Vendanor	2
Figura 2 - Sistema Aburi Gas	3
Figura 3 – Sistemas Siraga	4
Figura 4 - Diagrama de Gantt da calendarização conforme planeado	6
Figura 5 - Diagrama de Gantt da calendarização conforme executada	8
Figura 6 - Diagrama de casos de utilização	13
Figura 7 - Userflow de visualização dos produtos e máquinas (landing page)	14
Figura 8 - Userflow do processo de login	15
Figura 9 - Userflow do processo de compra	16
Figura 10 - Userflow do administrador	17
Figura 11 - Página inicial	18
Figura 12 - Lista de produtos.....	19
Figura 13 - Detalhes do produto.....	19
Figura 14 - Página do Mapa.....	20
Figura 15 - Style guide	27
Figura 16 - Estrutura MVC a ser implementada	30
Figura 17 - Diagrama lógico da Base de Dados.....	32
Figura 18 - Resultado da função select_all_triggers()	34
Figura 19 - Demonstração da análise feita pela extensão “WAVE”	36
Figura 20 - Erros e avisos iniciais	37
Figura 21 - Erros e avisos após uma análise e correção dos problemas encontrados	38
Figura 22 - Página inicial – Chrome iPhone 6	39
Figura 23 - Página inicial – Chrome iPad.....	40
Figura 24 - Página inicial – Firefox iPhone.....	40
Figura 25 - Página inicial – Firefox iPad	41
Figura 26 - Página inicial – Firefox.....	41
Figura 27 - Página inicial	44
Figura 28 – Loja	45
Figura 29 - Área do cliente	45
Figura 30 - Mapa	46
Figura 31 - Wireframe Detalhes do produto (com pop-up)	53
Figura 32 - Wireframe 1º ecrã do checkout	54
Figura 33 - Wireframe 2º ecrã de checkout	54
Figura 34 - Wireframe 3º ecrã de checkout	54
Figura 35 - Wireframe 4º ecrã de checkout	55
Figura 36 - Wireframe Página inicial com pop-up utilizador	55
Figura 37 - Wireframe Histórico de compras	56
Figura 38 - Wireframe favoritos	56

Figura 39 - Wireframe detalhes do utilizador	57
Figura 40 - Wireframe Mapa	57
Figura 41 - Wireframe página de login do admin.....	58
Figura 42 - Wireframe página inicial admin.....	58
Figura 43 - Wireframe logs das máquinas.....	59
Figura 44 - Wireframe lista das máquinas	59

1. INTRODUÇÃO

Com este projeto pretendemos criar uma aplicação Web que permita a compra online e posterior levantamento de botijas de gás numa das diversas máquinas de venda espalhadas pelo país. Esta plataforma irá também permitir aos utilizadores aceder e gerir as suas informações pessoais, como histórico de compras e tipos de botijas preferidas. Este projeto enquadra-se, portanto, no comércio de botijas de gás.

A importância deste projeto está diretamente relacionada com o melhoramento da conveniência na aquisição deste tipo de produtos por parte dos consumidores. Muitas das vezes, os estabelecimentos que vendem estes produtos praticam horários de venda pouco cómodos para a maioria dos consumidores. Este projeto pretende permitir a qualquer utilizador comprar e levantar botijas de gás a qualquer hora do dia. Assim como o melhoramento da comodidade, pretende-se também inovar, criando um processo mais seguro, controlado e simplificado para esta área de comércio.

A realização deste projeto começou pelo planeamento prévio do mesmo, de forma a garantir um desenvolvimento organizado e com prazos bem definidos. De seguida foram definidos os objetivos iniciais do projeto e levantados os requisitos, sucedendo-se a criação do design inicial da aplicação web (desenho de *wireframes* e prototipagem). Após isto, procedeu-se à fase de implementação, onde foi programada a plataforma web, base de dados entre outros recursos, que foram depois alojados num servidor real. Por fim, foi executada a fase de testes, onde foram realizados testes à plataforma com objetivo de garantir a sua estabilidade e fiabilidade.

Nos seguintes capítulos deste relatório são descritas as diferentes etapas estabelecidas e executadas para a realização deste projeto, desde o planeamento (capítulo 3) até à implementação (capítulo 6), descrevendo cada uma delas em detalhe.

2. LEVANTAMENTO DO ESTADO DA ARTE

Com o objetivo de descobrir e analisar soluções semelhantes já existentes no mercado foi efetuada uma pesquisa na internet por sistemas semelhantes ao que queremos criar. Nessa pesquisa verificámos que já existem diversas soluções no mercado, implementadas de diferentes formas, e como tal estas devem ser analisadas para ajudar na melhor implementação possível do nosso sistema.

Fica de seguida a análise feita a quatro sistemas de máquinas de venda de gás. Nesta análise foram levantados os pontos que considerámos mais importantes em cada um destes sistemas.

Vendador (figura 1):

- A compra e pagamento são feitos na máquina pelo utilizador.
- Permite devolução de botijas usadas no ato da compra.
- Existe uma aplicação que permite ver informações do sistema a quem faz a manutenção do mesmo (como estatísticas, estado de cada máquina, etc.).
- Alertas de manutenção enviados por SMS aos técnicos.

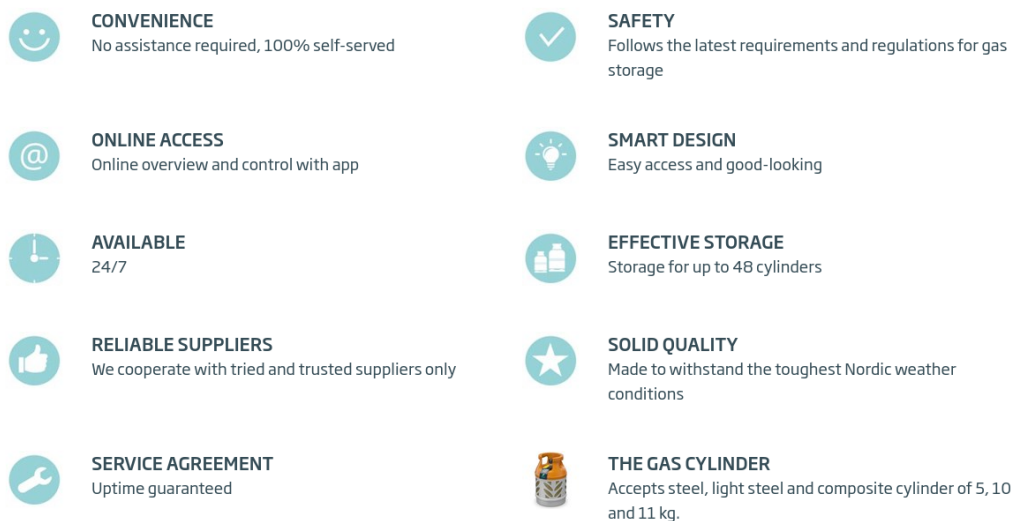


Figura 1 - Caraterísticas do sistema Vendador

Aburi Gas Automat (figura 2):

- Permite aos utilizadores comprarem botijas sem estarem registados no sistema.
- Permite devolução de botijas usadas no ato da compra de uma nova botija. Para isto o utilizador tem que estar registado.
- Cada garrafa é identificada com um tag RFID.
- Pagamento é efetuado na própria máquina e pode ser feito através de cartão bancário, dinheiro, ou através de um cartão de utilizador do sistema.



Figura 2 - Sistema Aburi Gas

Siraga Vendis (figura 3):

- Pagamento efetuado na máquina.
- Funciona com 4 marcas diferentes de botijas (no máximo).
- Funciona com base num sistema de reconhecimento de cor, para identificar as diferentes marcas.
- Permite devolução de botijas usadas no ato da compra de uma nova botija.
- Cada marca pode ser repostada independentemente das outras.

Siraga DistriGas (figura 3):

- Pagamento efetuado na máquina.
- Máquina com menor capacidade.
- Funciona apenas com uma marca de botijas.
- Funciona parcialmente com energia solar.

Após esta análise podemos concluir que todos estes sistemas têm ligeiras diferenças uns dos outros, o que os torna únicos no mercado, bem como muitas semelhanças ao sistema que pretendemos implementar. No entanto não existe nenhum serviço que permita comprar as botijas de gás online, sendo apenas possível fazê-lo junto a uma máquina no momento de levantamento da botija. O nosso objetivo para esta plataforma é implementar esta funcionalidade que consideramos estar em falta, criando para esse fim um website para permitir a compra online de botijas bem como a manutenção do sistema.

■ ■ ■ The gas cylinder distribution leaps forward !

Following the example of the LPG cylinder, the distribution market has been facing in 2010 a real revolution. Indeed, in addition to be attractive, cylinders must always be available to the purchaser. In order to meet this new demand, Siraga proposes a wide range of gas cylinder vending machines that operate 24/7. Each model of the vending machines meets specific expectations:

- **Vendis** is multi-brand (up to 4), which are identified thanks to a colour recognition system, and contains up to 168 cylinders. It is not only possible to exchange a cylinder but also to buy a new one. Eventually, each brand can be independently reloaded, 24/7.



- **Distri'Gas** is mono-brand; it holds between 18 and 48 cylinders, according to the number of modules chosen. Distri'Gas can be equipped with solar panels, allowing operations in full autonomy and, that way, contribute to the respect of the environment.



Figura 3 – Sistemas Siraga

3. PLANEAMENTO E EXECUÇÃO

3.1 Planeamento

Esta secção do relatório destina-se à apresentação do planeamento inicial do projeto, onde os prazos e tarefas apresentados correspondem a uma execução ideal do trabalho.

3.1.1 Atividades e esforço

A tabela 1 corresponde à lista de atividades elaborada durante a fase de planeamento do projeto e o esforço alocado a cada elemento.

Tabela 1 - Atividades e esforço planeado

Nº	Tarefa	Responsabilidade(%)				
	Início Fase I					
1	Levantamento do estado da arte	0	33	33	33	0
2	Identificação e descrição dos requisitos funcionais e não funcionais	50	0	0	0	50
3	Diagrama de casos de uso	50	0	0	0	50
4	Prototipagem de baixa fidelidade					
4.1	Desenvolvimento de Mockups	0	33	33	33	0
4.2	Feedback/Validação pelo cliente	50	0	0	0	50
4.3	Diagrama de userflow	0	33	33	33	0
4.4	Identificação de tecnologias	50	0	0	0	50
5	Elaboração do relatório Inicial	20	20	20	20	20
	Entrega Fase I	20	20	20	20	20
	Início Fase II					
6	Definição e implementação da BD					
6.1	Diagrama do modelo relacional	50	0	0	0	50
6.2	Implementação do SGBD	50	0	0	0	50
7	Definição da camada de apresentação da aplicação Web					
7.1	Html	20	20	20	20	20
7.2	Css / Bootstrap	20	20	20	20	20
7.3	Javascript / jQuery	20	20	20	20	20
7.4	Php / Laravel	20	20	20	20	20
8	Elaboração de relatório intermédio	20	20	20	20	20
	Entrega Fase II					
	Início Fase III					
9	Implementação da aplicação web	20	20	20	20	20
10	Teste, validação e identificação de eventuais falhas	20	20	20	20	20
11	Elaboração de relatório final	20	20	20	20	20
	Entrega Fase III	20	20	20	20	20
		António Pinho	João Novo	Luís Pinho	Pedro Martinho	Rui Duarte

3.1.2 Calendarização

A figura 4 apresenta o diagrama de Gantt que pretende ilustrar, de forma gráfica, a distribuição temporal das tarefas conforme o planeado:

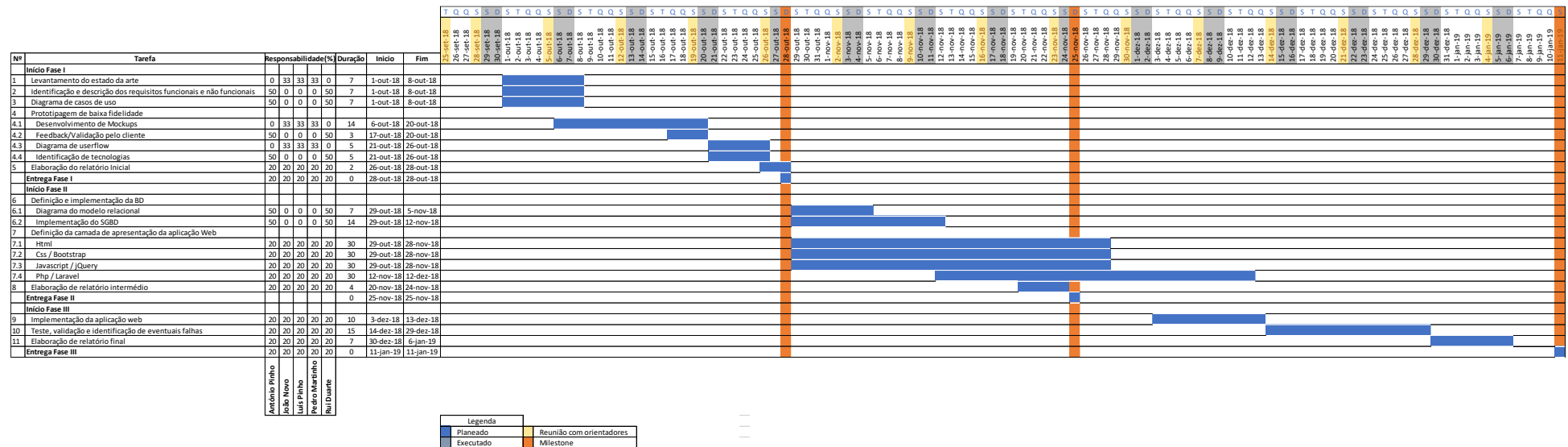


Figura 4 - Diagrama de Gantt da calendarização conforme planeado

3.2 Executado

Esta secção do relatório destina-se à apresentação das tarefas e prazos conforme executados.

3.2.1 Atividade e esforço

A tabela 2 apresenta a lista de atividades conforme executadas ao longo do projeto e o esforço alocado a cada elemento:

Tabela 2 - Atividades e esforço executado

Nº	Tarefa	Responsabilidade (%)				
	Início Fase I					
1	Levantamento do estado da arte	0	50	10	40	0
2	Identificação e descrição dos requisitos funcionais e não funcionais	50	0	0	0	50
3	Diagrama de casos de uso	50	0	0	0	50
4	Prototipagem de baixa fidelidade					
4.1	Desenvolvimento de Mockups	0	20	60	20	0
4.2	Feedback/Validação pelo cliente	35	0	10	20	35
4.3	Diagrama de userflow	0	0	0	100	0
4.4	Identificação de tecnologias	50	0	0	0	50
5	Elaboração do relatório Inicial	25	0	25	25	25
	Entrega Fase I	20	20	20	20	20
	Início Fase II					
6	Definição e implementação da BD					
6.1	Diagrama do modelo relacional	50	0	0	0	50
6.2	Implementação do SGBD	50	0	0	0	50
7	Definição da camada de apresentação da aplicação Web					
7.1	Html	30	20	10	10	30
7.2	Css / <i>Bootstrap</i>	30	20	10	10	30
7.3	Javascript / jQuery	0	80	0	0	20
7.4	Php / Laravel	20	50	0	5	25
8	Elaboração de relatório intermédio	20	20	20	20	20
	Entrega Fase II					
	Início Fase III					
9	Implementação da aplicação web	0	100	0	0	0
10	Teste, validação e identificação de eventuais falhas	20	20	20	20	20
11	Elaboração de relatório final	20	20	20	20	20
	Entrega Fase III					
		António Pinho	João Novo	Luís Pinho	Pedro Martinho	Rui Duarte

3.2.2 Calendarização

A figura 5 apresenta o diagrama de Gantt que ilustra de forma gráfica a distribuição temporal das tarefas conforme o executado.

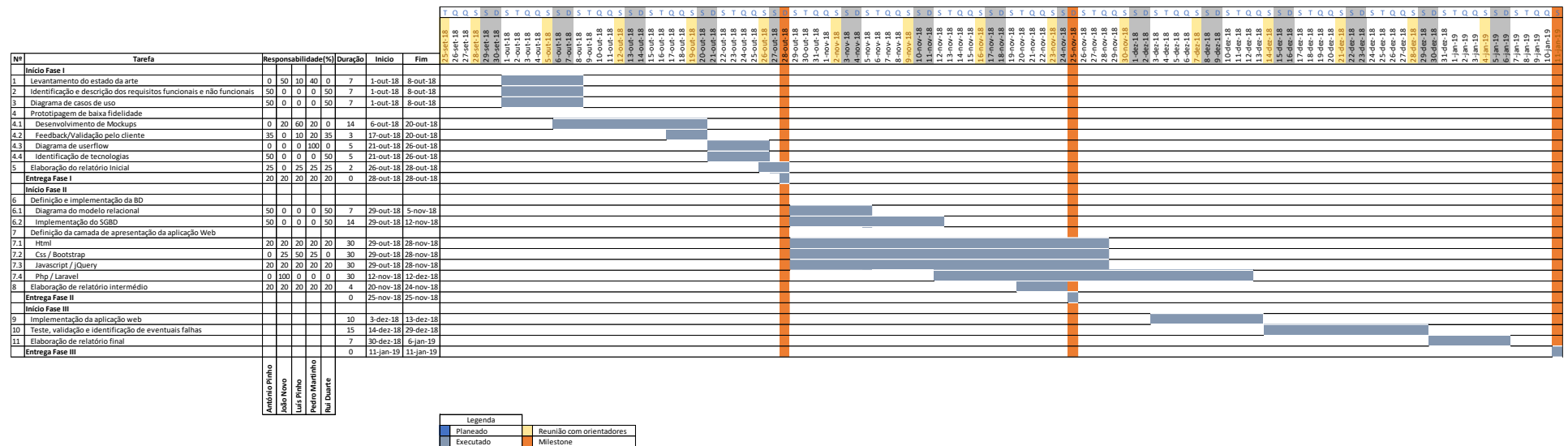


Figura 5 - Diagrama de Gantt da calendarização conforme executada

3.3 Requisitos

3.3.1 Reunião inicial

A reunião com o cliente que decorreu no dia 12 de outubro foi um ponto fundamental para entender, de facto, o que este pretende que a aplicação dê resposta e como o cliente entende que deverá ser executado.

No sentido de obter o máximo de informação possível do cliente, os grupos reuniram-se e elaboraram um documento único com uma série de questões (apêndice 2). Este documento abrange questões desde o público-alvo aos recursos e aspetos de web design. Estas questões provaram-se fundamentais para a elaboração do levantamento de requisitos. Após a reunião, foi criado um único documento que juntou toda a informação obtida do cliente face a estas questões.

Após elaboração do levantamento do estado da arte e da reunião com o cliente, deu-se seguimento à elaboração da lista de requisitos funcionais e não funcionais que o projeto se propõe a cumprir. A lista está ordenada por ordem de prioridade, sendo: “Alta” o mais prioritário e “Baixa”, o menos

3.3.2 Levantamento de requisitos funcionais

A tabela 3 apresenta os requisitos funcionais relevantes para o projeto, ordenados por prioridade.

Tabela 3 - Requisitos funcionais

Refª	Requisito funcionais	Prioridade
RF.1	O utilizador deve conseguir criar uma conta	Alta
RF.2	O utilizador e administrador autenticam-se no sistema	Alta
RF.3	O utilizador pode selecionar a botija a adquirir	Alta
RF.4	O utilizador pode verificar a disponibilidade de stock	Alta
RF.5	O utilizador pode efetuar compras	Alta
RF.6	O utilizador pode selecionar vários métodos de pagamento (paypal, mbway, cartão pré-pago)	Alta
RF.7	O utilizador visualiza a fatura gerada	Alta
RF.8	O utilizador acede ao QR code gerado para ser utilizado na máquina	Alta
RF.9	O utilizador acede a um código numérico de 10 dígitos para ser utilizado na máquina	Alta
RF.10	O utilizador acede à localização da máquina mais próxima com stock	Alta
RF.11	O utilizador acede a um mapa com a localização das máquinas	Alta
RF.12	O utilizador pode efetuar a compra com ou sem tara	Alta
RF.13	O administrador adiciona máquinas	Alta
RF.14	O administrador acede a informações tais como o stock e o estado das máquinas	Alta
RF.15	O administrador recebe alertas relacionados com as máquinas (stocks a 0, downtime das máquinas)	Alta
RF.16	O utilizador recebe uma mensagem com o estado da compra	Alta
RF.17	O utilizador pode associar o seu cartão físico à sua conta	Alta
RF.18	O utilizador pode verificar o saldo do cartão	Alta
RF.19	O utilizador pode criar uma lista de favoritos	Alta
RF.20	O administrador pode aceder a dados estatísticos tais como o uptime do sistema, vendas mensais por máquina)	Média
RF.21	O utilizador pode atualizar os seus dados	Baixa
RF.22	O administrador acede a logs do sistema	Baixa
RF.23	O utilizador pode consultar o histórico de compras	Baixa

3.3.3 Levantamento de requisitos não funcionais

A tabela 4 apresenta os requisitos não funcionais, ordenados por prioridade.

Tabela 4 - Requisitos não funcionais

Refª	Requisito não funcional	Prioridade
RNF.1	Compatível com os browsers mais populares (Chrome 49/59/63/67/68/69, Safari 11, IE11 e Firefox 62)	Alta
RNF.2	Adaptado para smartphones/tablet	Alta
RNF.3	Encriptação dos dados de autenticação	Alta
RNF.4	Deverá ser identificado o utilizador com a sessão ativa para que o utilizador saiba que está autenticado	Média

3.3.4 Descrição geral do Sistema

O processo de interação com a plataforma deverá ser simples, com elementos diretos e com pouco esforço na utilização de features/realização de tarefas. Após criar uma conta na plataforma usando os seus dados pessoais, poderá fazer login na mesma e explorar o catálogo de botijas disponíveis para venda. O processo de seleção e compra de botijas deverá ser o seguinte:

1. O utilizador navega por um catálogo que contém todas as botijas para venda no sistema. Caso o utilizador já tenha efetuado login, são apresentadas as botijas preferidas do utilizador.
2. Ao encontrar a botija desejada, poderá verificar a sua disponibilidade nos diferentes pontos de recolha.
3. Após escolher qual comprar (e selecionadas as preferências do utilizador para a compra) é feito o pagamento a partir de um dos métodos de pagamento disponíveis (MBWay, Paypal ou cartão físico).
4. Por fim, assim que for validado o pagamento, é gerado um código QR que permitirá ao utilizador levantar a botija selecionada em qualquer um dos pontos de recolha do país (é ainda gerado um código numérico de 10 dígitos que poderá ser introduzido na máquina para realizar a compra caso o utilizador não tenha possibilidade de usar o código QR). O utilizador poderá ainda definir se quer ou não reservar a botija no ponto de recolha selecionado por 1 hora a partir da compra.

Após chegar a um ponto de recolha, o utilizador apresenta o seu código de compra a um leitor da máquina de vendas. Após a validação deste código, é aberto um pórtico vazio no qual o utilizador poderá deixar a sua garrafa de gás vazia. Após deixar a garrafa e fechar o pórtico referente, é aberto automaticamente o pórtico com a botija pretendida pelo utilizador. O utilizador retira a botija e fecha o pórtico.

Dentro da plataforma, existe a área pessoal do utilizador onde terá acesso e a possibilidade de editar os seus dados pessoais, incluindo os seus métodos de pagamento e histórico de compras. Poderá ainda criar e editar listas de favoritos, que contêm as botijas de gás que o utilizador prefere e que compra mais frequentemente.

Existe também uma área na plataforma dedicada aos administradores da plataforma. Nesta área, o administrador poderá visualizar dados estatísticos como as botijas mais vendidas em cada máquina e os dias em que existem mais compras. Terá ainda uma área dedicada à gestão de máquinas, onde poderá ver a informação de cada uma, o seu stock e ainda adicionar máquinas novas ao sistema.

3.4 Modelo de casos de utilização

3.4.1 Visão geral

O diagrama de caso de uso da figura 6 apresenta um conjunto de funcionalidades propostas para o sistema implementado, funcionando como auxiliar no levantamento dos requisitos funcionais do sistema.

Cada caso de uso representa uma interação entre um utilizador e o sistema; interação essa que tem de ser considerada uma ação (unidade de trabalho significativa).

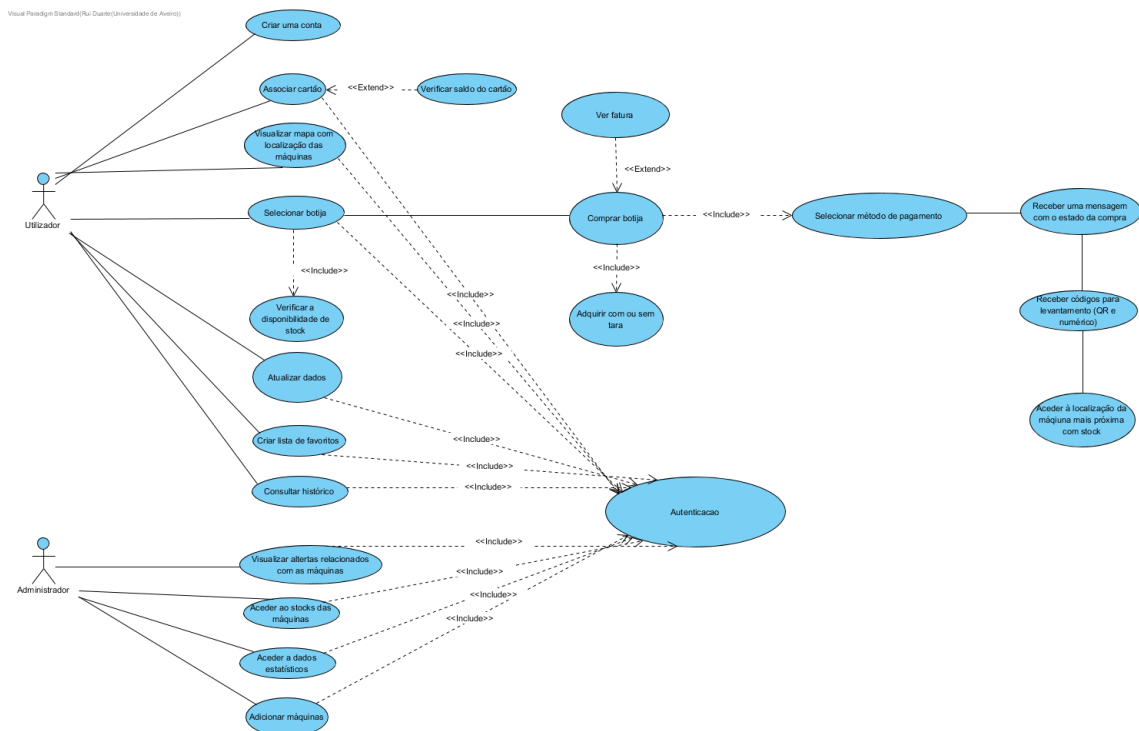


Figura 6 - Diagrama de casos de utilização

4. PROTOTIPAGEM

4.1 Userflows

Os seguintes userflows mostram cenários em que o utilizador se poderá encontrar e quais as suas possibilidades de interação com o sistema, como a visualização de produtos e máquinas (figura 7) do processo de login (figura 8), do processo de compra (figura 9) assim como o userflow do administrador (figura 10).

Este userflow mostra todas os caminhos e as suas condições que o utilizador pode seguir na página inicial, como visualizar produtos e visualizar os detalhes de um em específico.

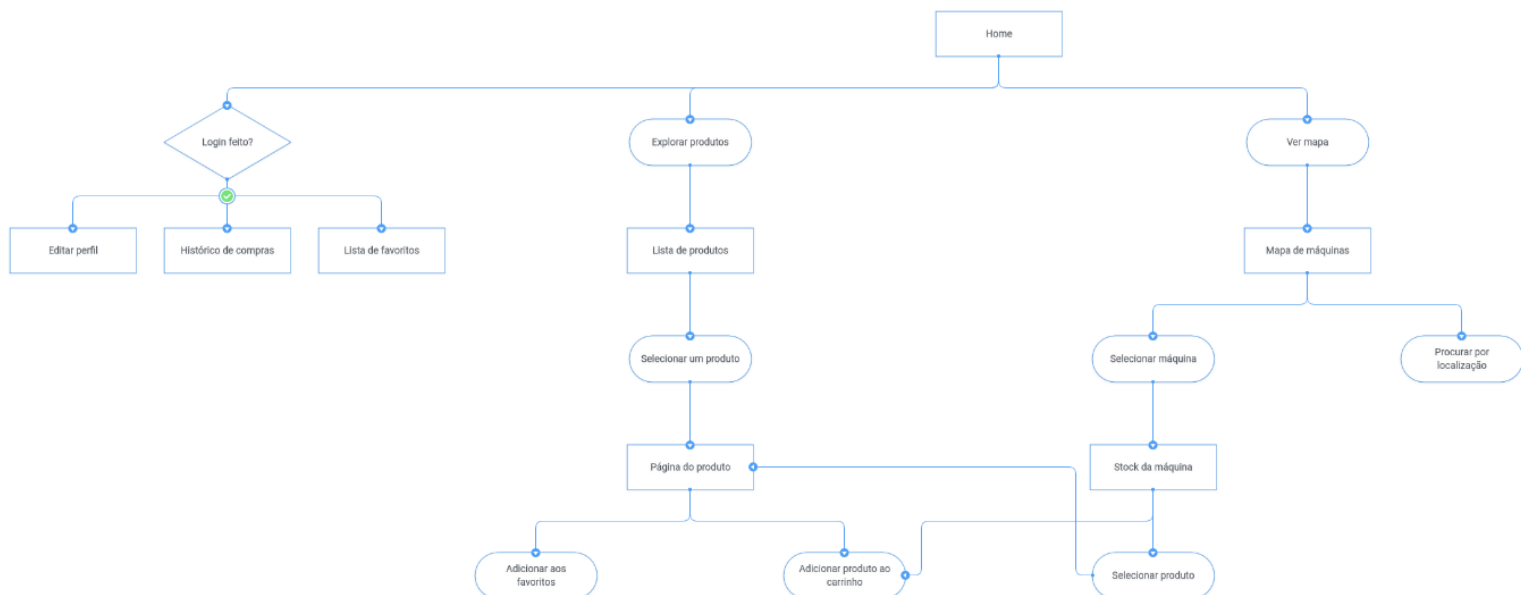


Figura 7 - Userflow de visualização dos produtos e máquinas (landing page)

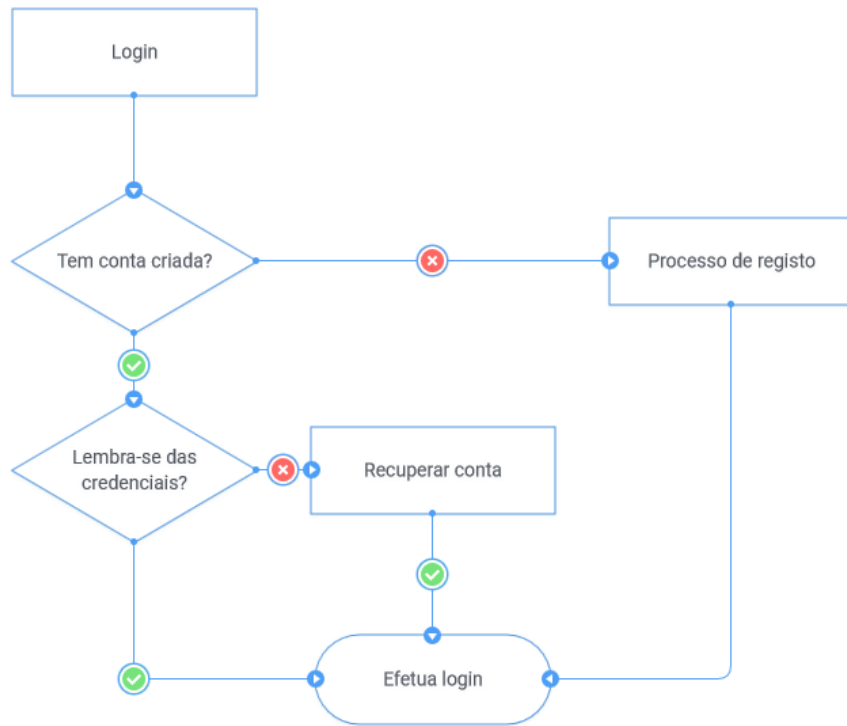


Figura 8 - Userflow do processo de login

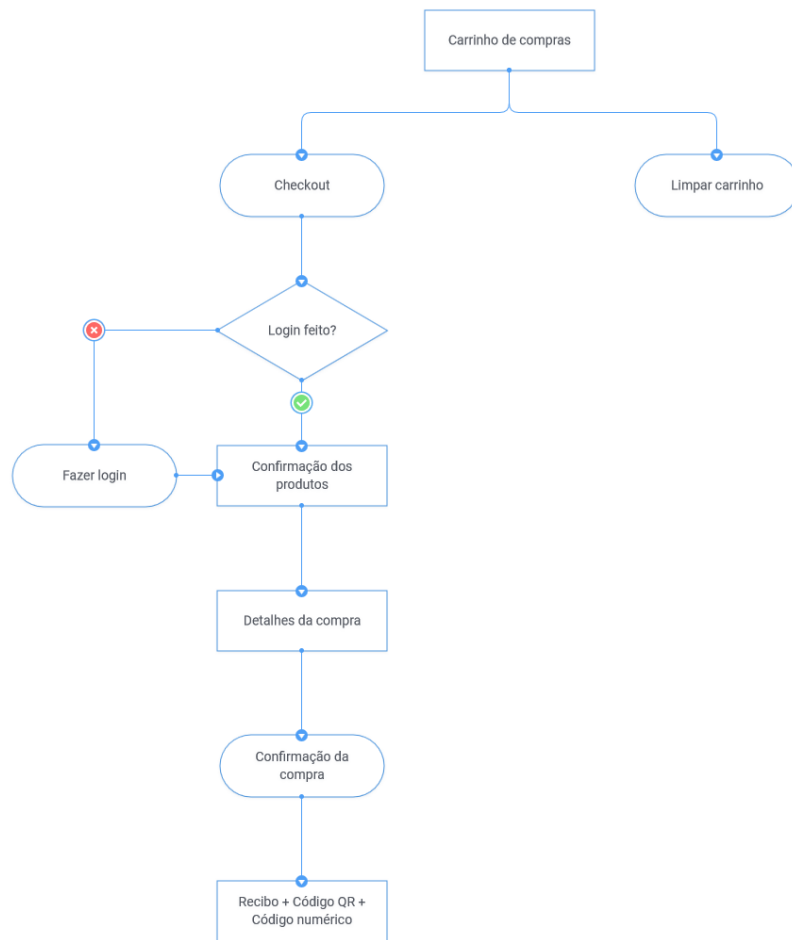


Figura 9 - Userflow do processo de compra

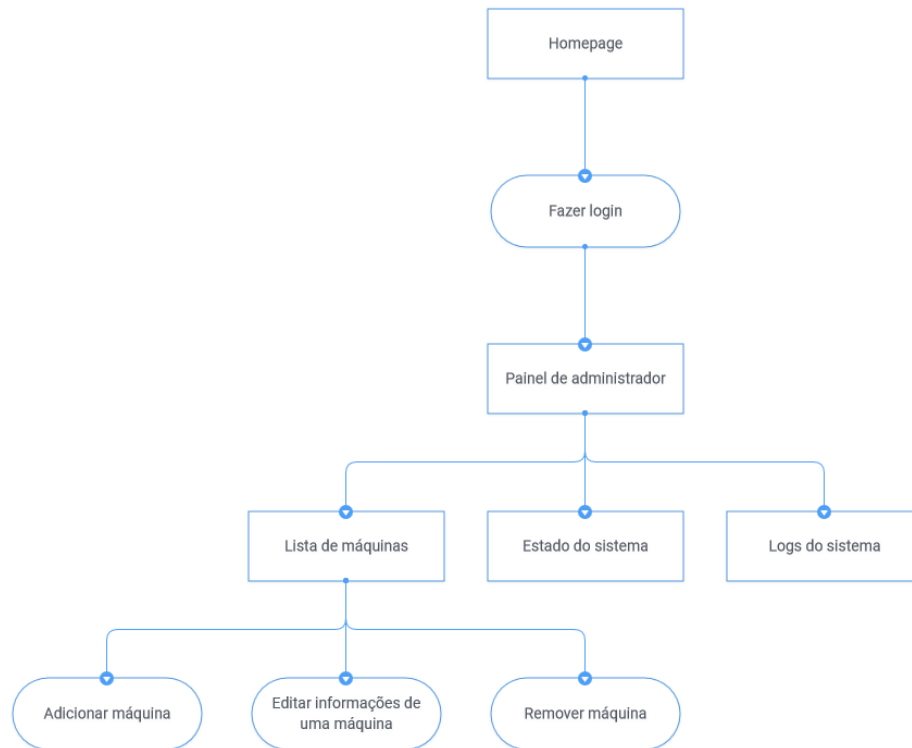


Figura 10 - Userflow do administrador

4.2 Wireframes

Um *wireframe* é um desenho básico de uma interface gráfica, com objetivo de criar o esqueleto da mesma. Foram criados estes *wireframes* de forma a criar a estrutura da página, identificando o espaço necessário e localização para cada parte da informação que será apresentada em cada página web. A criação destes *wireframes* para o projeto foi feita com recurso ao software Balsamic Mockups, um software próprio para a criação de *mockups* de baixa fidelidade.

De seguida encontram-se alguns dos principais *wireframes* referentes às páginas desta plataforma (página inicial, lista de produtos, detalhes de produto). Cada *wireframe* contém uma breve descrição do mesmo. As restantes *wireframes* podem ser consultadas na secção de apêndices do relatório (capítulo IV).

Nota: apesar não estar presente em todos os *mockups*, o *footer* é persistente e, tal como a *navbar*, irá aparecer em todas as páginas.

Página inicial - De destacar a *navbar* (presente em todas as páginas) que contém o logotipo do sistema, um campo de pesquisa, botões para a lista de produtos e mapa, assim como o carrinho de compras do utilizador e um elemento dedicado a mostrar o estado de login do utilizador (alterado consoante o utilizador tenha feito login na sua conta ou não).



Figura 11 - Página inicial

Lista de produtos – página que mostra todas as garrafas de gás e acessórios disponíveis na plataforma. Com filtros de pesquisa do lado esquerdo para permitir encontrar a garrafa desejada facilmente. A área de pesquisa encontra-se integrada na *navbar*.

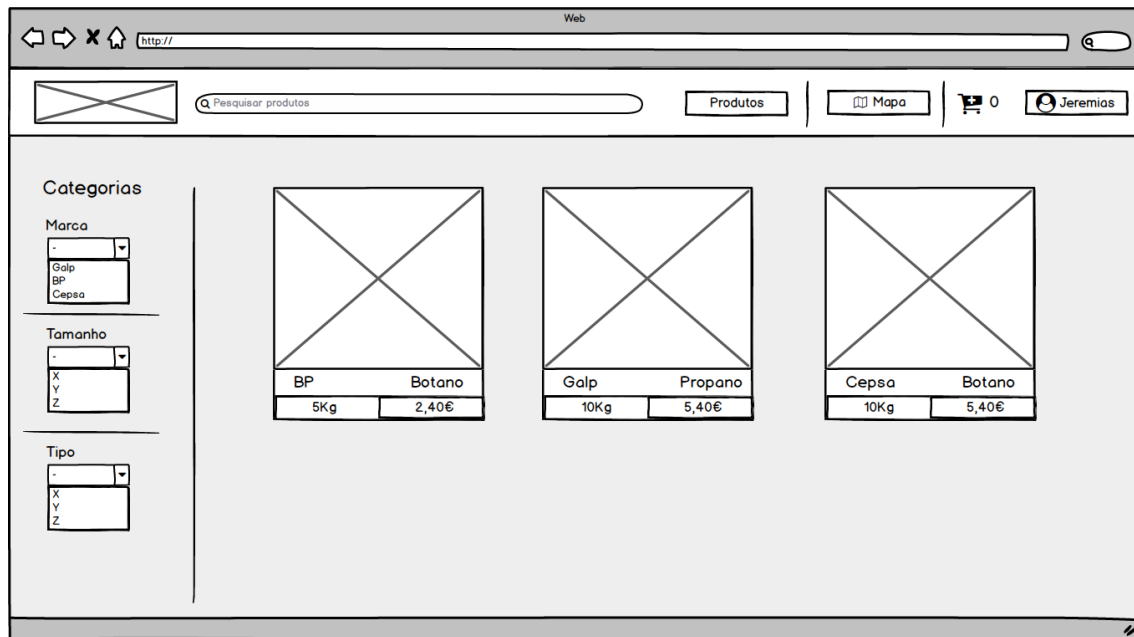


Figura 12 - Lista de produtos

Página de detalhes de um produto – mostra os detalhes de uma botija de gás ou produto específico (apresentada após clicar numa botija da lista anterior). Permite ainda adicionar o produto ao carrinho e aos favoritos do utilizador.

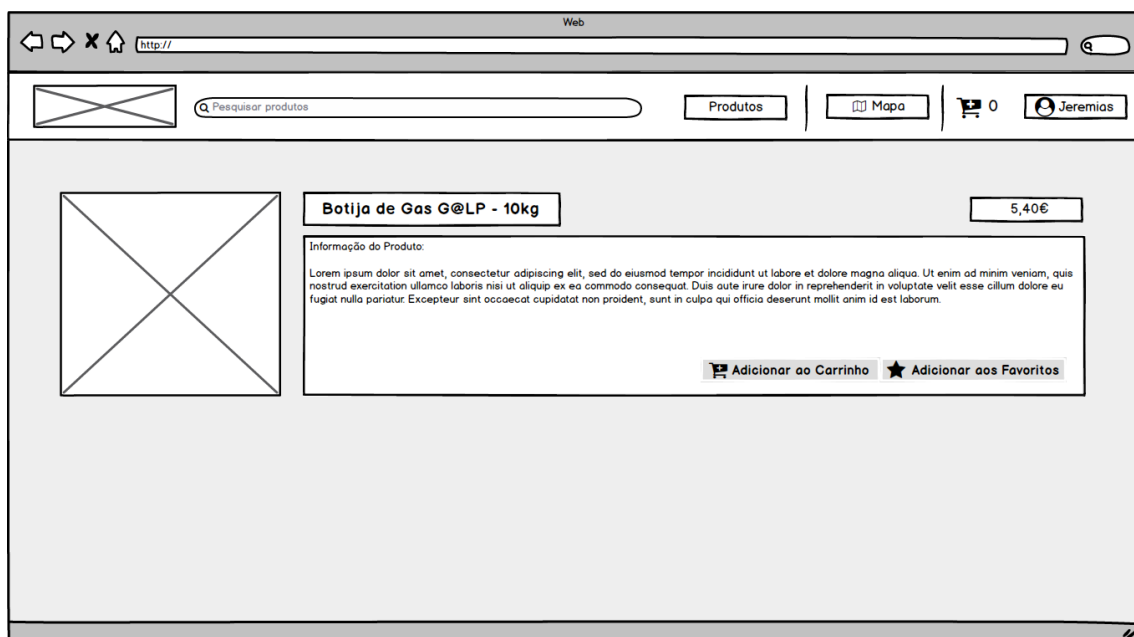


Figura 13 - Detalhes do produto

Página da localização das máquinas - apresenta um mapa onde se encontram as diversas máquinas da plataforma. Ao clicar numa das mesmas, poderá verificar o seu stock, comprar diretamente uma das garrafas disponíveis e visualizar a distância da posição atual do utilizador à máquina em questão. No topo do mapa encontra-se um campo de pesquisa que

permitirá ao utilizador “saltar” para uma determinada localização e visualizar todas as máquinas que se encontram lá.

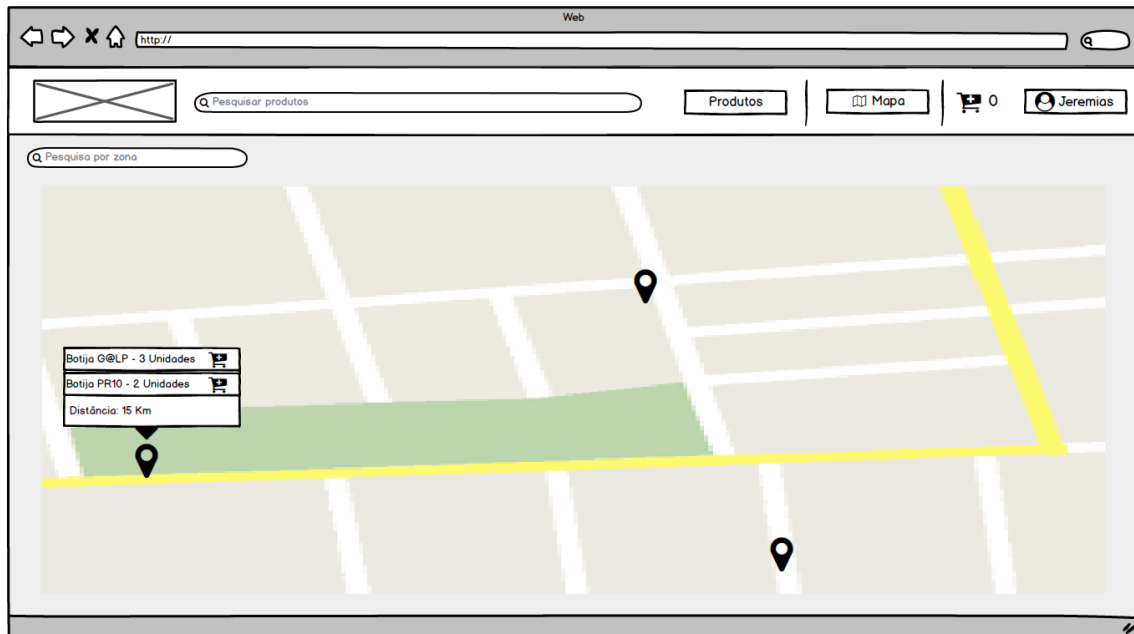


Figura 14 - Página do Mapa

4.3 Protótipo de baixa fidelidade

Após ter sido concebido um esqueleto base para cada uma das páginas principais da plataforma (os *wireframes*, demonstrados no tópico anterior) foi criado um protótipo de baixa fidelidade baseado nos mesmos. Este protótipo teve como objetivo demonstrar a interação do utilizador com o sistema usando uma série de passos para fazer o utilizador testar algumas das funcionalidades previstas. De seguida estão listados os *userflows* criados e posteriormente usados para a criação deste protótipo.

4.3.1 Avaliação com cliente

De forma a conduzir todo o processo de testes de usabilidade com os utilizadores de forma ágil e metódica, elaborou-se um guião (apêndice I) para definir os protocolos a seguir durante os testes.

Este documento tem por finalidade contemplar todas as situações que possam surgir no contexto dos testes e definir um plano de ação para cada caso.

4.3.2 Preparação

Para a realização dos testes, todos os elementos do grupo devem estar presentes. Em caso de falta, devem ser feitas as alterações necessárias de forma a assegurar que são definidos um coordenador e pelo menos um observador. Em última instância, estes papéis podem ser assumidos pela mesma pessoa.

Os papéis de coordenador e observador são assegurados por:

Coordenador: Luís Pinho

Observador: Pedro Martinho, Rui Duarte

4.3.3 Coordenador

O Coordenador está encarregue de interagir com o utilizador durante o processo de testes. É o Coordenador que irá apresentar as tarefas ao utilizador.

Caso o utilizador levante questões relativamente à forma como proceder durante uma das tarefas, este deverá fornecer direções claras e concretas no sentido de o esclarecer. As informações fornecidas não devem conter pistas ou informações que facilitem a navegação na aplicação, mas apenas elucidar o objetivo de cada tarefa proposta.

O Coordenador deve:

- Conduzir os testes,
- Dar instruções clara e precisas (tarefas),
- Interagir com o utilizador,
- Ajudar o utilizador

O Coordenador não deve:

- Anotar comentários e observações do utilizador,
- Fazer comentários ou apontar pormenores da interface

4.3.4 Observador

O Observador assume um papel passivo durante a realização dos testes com o utilizador pelo que não deverá interagir diretamente com ele.

O observador deverá ir registando tudo o que considerar de relevância para a avaliação da interface, bem como preencher a Grelha de Avaliação (apêndice II).

O Observador deve:

- Observar o utilizador e tomar notas,
- Anotar medidas de desempenho,
- Anotar comentários e observações do utilizador.

O Observador não deve:

- Interagir com o utilizador,
- Fazer comentários.

4.3.5 Utilizador

O Utilizador apenas deverá realizar os testes de usabilidade.

O Utilizador tem a opção de recusar realizar uma ou mais tarefas ou até desistir do estudo.

O Utilizador irá realizar os testes de usabilidade. Para esse efeito será confrontado com vários *mockups* e será instruído para realizar diferentes tarefas (apêndice III).

É fortemente encorajado que o utilizador pense em voz alta ao realizar cada tarefa e que ofereça feedback e sugestões que possam melhorar a usabilidade do sistema.

No decorrer dos testes o utilizador poderá executar as seguintes ações:

- Executar as tarefas propostas pelo Coordenador,
- Pensar em voz alta durante a realização de cada tarefa,
- Colocar questões relativas à realização de cada tarefa,
- Desistir do estudo,
- Passar à tarefa seguinte sem concluir a atual.

4.3.6 Análise de resultados

Foram realizadas duas sessões de avaliação dos protótipos com dois grupos de utilizadores diferentes. Após estas sessões de avaliação foram efetuadas uma análise e uma síntese de resultados, com o intuito de alterar a interface atual de forma a criar uma melhor experiência de utilização no futuro. Ficam de seguida as observações mais importantes feitas pelos utilizadores:

- Interface de utilizador:
 - Deve ser gerado um QR code por cada produto comprado;
 - Deve ser possível escolher localizações de reserva diferentes para cada produto; que o utilizador comprar;
 - A lista de favoritos deve aparecer na página inicial do utilizador se o login estiver feito;
 - Deve existir uma função de compra rápida com um método de pagamento predefinido;
 - Associar cartão físico no processo de registo;
 - Meter preços com tara e sem tara na lista de produtos;
 - Compra direta a partir da lista de produtos (sem abrir a página do produto);
 - Adicionar aos favoritos a partir da lista de produtos;
 - Adicionar Paypal ou MBWay como método de pagamento.

- Interface de administrador:
 - Disponibilizar a opção de pesquisar máquinas por zona geográfica;
 - Gestão das máquinas com base em estatísticas (por exemplo, marca, região, nº de vendas, etc.).

4.4 Acessibilidade

De seguida são apresentados alguns dos princípios de acessibilidade mais comuns e a nossa estratégia de implementação dos mesmos no nosso projeto, de forma a que pessoas com incapacidades audiovisuais possam navegar de forma fluida no nosso website.

- Links com significado – os links no nosso website não são genéricos e fornecem ao utilizador indicações claras (não serão apresentados links por si, mas sim palavras ou pequenas frases que sirvam como links);
- Capacidade de alterar o tamanho da letra – o website em causa não permite alterar o tipo de letra, mas como o tamanho das fontes foi especificada em "em" ao invés de "px", este permite que as mesmas se adaptem a vários tamanhos de ecrã dinamicamente;
- Fornecer descrição das imagens – as imagens contêm a descrição da mesma no código (etiqueta "alt");
- Tabelas legíveis – as tabelas contêm cabeçalho, corpo e rodapé bem definidos no HTML, através das tags apropriadas, de forma a que possam ser lidas linha a linha. De forma a que estas fiquem mais legíveis, a cor das linhas pares será diferente das ímpares, criando assim um contraste melhor;
- Acesso rápido ao conteúdo, através da colocação de um link invisível no topo da página – será aplicado ao projeto (no entanto, foi classificado como um requisito de baixa prioridade);
- Legendagem e transcrição de conteúdo – o website não contém vídeos e/ou ficheiros de áudio. Como tal, este ponto não é aplicável;
- Daltonismo – A paleta de cores do nosso site permite um contraste elevado entre as mesmas, fazendo com que a informação seja facilmente distinguível. Foi tido o cuidado de incluir texto nas mensagens de sucesso e/ou erro que apenas contivessem cores;
- Navegação usando o teclado, através de "tabs" e/ou teclas de atalho – será aplicado ao projeto (no entanto, foi classificado como um requisito de baixa prioridade).

4.5 Protótipo de alta fidelidade

Após a realização dos protótipos de baixa fidelidade e tendo em atenção o feedback obtido da avaliação com utilizadores, deu-se início à prototipagem de alta fidelidade.

Esta etapa consiste em pormenorizar e detalhar mais o desenvolvido até aqui. Nesta fase já foram tomados em consideração aspetos como o esquema cromático, tipografia e responsividade no sentido de se assemelhar o máximo ao produto final. No entanto, todas as páginas são ainda estáticas, apenas com o código suficiente para inicializar as diferentes componentes e torná-la responsiva.

De forma a acelerar o processo de criação deste protótipo, foi realizada uma pesquisa com objetivo de encontrar um *template* já feito que fosse adequado a alguns dos nossos objetivos para esta página, como conter uma loja online, elementos gráficos com um estilo consistente, assim como já ter implementado todo o comportamento que uma página deverá ter para ser responsiva. O único requisito em termos de design que foi estabelecido como critério de pesquisa foi que o *template* deverá ser baseado na *framework* “*Bootstrap*”. Após esta pesquisa, chegámos a um consenso e escolhemos a uma *template* gratuita chamada “Fashe”, na qual foi baseada toda esta aplicação.

Encontram-se *screenshots* deste protótipo na secção V do apêndice.

Com o intuito de organizar todas as decisões feitas a nível de design para o website, foi criado um guia de estilos. Este guia de estilos foi criado com base numa página HTML (usando já os estilos CSS que são usados no website final), o que permite não só aceder à mesma a partir de qualquer lado (hospedada no servidor) assim como permitir a qualquer um dos elementos do grupo poder replicar os elementos gráficos mais facilmente copiando o código deste documento para cada um dos elementos usados. No guia de estilos, estão presentes os seguintes elementos:

- Esquema de cores adotado;
- Parágrafos (H1 – H6);
- Botões (primário, sucesso, erro, acentuação, neutro);
- *Input* (telemóvel, email, morada, nome, entre outros);
- Elemento de produto (a ser usado na lista de produtos);
- *Dropdown*

A *styleguide* pode ser consultada neste link: <http://estga-dev.clients.ua.pt/~ptdw-2018-gr2/styleGuide>.

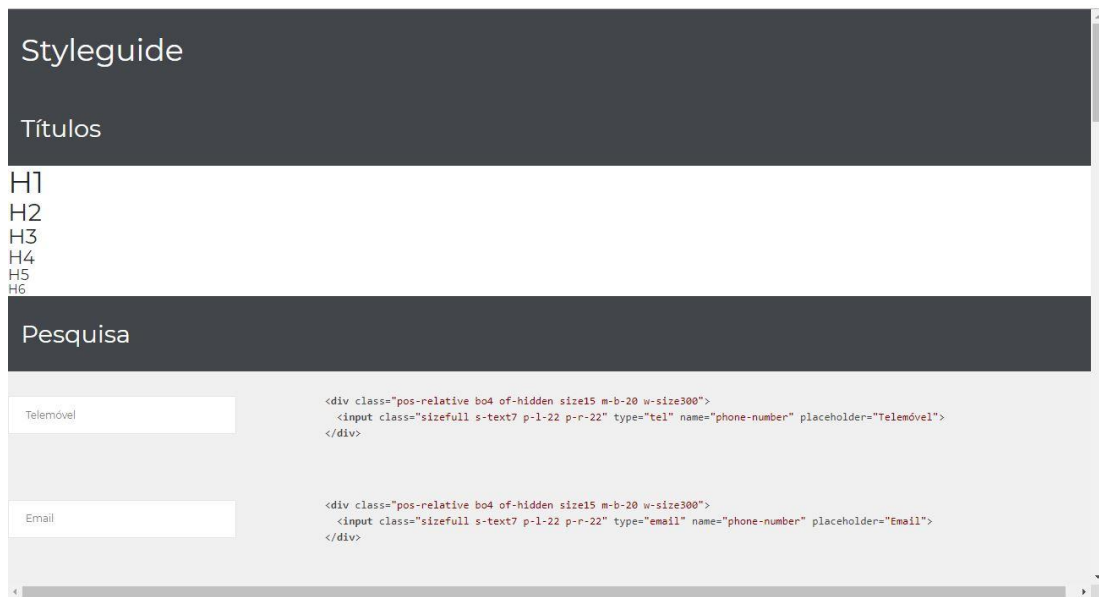


Figura 15 - Style guide

4.5.1 Avaliação com cliente

De forma a conduzir todo o processo de testes de usabilidade com os utilizadores de forma ágil e metódica, elaborou-se um guião (apêndice I) para definir os protocolos a seguir durante os testes.

Este documento tem por finalidade contemplar todas as situações que possam surgir no contexto dos testes e definir um plano de ação para cada caso.

4.5.2 Preparação

Para a realização dos testes, todos os elementos do grupo devem estar presentes. Em caso de falta, devem ser feitas as alterações necessárias de forma a assegurar que são definidos um coordenador e pelo menos um observador. Em última instância, estes papéis podem ser assumidos pela mesma pessoa.

Os papéis de coordenador e observador são assegurados por:

Coordenador: Luís Pinho

Observador: Pedro Martinho, Rui Duarte

4.5.3 Análise de resultados

Considerando que os protótipos não se encontravam em condições de apresentação, quer do ponto de vista de design gráfico quer de integração, optámos por não efetuar a avaliação com os utilizadores desta fase da prototipagem.

5. IMPLEMENTAÇÃO

5.1 Identificação e Análise de Tecnologias

Durante o realizar deste projeto, irá ser necessária a utilização de plataformas, ferramentas e ainda soluções tecnológicas para poder implementar o produto final.

5.1.1 Plataforma

É utilizada a plataforma do code.ua, que é assente em Redmine, onde a funcionalidade desta consiste em auxiliar na gestão do projeto no que toca em manter um registo de documentos importantes e alojamento do repositório onde vai ter a aplicação e com possibilidade de registar alterações feitas na mesma.

5.1.2 Ferramentas

Para diagramas como o diagrama de casos de uso, diagrama do site e outros semelhantes é usado o Visual Paradigm. Para o diagrama de Gantt foi usado o Excel.

Na criação e edição de código para os protótipos de média e alta fidelidade e na produção do produto final, é usado um IDE à escolha de cada um dos elementos do grupo. Das várias opções, são usados o Visual Studio Code e o PHP Storm.

5.2 Soluções tecnológicas

No que toca ao sistema de base de dados deu-se preferência ao PostgreSQL que tem como principais vantagens:

- ✓ Ser gratuito (Open Source);
- ✓ Performance bastante admirável;
- ✓ Multiplataforma;
- ✓ Altamente escalável.

O PostgreSQL tem algumas vantagens com relação ao MySQL, como integridade referencial e suporte a transações nativas. O MySQL também oferece esses recursos, mas de forma não nativa, recorrendo a terceiros (tabelas do tipo InnoDB). Por fim, de salientar que é o sistema que tem sido usado em projetos académicos anteriores.

Para a aplicação é usada a *framework* Laravel que utiliza a linguagem PHP 7 pois estamos perante uma aplicação que vai ser utilizada em ambiente local e de acesso restrito. A segurança e sigilo são dois fatores que a linguagem permite realizar e esta *framework* facilita a implementação de um modelo MVC com a mesma.

Tendo sido definido implementar um mapa na plataforma, foi também necessário escolher uma *framework* de mapas Web. Após uma pesquisa das diferentes soluções disponíveis, optámos por usar o MapBox, sendo que é possível usar uma conta gratuita (sem ser necessário inserir cartão de crédito), tem um limite grande nas chamadas à API do serviço e fornece todas as funcionalidades que são necessárias para a nossa situação (suporte para GeoJSON e representação de *layers* no mapa).

5.3 Estrutura MVC a ser implementada

A estrutura a ser implementada seguirá a figura 16. Na figura é possível perceber onde se encontraram as migrations, os controladores, requests, modelos, vem como onde serão colocados os ficheiros css, js e documentos gerados pela aplicação.

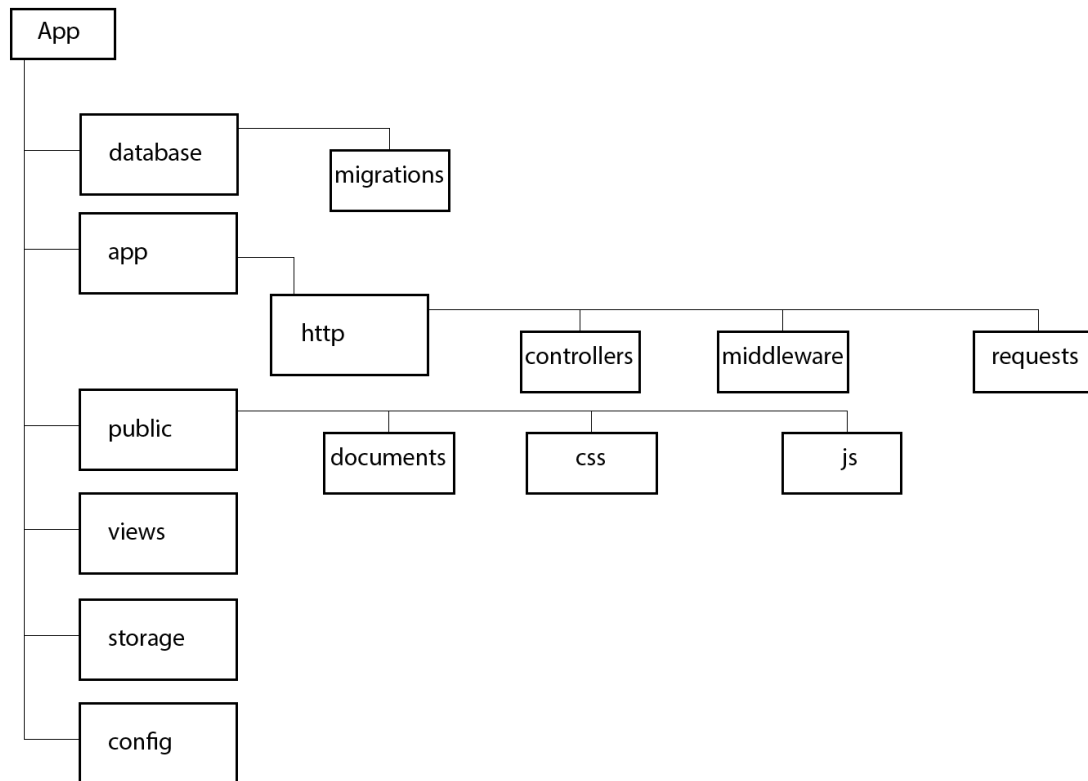


Figura 16 - Estrutura MVC a ser implementada

5.4 Base de dados

Pretendeu-se construir um modelo capaz de identificar todas as características do projeto, as atividades mais importantes, e priorizar as diferentes etapas do trabalho a desenvolver.

- ✓ Para a modelação da base de dados, tomou-se em atenção que esta deveria:
- ✓ Assegurar o cruzamento eficiente de informação, para evitar redundância de dados;
- ✓ Permitir obter ganhos importantes na análise e consulta da informação gerada;
- ✓ Reduzir o tempo na tomada de decisão, tornando a aplicação a desenvolver uma ferramenta relevante para a gestão.

Após a análise de requisitos e da criação do diagrama de casos de uso, procedeu-se à normalização dos conceitos. Seguiu-se a criação do diagrama lógico da base de dados, como demonstrado na figura 17.

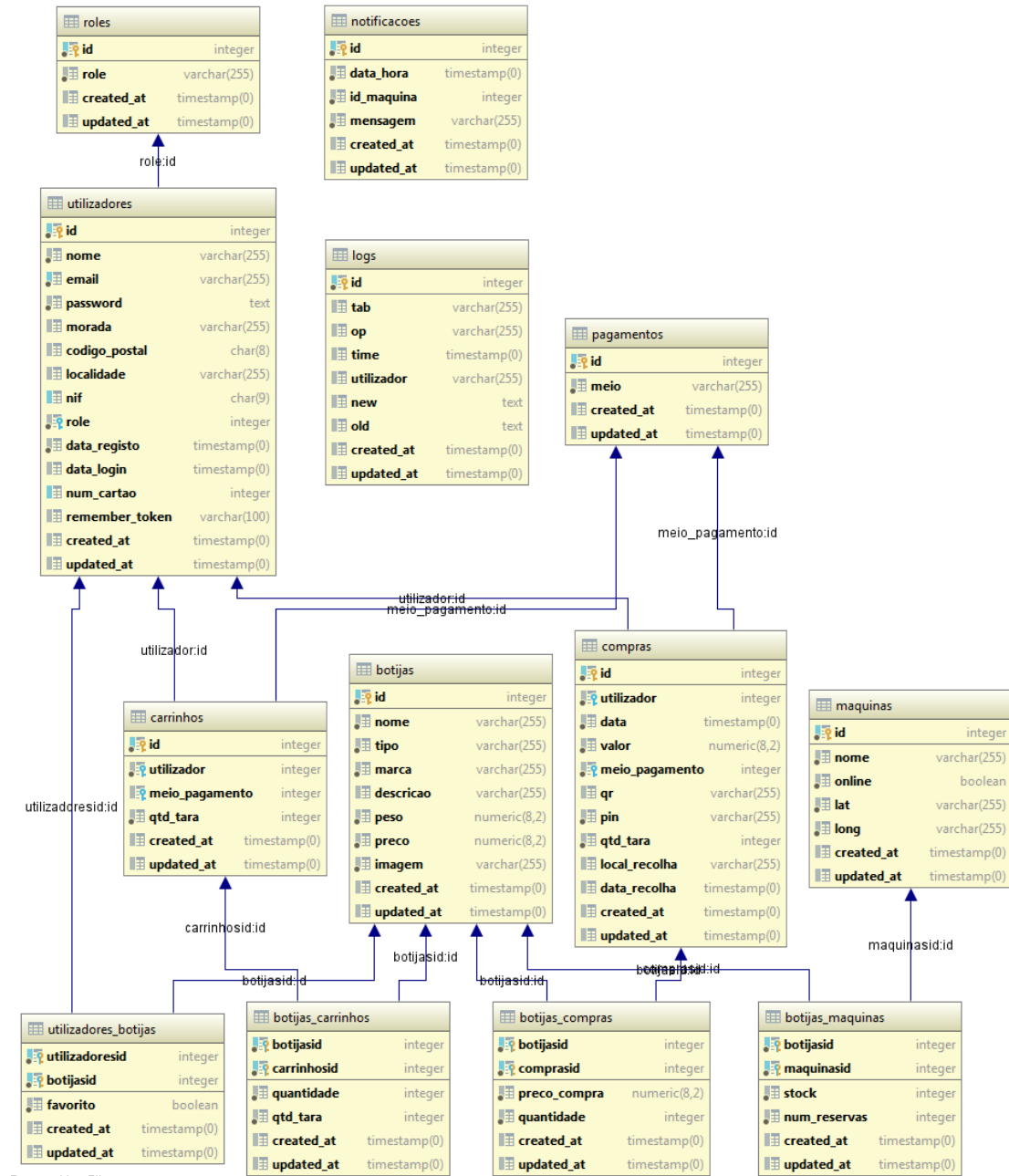


Figura 17 - Diagrama lógico da Base de Dados

5.4.1 SGBD alvo

O SGBD escolhido para desenvolver este projeto é o PostgreSQL. O PostgreSQL é um sistema de gestão de bases de dados objeto-relacional baseado no POSTGRES desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley.

Esta base de dados foi pensada tendo em vista o modelo funcional e de integridade dos seus dados. Recorreram-se a múltiplas tabelas por se considerar que algumas destas apenas fariam sentido se fossem únicas, evitando dados repetidos em várias tabelas. Com esta metodologia obtêm-se dados mais fidedignos no manuseamento das tabelas, como se pode verificar no modelo relacional. Adicionalmente, é possível ter um maior controlo sobre os dados inseridos pelos utilizadores do sistema aquando da inserção de dados errados.

5.4.2 Restrições (*Constraints*)

***Constraints* NOT NULL**

Os *constraints* NOT NULL permitem tornar um campo de preenchimento obrigatório. Foram usados em todos os atributos com a exceção dos considerados opcionais.

***Constraints* UNIQUE**

Os *constraints* UNIQUE permitem somente a introdução de valores únicos. Considerando que as chaves primárias e estrangeiras são sempre únicas, apenas foi aplicado este conceito aos atributos referentes ao email ("email"), número de contribuinte ("nif") e número de cartão("num_cartao") da relação utilizadores

5.4.3 Procedimentos de execução automática (*Triggers*)

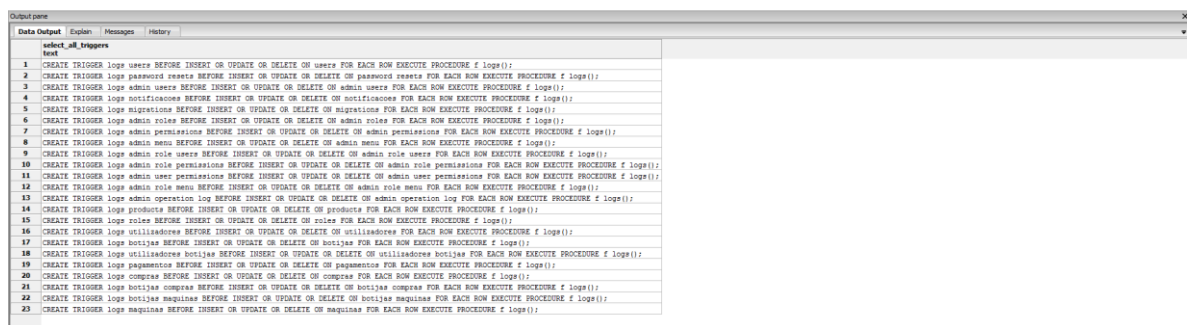
Um *trigger* é um tipo especial de *Stored Procedure* que é invocado automaticamente sempre que uma *query* de ação (INSERT, UPDATE ou DELETE) é executada sobre uma tabela ou *view* à qual o *trigger* esteja associado. Permitem implementar as regras definidas no desenho da BD, implementar regras subjacentes à lógica da aplicação e garantir a integridade e consistência dos dados.

Em termos de auditoria, foi criado uma relação *logs* (registos) com o objetivo de registar todas as operações do tipo INSERT, UPDATE e DELETE efetuadas em todas as tabelas da BD, com a óbvia exceção da própria relação de *logs*.

A função “f_logs()” foi a responsável por tal operação, gravando os dados alterados em formato JSON para a relação NEW ou OLD (dependendo do tipo de operação) como abaixo apresentado.

```
CREATE OR REPLACE FUNCTION f_logs() RETURNS trigger AS $$
DECLARE activeuser int:= (SELECT f_activeuser());
BEGIN
    IF TG_OP = 'INSERT' THEN
        INSERT INTO logs (tab, op, utilizador, new)
        VALUES (TG_RELNAME, TG_OP, activeuser, row_to_json(NEW));
        RETURN NEW;
    ELSIF TG_OP = 'UPDATE' THEN
        INSERT INTO logs (tab, op, utilizador, new, old)
        VALUES (TG_RELNAME, TG_OP, activeuser, row_to_json(NEW),
        row_to_json(OLD));
        RETURN NEW;
    ELSIF TG_OP = 'DELETE' THEN
        INSERT INTO logs (tab, op, utilizador, old)
        VALUES (TG_RELNAME, TG_OP, activeuser, row_to_json(OLD));
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;
```

De seguida é necessário criar um *trigger* para cada uma das relações a monitorizar. Para automatizar tal operação é criada a função “select_all_triggers()” que imprime um linha de texto por relação com o código para a criação do *trigger*. A figura 18 mostra a função em funcionamento.



Output name	Output
select_all_triggers	test
1	CREATE TRIGGER logs_users BEFORE INSERT OR UPDATE OR DELETE ON users FOR EACH ROW EXECUTE PROCEDURE f_logs();
2	CREATE TRIGGER logs_password_resets BEFORE INSERT OR UPDATE OR DELETE ON password_resets FOR EACH ROW EXECUTE PROCEDURE f_logs();
3	CREATE TRIGGER logs_admin_users BEFORE INSERT OR UPDATE OR DELETE ON admin_users FOR EACH ROW EXECUTE PROCEDURE f_logs();
4	CREATE TRIGGER logs_notifications BEFORE INSERT OR UPDATE OR DELETE ON notifications FOR EACH ROW EXECUTE PROCEDURE f_logs();
5	CREATE TRIGGER logs_migrations BEFORE INSERT OR UPDATE OR DELETE ON migrations FOR EACH ROW EXECUTE PROCEDURE f_logs();
6	CREATE TRIGGER logs_admin_roles BEFORE INSERT OR UPDATE OR DELETE ON admin_roles FOR EACH ROW EXECUTE PROCEDURE f_logs();
7	CREATE TRIGGER logs_admin_permissions BEFORE INSERT OR UPDATE OR DELETE ON admin_permissions FOR EACH ROW EXECUTE PROCEDURE f_logs();
8	CREATE TRIGGER logs_admin_menus BEFORE INSERT OR UPDATE OR DELETE ON admin_menus FOR EACH ROW EXECUTE PROCEDURE f_logs();
9	CREATE TRIGGER logs_admin_role_users BEFORE INSERT OR UPDATE OR DELETE ON admin_role_users FOR EACH ROW EXECUTE PROCEDURE f_logs();
10	CREATE TRIGGER logs_admin_role_permissions BEFORE INSERT OR UPDATE OR DELETE ON admin_role_permissions FOR EACH ROW EXECUTE PROCEDURE f_logs();
11	CREATE TRIGGER logs_admin_user_permissions BEFORE INSERT OR UPDATE OR DELETE ON admin_user_permissions FOR EACH ROW EXECUTE PROCEDURE f_logs();
12	CREATE TRIGGER logs_admin_role_menus BEFORE INSERT OR UPDATE OR DELETE ON admin_role_menus FOR EACH ROW EXECUTE PROCEDURE f_logs();
13	CREATE TRIGGER logs_admin_operation_log BEFORE INSERT OR UPDATE OR DELETE ON admin_operation_log FOR EACH ROW EXECUTE PROCEDURE f_logs();
14	CREATE TRIGGER logs_products BEFORE INSERT OR UPDATE OR DELETE ON products FOR EACH ROW EXECUTE PROCEDURE f_logs();
15	CREATE TRIGGER logs_roles BEFORE INSERT OR UPDATE OR DELETE ON roles FOR EACH ROW EXECUTE PROCEDURE f_logs();
16	CREATE TRIGGER logs_utilizadores BEFORE INSERT OR UPDATE OR DELETE ON utilizadores FOR EACH ROW EXECUTE PROCEDURE f_logs();
17	CREATE TRIGGER logs_botijas BEFORE INSERT OR UPDATE OR DELETE ON botijas FOR EACH ROW EXECUTE PROCEDURE f_logs();
18	CREATE TRIGGER logs_utilizadores_botijas BEFORE INSERT OR UPDATE OR DELETE ON utilizadores_botijas FOR EACH ROW EXECUTE PROCEDURE f_logs();
19	CREATE TRIGGER logs_pagamentos BEFORE INSERT OR UPDATE OR DELETE ON pagamentos FOR EACH ROW EXECUTE PROCEDURE f_logs();
20	CREATE TRIGGER logs_compras BEFORE INSERT OR UPDATE OR DELETE ON compras FOR EACH ROW EXECUTE PROCEDURE f_logs();
21	CREATE TRIGGER logs_botijas_compras BEFORE INSERT OR UPDATE OR DELETE ON botijas_compras FOR EACH ROW EXECUTE PROCEDURE f_logs();
22	CREATE TRIGGER logs_botijas_maquinas BEFORE INSERT OR UPDATE OR DELETE ON botijas_maquinas FOR EACH ROW EXECUTE PROCEDURE f_logs();
23	CREATE TRIGGER logs_maquinas BEFORE INSERT OR UPDATE OR DELETE ON maquinas FOR EACH ROW EXECUTE PROCEDURE f_logs();

Figura 18 - Resultado da função select_all_triggers()

O *trigger* “tg_password” permite encriptar o conteúdo do atributo ‘password’ referente à relação *utilizador* aquando da inserção ou alteração de um determinado utilizador, recorrendo à extensão “pgcrypto”. Encriptação essa feita através de um algoritmo BF (Blowfish) com 8 iterações e com 128bits de *salt* que, quando comparado com MD5, é bastante mais seguro e difícil de quebrar. A tabela 5 apresenta as diferenças de velocidade entre os vários tipos de algoritmos.

Tabela 5 –Comparação de tempo de descriptação dos diferentes algoritmos

Algorithm	Hashes/sec	For [a-z]	For [A-Za-z0-9]	Duration relative to md5 hash
crypt-bf/8	1792	4 years	3927 years	100k
crypt-bf/7	3648	2 years	1929 years	50k
crypt-bf/6	7168	1 year	982 years	25k
crypt-bf/5	13504	188 days	521 years	12.5k
crypt-md5	171584	15 days	41 years	1k
crypt-des	23221568	157.5 minutes	108 days	7
sha1	37774272	90 minutes	68 days	4
md5 (hash)	150085504	22.5 minutes	17 days	1

Salt é o valor obtido pela função *get_salt()*. Esta última função gera um valor aleatório que permite que utilizadores com passwords iguais tenham uma hash (password encriptada) completamente diferente. O código abaixo indicado apresenta o trigger em causa.

```
CREATE OR REPLACE FUNCTION f_password() RETURNS trigger AS $$
BEGIN
    NEW.password := crypt(NEW.password, gen_salt('bf', 8));
    RETURN NEW;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;
```

```
CREATE TRIGGER t_password BEFORE INSERT OR UPDATE OF password ON
utilizadores
FOR EACH ROW EXECUTE PROCEDURE f_password();
```

5.1 Testes

Atualmente existem diversas formas de navegar a Web, desde o método de navegação até ao tipo de dispositivo que é usado para este fim. Por isto, é necessário garantir que plataforma ser desenvolvida é acessível ao maior número possível de utilizadores e apresentada o mais corretamente e coerente possível.

Com o objetivo de validar o uso e funcionamento adequado da plataforma em diversos cenários de utilização, foram realizados alguns testes à mesma de forma a garantir o uso correto das convenções e tecnologias atuais. Os testes executados e os seus respetivos resultados são descritos nos pontos 5.1 (acessibilidade) e 5.2 (portabilidade).

Devido ao site não estar alojado num servidor publico, não foi possível utilizar as ferramentas de verificação de acessibilidade mais comuns, como o “W3C Validator”.

5.1.1 Testes de acessibilidade

Os testes de acessibilidade foram feitos recorrendo à extensão “WAVE” para o browser Google Chrome. Esta extensão foi criada pela “WEBAIM” e verifica os elementos da página em questão em busca de potenciais erros. Verifica até ao *standard* WCAG 2.0 AA. Em baixo encontra-se um *screenshot* com um teste feito à página inicial da plataforma e à loja (estes testes são preliminares e representam apenas um estado de *Release Candidate* da página, os resultados finais estão representados mais à frente).



Figura 19 - Demonstração da análise feita pela extensão “WAVE”

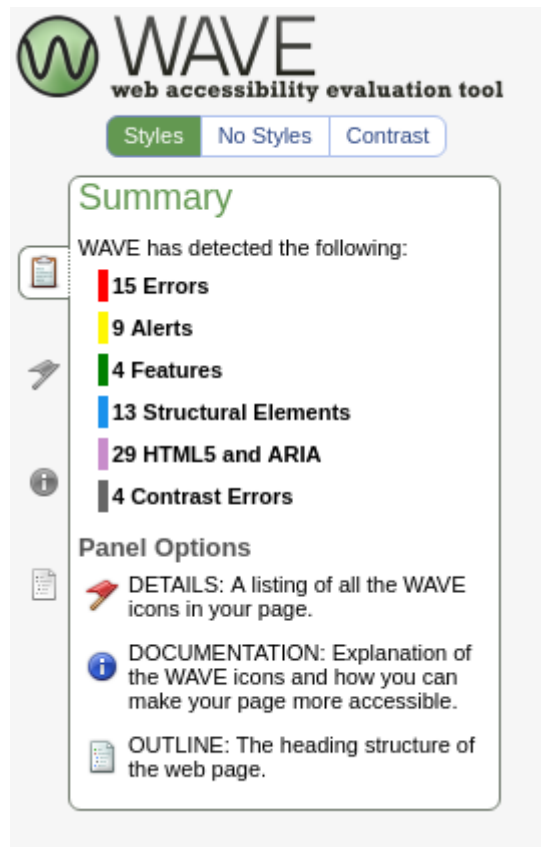


Figura 20 - Erros e avisos iniciais

Como é possível verificar, existem alguns erros e avisos na versão inicial desta página. Após uma análise e otimização da página, foram corrigidos a maior parte destes erros, desde *labels* em falta para elementos de *forms* até mau contraste entre elementos da página. Alguns destes erros não foram corrigidos pois não se aplicam à nossa página nem ou não críticos na utilização da página (como o texto na imagem inicial, ao qual a ferramenta aponta a ocorrência de mau contraste entre o texto branco e a imagem de tons escuros).

O mesmo teste foi feito nas restantes páginas da plataforma, com resultados semelhantes (os mesmos não serão documentados dada a sua similaridade com os erros anteriormente descritos). Após a correção e retificação destes erros, os erros e avisos foram bastante reduzidos, o que tornou a plataforma melhor em termos de acessibilidade. Alguns dos erros corrigido incluem:

- Definir a linguagem das páginas
- Alterar textos alternativos repetidos
- Atribuir *labels* a *inputs* sem o mesmo

Nesta imagem é possível ver o número de erros após uma retificação da página (número mais reduzido).

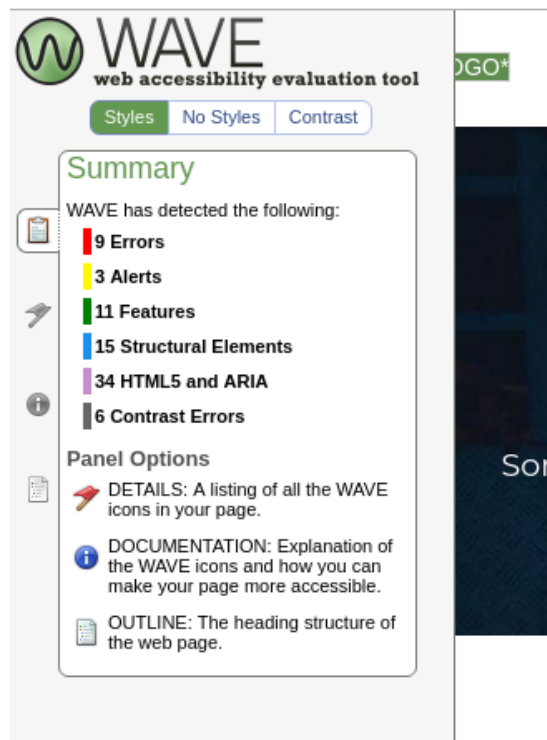


Figura 21 - Erros e avisos após uma análise e correção dos problemas encontrados

5.1.2 Testes de portabilidade

Os testes de portabilidade executados consistiram em testar o acesso e utilização da plataforma em diferentes dispositivos (smartphone, tablet e computador) assim como em diferentes browsers em cada dispositivo (Chrome, Firefox, Microsoft Edge), garantindo que o aspeto da plataforma não é alterado significativamente quando mudadas estas variáveis.

Mais uma vez, devido ao facto de a página estar offline e num servidor privado, não foi possível recorrer a uma ferramenta automática que verificasse numa grande quantidade de navegadores e dispositivos, pelo que este texto foi feito manualmente. Devido a isto, a plataforma foi apenas testada no Chrome, Firefox e Internet Explorer (em desktop, iPhone 8 e iPad emulados pelos browsers, à exceção do Internet Explorer).

Os resultados são bastante positivos, dado que a plataforma é representada corretamente em todos os dispositivos e em todos os browsers. As funcionalidades mantêm-se inalteradas e a performance é bastante boa e semelhante em todas as versões. Para demonstrar este comportamento, os *screenshots* em baixo representam a página inicial nos diferentes dispositivos (



Figura 22 - Página inicial – Chrome iPhone 6

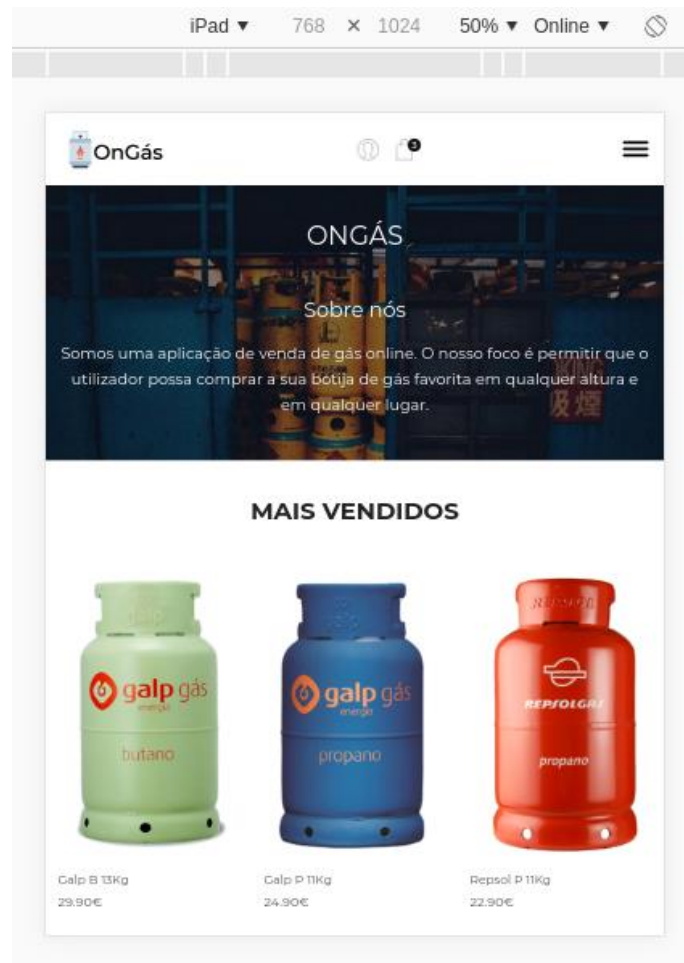


Figura 23 - Página inicial – Chrome iPad

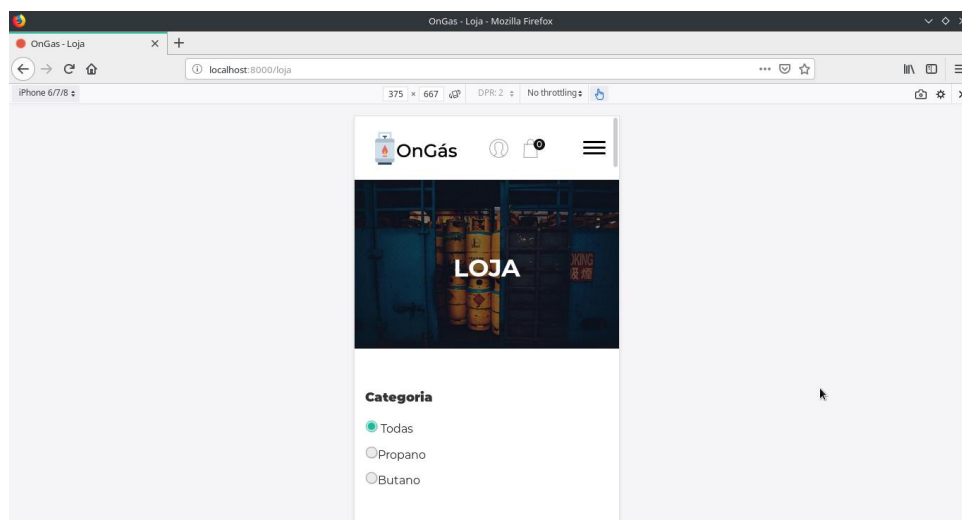


Figura 24 - Página inicial – Firefox iPhone

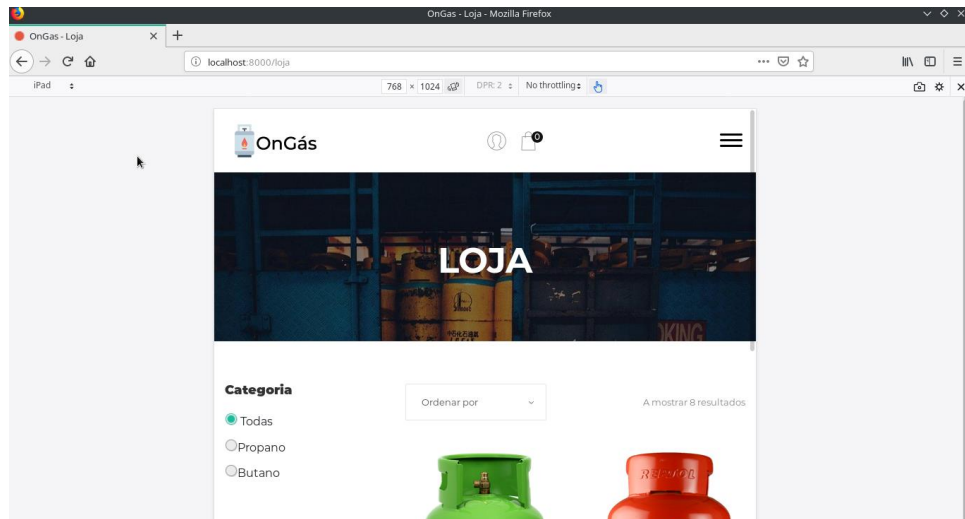


Figura 25 - Página inicial – Firefox iPad

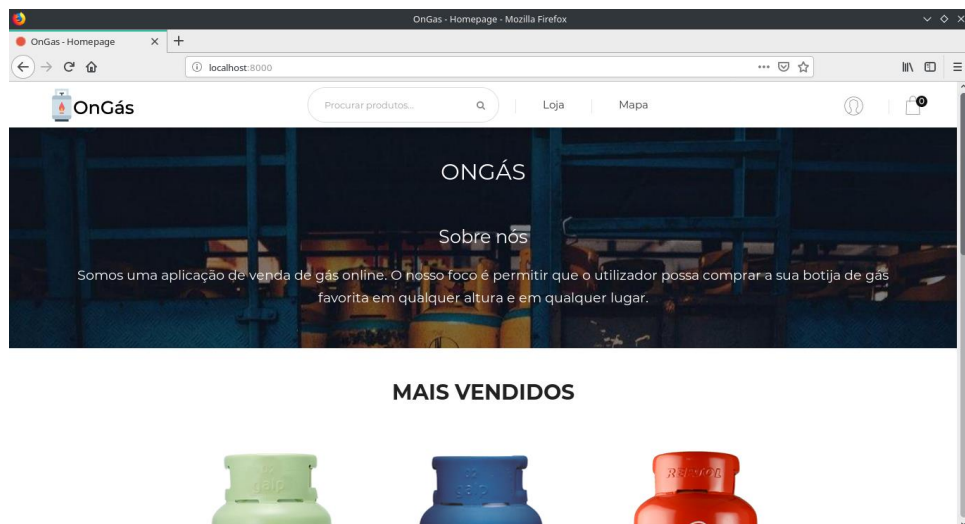


Figura 26 - Página inicial – Firefox

6. PRODUÇÃO, DISTRIBUIÇÃO E LANÇAMENTO DA VERSÃO

FINAL

A plataforma (tanto *frontend* como *backend*) será distribuída a partir de um servidor, nomeadamente o servidor da ESTGA (onde será alojada).

6.1 Processo de implementação

O processo de instalação desta plataforma envolve uma série de passos onde são instalados diversos componentes do PHP e da framework Laravel, assim como a sua configuração.

Os passos necessários são os seguintes:

1. Instalar PHP com versão igual ou superior a 7.1.3
2. Ativar as seguintes extensões:
 - a. OpenSSL PHP Extension
 - b. PDO PHP Extension
 - c. Mbstring PHP Extension
 - d. Tokenizer PHP Extension
 - e. XML PHP Extension
 - f. Ctype PHP Extension
 - g. JSON PHP Extension
 - h. BCMath PHP Extension
3. Instalar o composer
4. Configurar o env
 - a. Executar o comando `php artisan key:generate` (Para pôr a app key)
 - b. Preencher os dados de autenticação para a base dados.
 - i. `DB_CONNECTION=pgsql` (para usar postgres)
 - ii. `DB_HOST=127.0.0.1` (para usar o localhost, pode usar outro host)
 - iii. `DB_PORT=5432` (Port do postgres)
 - iv. `DB_DATABASE=ptdw-2018-gr2` (Nome da base dados)
 - v. `DB_USERNAME=ptdw-2018-gr2` (Nome do utilizador da base de dados)
 - vi. `DB_PASSWORD=password`
5. Executar os comandos na pasta do laravel:
 - a. `composer install`
 - b. `composer require werneckbh/laravel-qr-code`
 - c. `composer require encore/laravel-admin`

- d. php artisan vendor:publish --
provider="Encore\Admin\AdminServiceProvider"
 - e. php artisan migrate:fresh
 - f. php artisan admin:install
6. Instalação feita.

6.2 Arquitetura e organização da aplicação

De uma forma geral, a aplicação encontra-se dividida em três áreas principais: **loja/comércio** (página inicial, loja, mapa. Detalhes de produto), **área do processo de compra** (carrinho, dados da compra, todo o processo de compra) e **área pessoal do cliente** (ver editar dados pessoais, histórico de compras, ver favoritos).

7. ANÁLISE DE RESULTADOS

Os resultados deste projeto vão na maioria de encontro ao definido inicialmente, com a loja online totalmente funcional, com as funcionalidades implementadas:

- ✓ Pesquisa e seleção de produtos (sendo possível filtrar a pesquisa);
- ✓ Adicionar botijas à lista de favoritos do utilizador;
- ✓ Processo de compra (comprar produtos);
- ✓ Pesquisar e encontrar máquinas de venda no mapa;
- ✓ Criar conta de utilizador e fazer login;
- ✓ Visualizar dados pessoais, favoritos e histórico de compras;
- ✓ Páginas de contactos e informação sobre a plataforma.

A base de dados foi também desenhada e implementada com sucesso, sendo que está preparada para suportar todas as funcionalidades implementadas.

Contudo, não possível implementar alguns dos componentes inicialmente definidos no levantamento requisitos, como a área de administração.

Em baixo encontram-se alguns *screenshots* das principais páginas finais da plataforma (encontram-se mais *screenshots* nos anexos).

Página inicial, a página que o utilizador vê a primeira que entra na plataforma. Permite ver as botijas mais vendidas na plataforma. Contém uma pequena descrição da mesma. No topo da página, encontra-se uma *navbar* (presente em todas as páginas da plataforma), permite aceder ao mapa, loja, pesquisar produtos, fazer login/criar conta, aceder aos dados do utilizador e visualizar o carrinho atual.



Figura 27 - Página inicial

Loja, onde é apresentada uma grelha com todos os produtos disponíveis na plataforma. É possível filtra estes produtos usando os filtros no lado esquerdo da página.

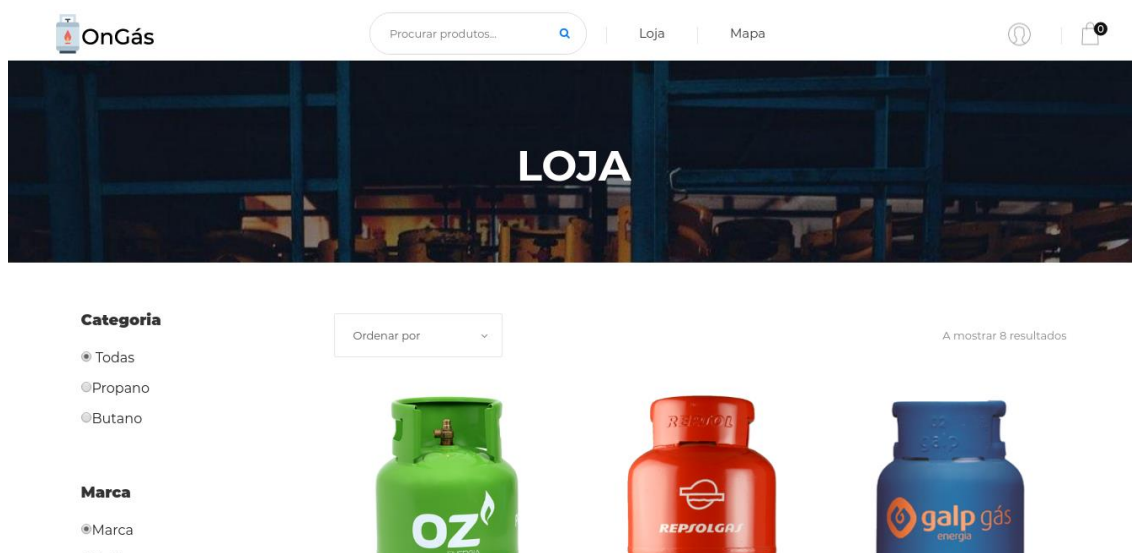


Figura 28 – Loja

Perfil do utilizador, onde o utilizador pode visualizar e editar os seus dados pessoais, assim como aceder ao seu histórico de compras e lista de favoritos.

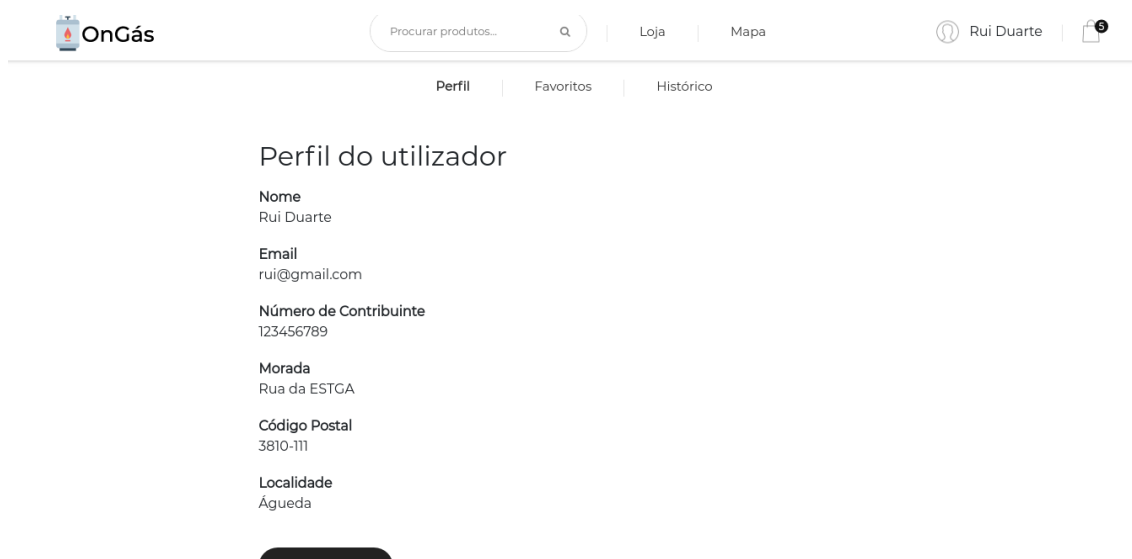


Figura 29 - Área do cliente

Mapa, onde o utilizador pode ver todas as máquinas em que pode levantar as suas botijas (todas as máquinas registadas na plataforma). Para além disto, é possível ver o stock de cada máquina e aceder aos detalhes de cada produto.

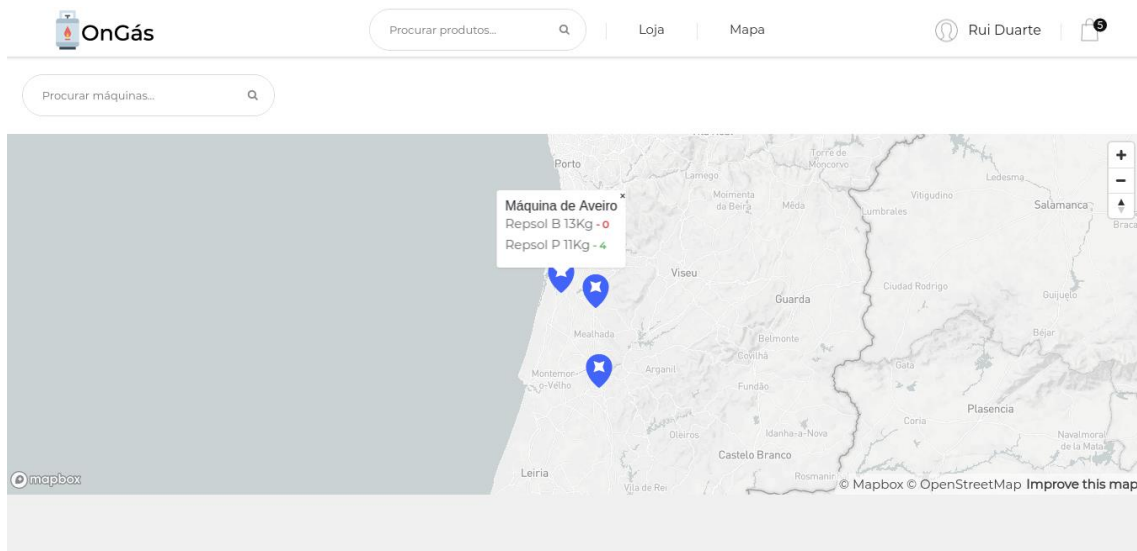


Figura 30 - Mapa

8. REFLEXÃO E CONCLUSÃO

O trabalho realizado até à data revelou-se um bom caso de estudo e oportunidade de aprendizagem relativamente ao desenvolvimento de aplicações web e aspetos de web design.

Assim, resumam-se os aspetos essenciais explorados até esta fase do projeto:

- Definir a arquitetura da aplicação Web tendo por base os requisitos previamente definidos,
- Aplicação dos conhecimentos adquiridos em IHC e Web Design,
- Aplicação dos conhecimentos adquiridos em DWM.

A elaboração deste projeto proporcionou uma excelente oportunidade para consolidar temas desenvolvidos ao longo das cadeiras lecionadas, bem como um desafio que maturou uma ética de trabalho já existente dos diversos elementos do grupo.

Existiram algumas dificuldades durante o curso do desenvolvimento do projeto a nível de organização e coordenação de tarefas entre os elementos do grupo, dificuldades estas que se refletiriam no resultado do produto final. Apesar de não ter sido cumprir todos os objetivos definidos (avaliação com os utilizadores do protótipo de alta fidelidade, área de administrador), foram completados na mesma a maioria dos requisitos base e necessários para criar uma plataforma funcional e que corresponde aos objetivos de uma plataforma de venda online.

BIBLIOGRAFIA

NUNES, Mauro; O'NEILL, Henrique. Fundamental de UML. 6ª ed. FCA

SILVA, Alberto; VIDEIRA, Carlos. UML, Metodologias e Ferramentas CASE. 2ª ed. Vol. I. Centroatlantico.pt. Portugal, 2005

PostgreSQL - Best way for Password Encryption using pgcrypto's Cryptographic functions:
<https://www.dbrnd.com/2016/03/postgresql-best-way-for-password-encryption-using-pgcrypto-cryptographic-functions/>

Documentação PostgreSQL 9.6.6: <https://www.postgresql.org/docs/9.6/static/>

Documentação Laravel: <https://laravel.com/docs/5.7>

Documentação Bootstrap: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>

Documentação FontAwesome: <https://fontawesome.com/how-to-use/on-the-web/setup/getting-started?using=web-fonts-with-css>

APÊNDICE

I - Guião do coordenador

- Introdução (boas vindas)
- Agradecimentos

Os papéis de coordenador e observador irão ser assegurados por:

Coordenador: Luís Pinho

Observador: Pedro Martinho, Rui Duarte

O Utilizador irá realizar os testes de usabilidade. Para esse efeito será confrontado com vários mockups e será instruído para realizar as tarefas

- Executar as tarefas propostas pelo Coordenador,
- Pensar em voz alta durante a realização de cada tarefa,
- Colocar questões relativas à realização de cada tarefa,
- Desistir do estudo,
- Passar à tarefa seguinte sem concluir a atual.

Tarefas a executar pelos utilizadores

Tabela 1 - Tarefas dos administradores

Nº Tarefa	Nome da tarefa
Tarefa 1	Efetuar o login na plataforma
Tarefa 2	Aceder à lista de máquinas
Tarefa 3	Aceder ao estado do sistema

Tabela 2 - Tarefas dos utilizadores

Nº Tarefa	Nome da tarefa
Tarefa 1	Efetuar o registo na plataforma
Tarefa 2	Efetuar o login na plataforma
Tarefa 3	Selecionar botija 2
Tarefa 4	Comprar botija
Tarefa 5	Visualizar código QR
Tarefa 6	Ver o histórico de compras
Tarefa 7	Visualizar o mapa

II – Grelha de avaliação

Tabela 1 - Grelha de avaliação para a função de administrador

Tarefa		Execução			Comentários Verbais			Erros	Dificuldades	Interpretações Incorretas	Observações
		C D	A D	S D	Positivos	Neutros	Negativos				
1	Efetuar o login na plataforma										
2	Aceder à lista de máquinas										
3	Aceder ao estado do sistema										

Tabela 2 - Grelha de avaliação para a função de utilizador

Tarefa		Execução			Comentários Verbais			Erros	Dificulda- des	Interpretações Incorretas	Observações
		CD	AD	SD	Positivo s	Neutro s	Negativo s				
1	Efetuar o registo na plataforma										
2	Efetuar o login na plataforma										
3	Selecionar a botija 2										
4	Comprar botija										
5	Visualizar código QR										
6	Ver o histórico de compras										
7	Visualizar o mapa com a localização das máquinas										

III – Tarefas a executar

Tabela 1 - Tarefas dos administradores

Nº Tarefa	Nome da tarefa
Tarefa 1	Efetuar o login na plataforma
Tarefa 2	Aceder à lista de máquinas
Tarefa 3	Aceder ao estado do sistema

Tabela 2 - Tarefas dos utilizadores

Nº Tarefa	Nome da tarefa
Tarefa 1	Efetuar o registo na plataforma
Tarefa 2	Efetuar o login na plataforma
Tarefa 3	Selecionar botija 2
Tarefa 4	Comprar botija
Tarefa 5	Visualizar código QR
Tarefa 6	Ver o histórico de compras
Tarefa 7	Visualizar o mapa

IV - Wireframes

Wireframes referidos no ponto 4.1.

Wireframes do utilizador

Pop-up no elemento do carrinho. Mostra o nome dos elementos no carrinho, a sua quantidade, preço total e botões para limpar os conteúdos do carrinho e passar para o checkout.



Figura 31 - Wireframe Detalhes do produto (com pop-up)

1º ecrã do processo de checkout: mostra um resumo dos produtos que serão comprados.

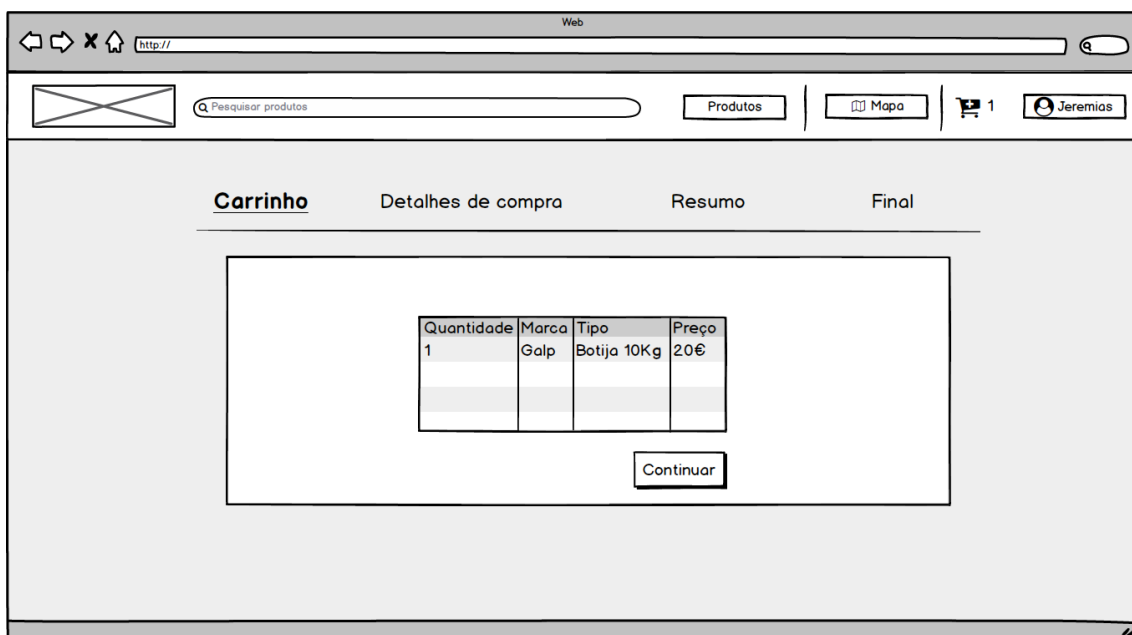


Figura 32 - Wireframe 1º ecrã do checkout

2º ecrã do processo de checkout: permite ao utilizador definir os detalhes desta compra específica, como o método de pagamento, máquina de recolha e se tem ou não para deixar na máquina (para efeitos de ajuste no preço final).

The wireframe shows a web browser window with a navigation bar at the top containing a search bar, 'Produtos', 'Mapa', a shopping cart icon with '1', and a user profile 'Jeremias'. Below the navigation bar is a breadcrumb trail: 'Carrinho' > 'Detalhes de compra' > 'Resumo' > 'Final', with 'Detalhes de compra' highlighted. The main content area contains three sections: 'Método de pagamento' with radio buttons for 'Referência multibanco', 'Cartão de crédito', and 'Cartão físico associado' (selected); 'Compra com entrega de garrafa vazia' with radio buttons for 'Sim' (selected) and 'Não (+2€)'; and 'Pretende escolher um ponto de recolha?' with radio buttons for 'Sim (escolha na lista ao lado)' and 'Não (pode levantar em qualquer ponto de recolha)' (selected). A dropdown menu for 'Lista de pontos de recolha' shows 'Aveiro', 'Guarda', and 'Vila Nova de Gaia'. A 'Continuar' button is at the bottom right.

Figura 33 - Wireframe 2º ecrã de checkout

3º ecrã do processo de checkout: resumo da compra já com todos os detalhes da compra definidos. Mostra também o total a pagar. É o paço final na confirmação de compra.

The wireframe shows the same web browser window as Figure 33, but the breadcrumb trail is 'Carrinho' > 'Detalhes de compra' > 'Resumo' > 'Final', with 'Resumo' highlighted. The main content area contains a table with the following data:

Quantidade	Marca	Tipo	Preço
1	Galp	Botija 10Kg	20€

Below the table, the following details are listed:

- Pagamento: Cartão físico
- Com tara: Sim
- Local de recolha: Não especificado

To the right of these details, a summary of costs is shown:

+	20€
<hr/>	
	Total: 20€

A 'Continuar' button is located at the bottom right.

Figura 34 - Wireframe 3º ecrã de checkout

4º e último ecrã do processo de checkout: confirma que foi feita a compra com sucesso. Apresenta o QR code que irá permitir ao utilizador levantar a garrafa na máquina escolhida (assim como um código que o mesmo poderá usar caso não tenha acesso ao QR code no ato de levantamento). Apresenta ainda um botão que o redireciona para a página inicial.

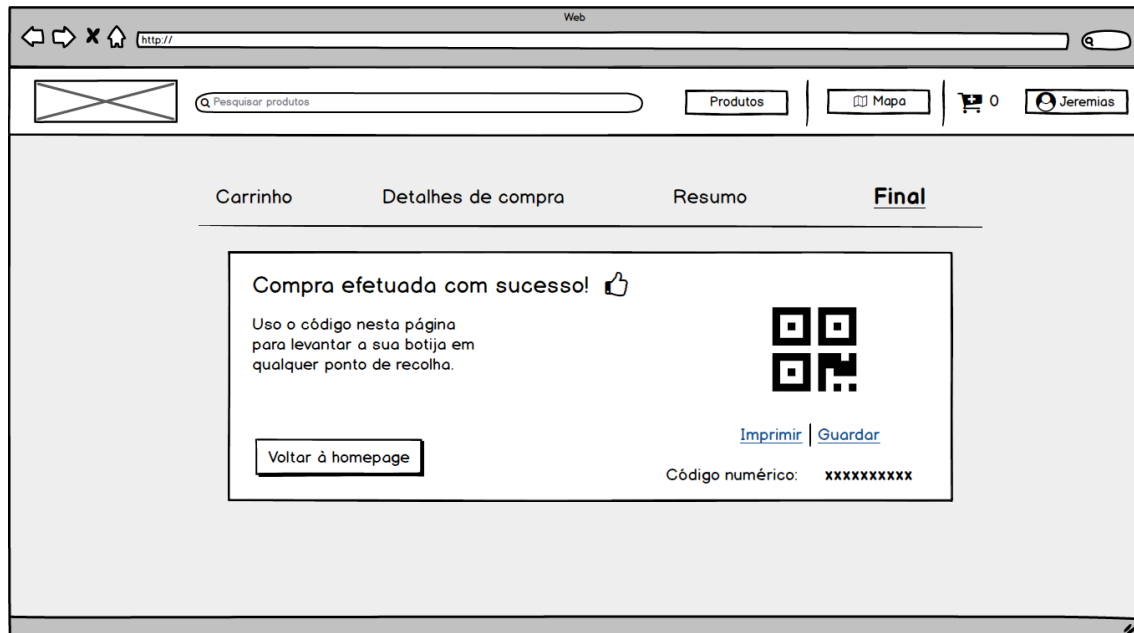


Figura 35 - Wireframe 4º ecrã de checkout

Página inicial com o popup de utilizador aberto. Apresenta as opções que o utilizador poderá selecionar relativamente às suas informações pessoais. Permite entrar na sua conta, gerir os seus cartões (esta opção foi uma das que foi alterada posteriormente no protótipo de alta fidelidade), visualizar o histórico de compras, lista de favoritos, editar o perfil e fazer logout.



Figura 36 - Wireframe Página inicial com pop-up utilizador

Página com o histórico de compras do utilizador. Neste *mockup* é possível visualizar o footer. Este contém um mapa com a localização da sede, morada e contactos da empresa, uma secção com algumas opções informativas da empresa e uma outra com links para o perfil da empresa nas diversas redes sociais em que se encontra.

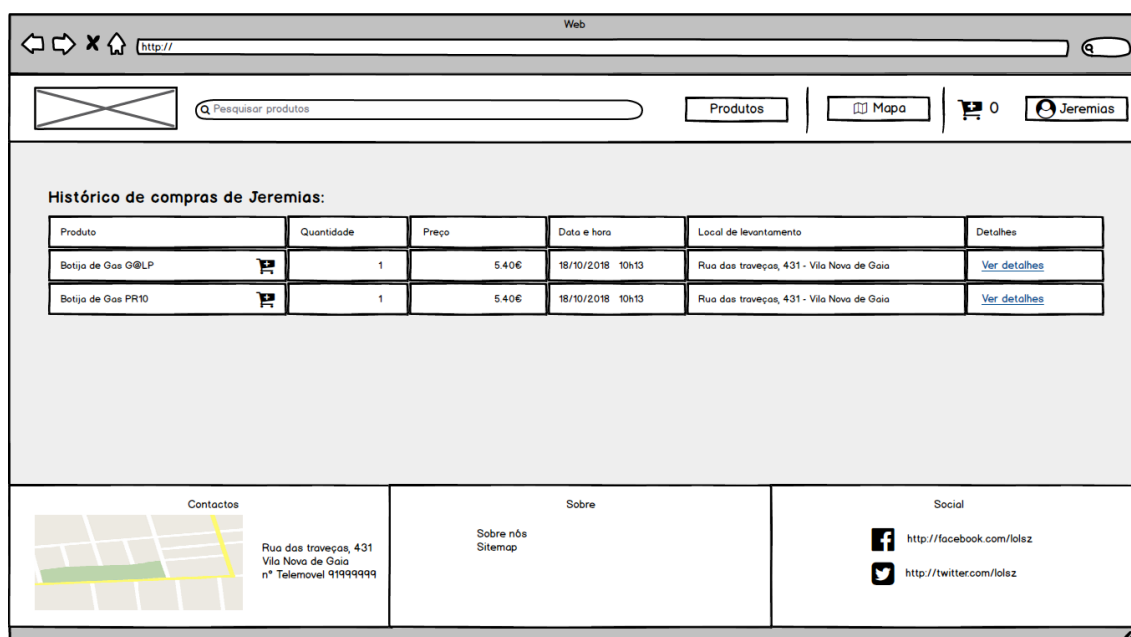


Figura 37 - Wireframe Histórico de compras

Página com a lista das botijas e acessórios favoritos do utilizador.

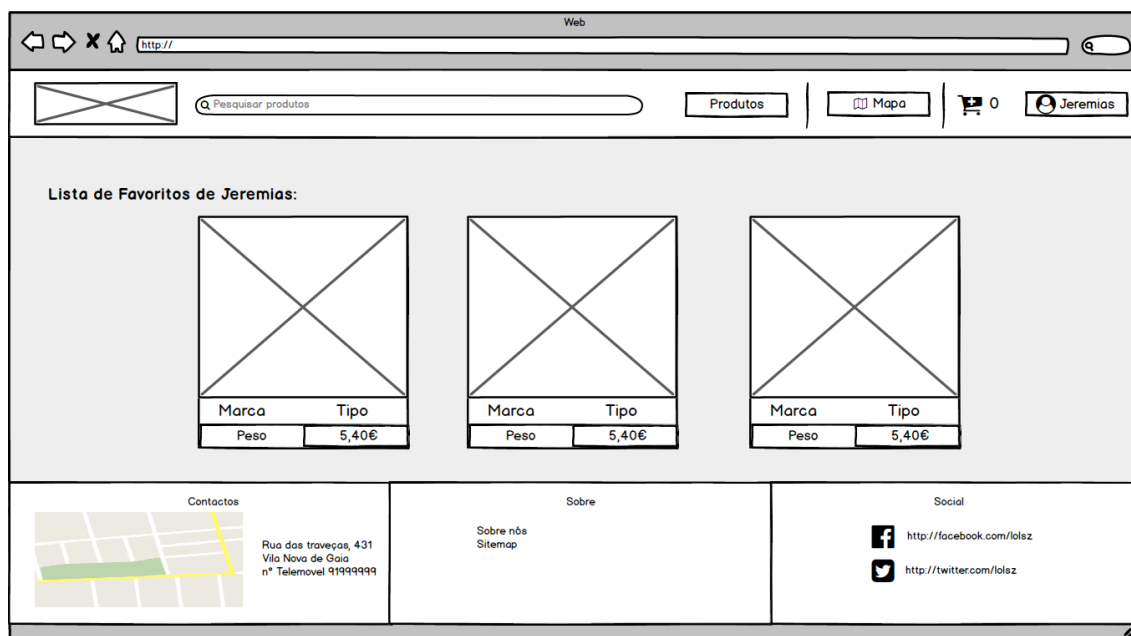


Figura 38 - Wireframe favoritos

Página de perfil do utilizador, onde pode ver as suas informações pessoais (podendo editar as mesmas) e uma opção para editar os seus métodos de pagamento.



Figura 39 - Wireframe detalhes do utilizador

Página que apresenta o mapa onde se encontram as diversas máquinas da plataforma. Ao clicar numa das mesmas, poderá verificar o seu stock, comprar diretamente uma das garrafas disponíveis e visualizar a distância da posição atual do utilizador à máquina em questão. No topo do mapa encontra-se um campo de pesquisa que permitirá ao utilizador “saltar” para uma determinada localização e visualizar todas as máquinas que se encontram lá.

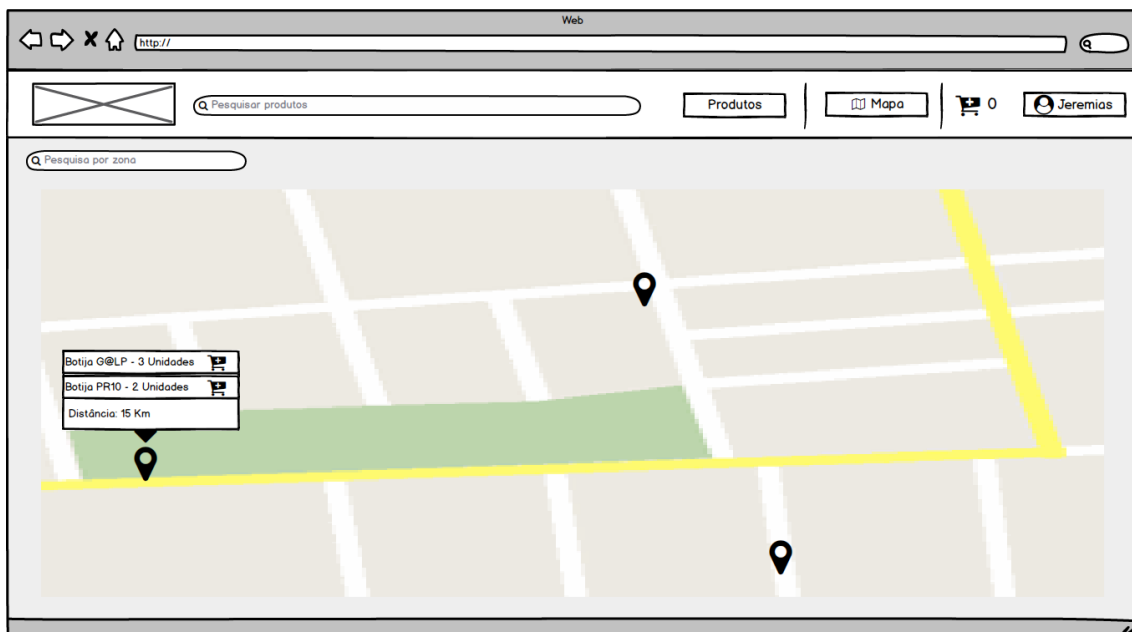


Figura 40 - Wireframe Mapa

Wireframes do admininstrador

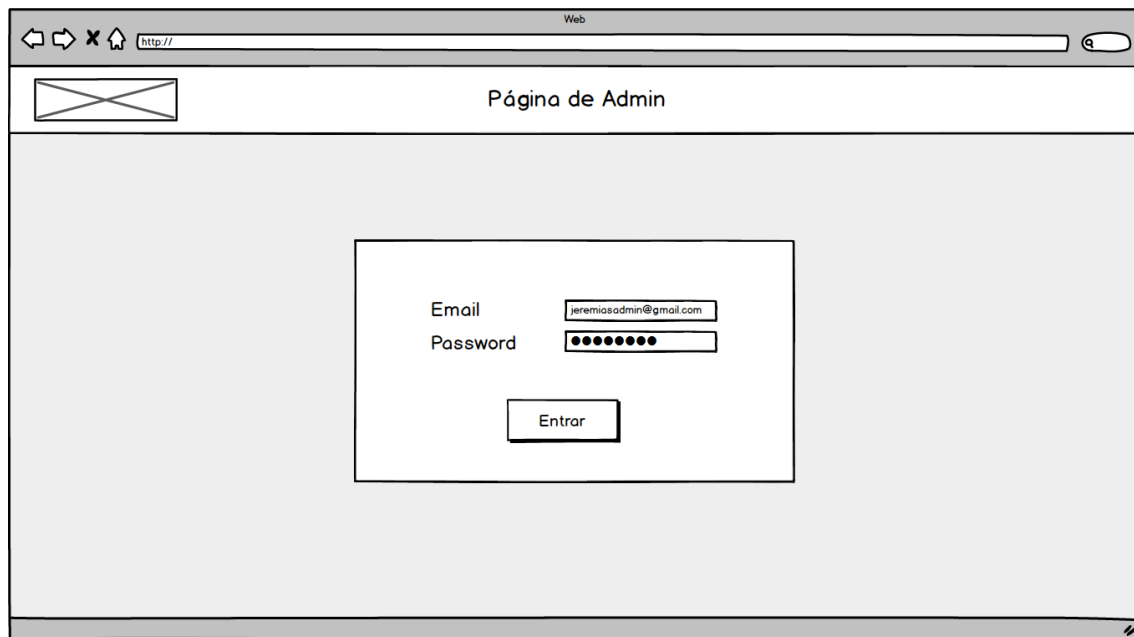


Figura 41 - Wireframe página de login do admin

Página inicial do administrador. Contém as opções mais comuns disponíveis (com outras subopções dentro destas). A navbar presente permite ver alertas geradas e fazer logout.

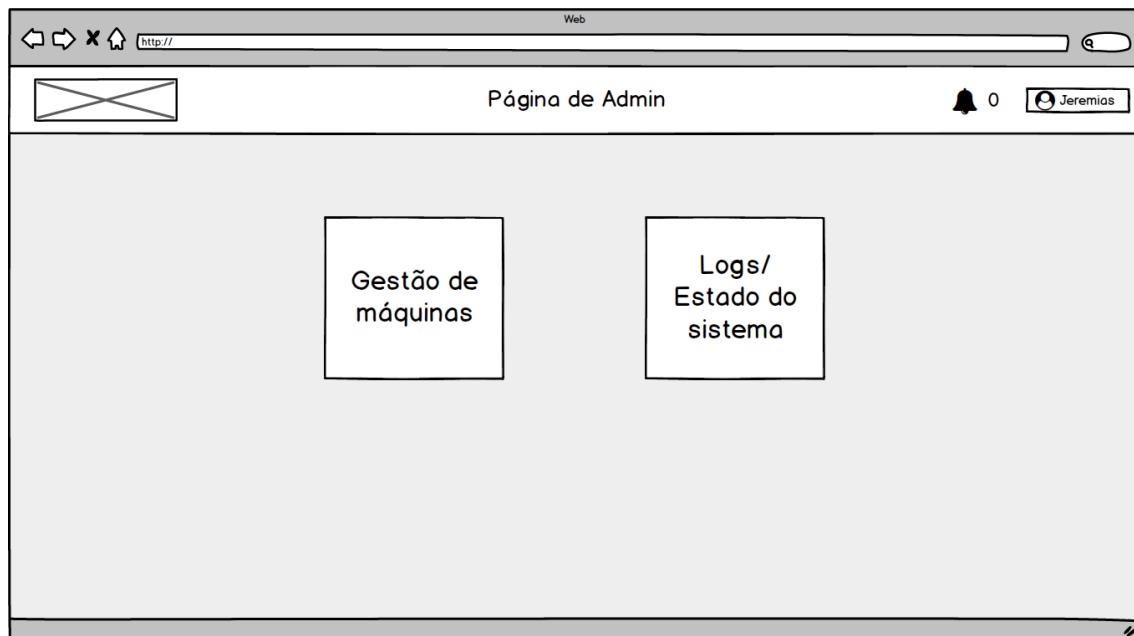


Figura 42 - Wireframe página inicial admin

Página de logs do sistema. Permite ver todas as ocorrências no sistema como falhas de equipamento, rotura de stock nas diversas máquinas, entre outros. É possível ainda verificar o estado geral do sistema e uptime (com mais informações no futuro) e voltar à página inicial.

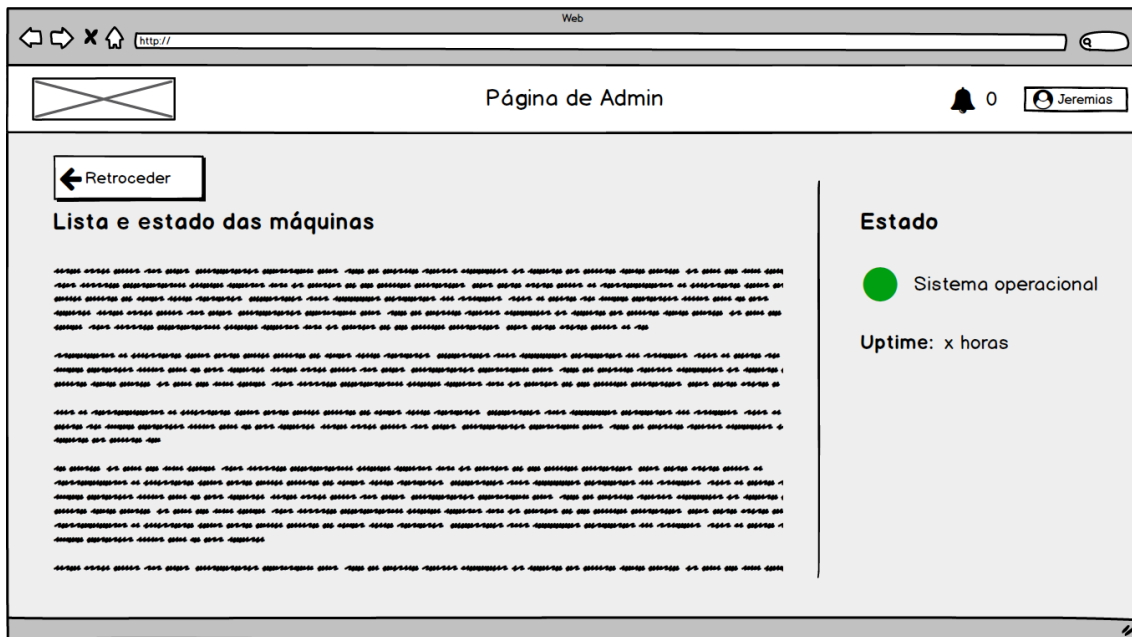


Figura 43 - Wireframe logs das máquinas

Página de listagem das máquinas registadas no sistema. Aqui, é possível visualizar a informação de cada uma, estado e stock. É ainda possível ver o stock da mesma e erros ocorridos. O administrador encontra nesta página a opção para adicionar novas máquinas ao sistema.

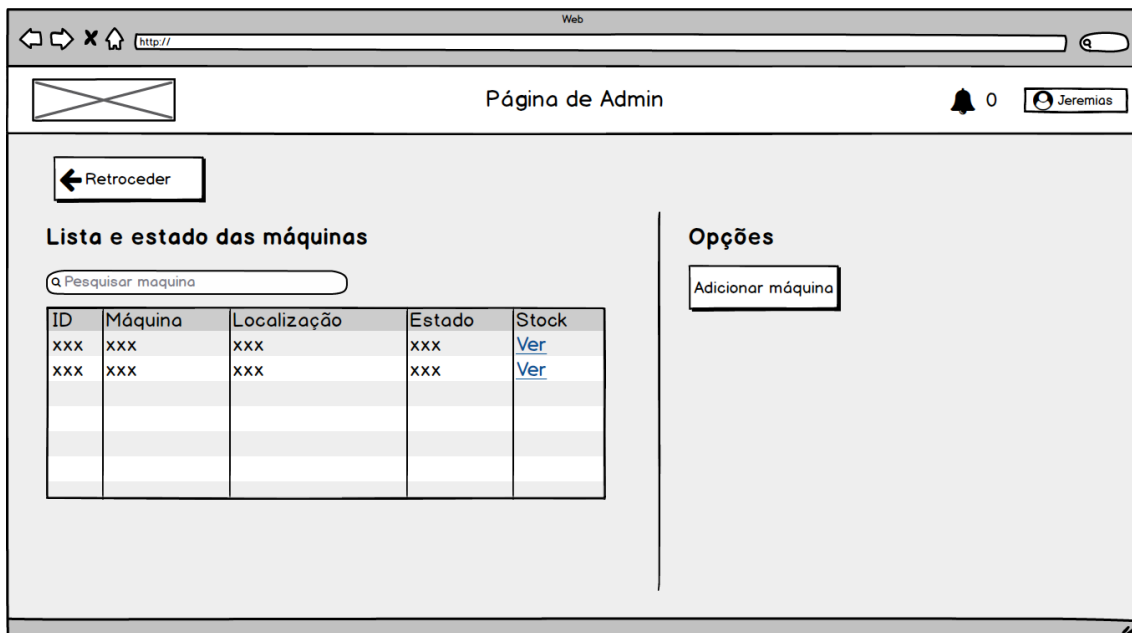


Figura 44 - Wireframe lista das máquinas

V - Screenshots do protótipo de alta fidelidade

VI – Planeamento da Fase 2 e 3 detalhado

Nº	Tarefa	Responsabilidade(%)				
	Início Fase II					
1	Definição e implementação da BD					
1.1	Diagrama do modelo relacional	50	0	0	0	50
1.2	Implementação do SGBD	50	0	0	0	50
2	Definição da camada de apresentação da aplicação Web					
2.1	Html					
2.1.1	Registo	50	0	0	0	50
2.1.2	Login	50	0	0	0	50
2.1.3	Carrinho (5 views)	50	0	0	0	50
2.1.4	Histórico de compras	50	0	0	0	50
2.1.5	Favoritos (2 views)	50	0	0	0	50
2.1.6	Homepage	0	100	0	0	0
2.1.7	Header & Footer	0	100	0	0	0
2.1.8	Contactos	0	0	50	50	0
2.1.9	Mapa	0	0	50	50	0
2.1.10	Sobre	0	0	50	50	0
2.1.11	Perfil (2 views)	50	0	0	0	50
2.1.12	Detalhes do produto	0	100	0	0	0
2.1.13	Integração das views e criação de rotas	0	100	0	0	0
3	CSS					
	Integração do template	0	100	0	0	0
	Style Guide	0	0	50	50	0
4	Elaboração de relatório intermédio	20	20	20	20	20
	Entrega Fase II					
		António Pinho	João Novo	Luís Pinho	Pedro Martinho	Rui Duarte