



UNIVERSIDADE  
LUSÓFONA

# Price is Just

## Trabalho Final de Curso

Relatório Final

Sara Diogo

Orientador: Prof. Fernando Teodósio

Trabalho Final de Curso | LEI | 25-06-2021

## **Direitos de cópia**

Price is Just, Copyright de Sara Diogo, ULHT.

A Escola de Comunicação, Arquitetura, 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 objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

---

## Resumo

Atualmente verificamos que as inovações tecnológicas não param de emergir e ganham cada vez mais espaço na sociedade. Como tal, este Trabalho Final de Curso tem como objetivo desenvolver uma aplicação que permite aos utilizadores verificar em que supermercado, à sua volta, a sua lista de compras é mais barata. Esta aplicação é disponibilizada em multiplataformas, permitindo aos utilizadores pesquisar e comparar produtos nos diversos supermercados, calcular em que supermercado o produto se encontra mais barato, tal como a possibilidade de criar uma lista de compras apresentando logo o total da mesma. Pretende-se que esta solução tenha uma forte aplicabilidade no processo de compras dos consumidores, antecipando as necessidades dos mesmos e levando-os a economizar dinheiro.

**Palavras-Chave:** Projeto Final, Aplicação, Web, Mobile, iOS, Android, Multiplataforma, Supermercados, Compras, Produtos, Preço mais baixo, Poupar.

## **Abstract**

Currently we see that technological innovations are constantly emerging and gaining more and more relevance in society. As such, this Final Course Work aims to develop an application that allows users to check which supermarket, around them, their shopping list is less expensive on.

This application is available in multiplatform, allowing users to search and compare products in different supermarkets, calculate in which supermarket the product is cheaper, as well as the possibility of creating a shopping list showing the total of it right away. It is intended that this solution has a strong applicability in the purchasing process of consumers, anticipating their needs and leading them to save money.

**Keywords:** Final Project, Application, Web, Mobile, iOS, Android, Multiplatform, Supermarkets, Shopping, Products, Lowest price, Saving, Savings.

---

# Índice

Resumo .....	iii
Abstract.....	iv
Índice .....	v
Lista de Figuras .....	vi
Lista de Tabelas .....	vii
1 Identificação do Problema.....	1
2 Levantamento e Análise de Requisitos .....	4
3 Viabilidade e Pertinência .....	10
4 Solução Desenvolvida .....	20
5 Benchmarking .....	31
6 Método e Planeamento .....	35
7 Resultados .....	37
8 Conclusão e Trabalhos Futuros .....	38
Bibliografia.....	39
Apêndice A – Questionário e Respetivas Respostas .....	41
Apêndice B - Tarefas dos Testes de Usabilidade .....	46
Apêndice C - Protótipos Não Funcionais .....	47
Apêndice D – Questionário de Satisfação .....	55
Apêndice E – Resultados do Questionário de Satisfação .....	56
Apêndice F – Resultado Final .....	58
Glossário.....	63

## Lista de Figuras

Figura 1. Diagramas dos Casos de Uso das funcionalidades do sistema.....	8
Figura 2. Desenho da Base de Dados.....	9
Figura 3. Análise Swot.....	12
Figura 4. Exemplo do teste utilizando a ferramenta Page Speed Insights.....	17
Figura 5. Mapa de acessibilidade consoante tipo de utilizador.....	18
Figura 6. Arquitetura MVC implementada por Laravel.....	20
Figura 7. Exemplo de uma Route.....	22
Figura 8. Exemplo de um Controller.....	23
Figura 9. Exemplo de uma View.....	23
Figura 10. Exemplo de Migration.....	24
Figura 11. Exemplo de Model.....	24
Figura 12. Exemplo de uma Factory.....	25
Figura 13. Exemplo de uma Seeder.....	25
Figura 14. Configuração da base de dados.....	26
Figura 15. Ficheiro .env de configuração.....	26
Figura 16. - Execução do comando para migrar o modelo de dados.....	27
Figura 17. Exemplo de Funcionalidades do Voyager.....	28
Figura 18. Representação genérica de Aplicação Multiplataformas.....	29
Figura 19. Funcionalidade do Google Maps em Mobile e Web.....	29
Figura 20. Implementação do filtro de supermercados.....	30
Figura 21. Website KuantoKusta Supermercados sem resultados.....	34
Figura 22. Comparação de produto: Price is Just vs Save Cook.....	34
Figura 23. Cronograma com respetiva calendarização.....	36
Figura 24. Cronograma Final.....	36
Figura 25. Protótipo Não Funcional - Mobile.....	50
Figura 26. Protótipo Não Funcional - Website.....	54
Figura 27. Questionário de Satisfação.....	55
Figura 28. Home Page.....	58
Figura 29. Registo de Utilizador.....	58
Figura 30. Lista de Produtos.....	59
Figura 31. Descrição do Produto.....	59
Figura 32. Registar Novo Produto.....	60
Figura 33. Lista de Compras.....	60
Figura 34. Estatísticas de Consumo.....	60
Figura 35. Home Page e Lista de Produtos.....	61
Figura 36. Registo de Novo Produto e Lista de Compras.....	61
Figura 37. Estatísticas de Consumo e Reportar Erro.....	62

---

## Lista de Tabelas

Tabela 1. Requisitos Funcionais com respetiva prioridade e grau de cumprimento.....	7
Tabela 2. Requisitos Não Funcionais com respetivo grau de prioridade.....	8
Tabela 3. Análise quantitativa das frustrações dos entrevistados.....	10
Tabela 4. Análise quantitativa em relação à utilização da aplicação.....	11
Tabela 5. Testes Funcionais.....	16
Tabela 6. Testes Não Funcionais.....	16
Tabela 7. Lista de comandos mais utilizados.....	21
Tabela 8. Lista de Métodos disponibilizados e respetivas relações.....	22
Tabela 9. Análise do Mercado - Comparação Inicial.....	33
Tabela 10. Análise do Mercado - Comparação Final.....	33
Tabela 11. Respostas às entrevistas.....	45



# 1 Identificação do Problema

Neste capítulo é feita uma abordagem inicial ao tema, identificando os possíveis problemas e frustrações que as pessoas sentem quando realizam de compras, servindo como base para que a solução desenvolvida neste TFC <sup>1</sup>seja relevante.

Ao longo do tempo, é de notar que as tecnologias têm apresentado inúmeras mudanças em vários segmentos da sociedade e conseqüentemente na vida das pessoas. A possibilidade de instalar aplicações que auxiliam desde as tarefas mais simples a tarefas mais complexas, tem vindo a mudar constantemente a maneira como as pessoas comunicam, a forma como se relacionam, trabalham etc. Atualmente as aplicações web/mobile, assumem uma nova postura permitindo tornar as empresas mais próximas dos consumidores, bem como a possibilidade de haver uma comunicação direta e constante com os clientes, podendo a mesma ser feita a qualquer hora e em qualquer lugar.

Desenvolveu-se então uma aplicação que permite aos utilizadores, saberem onde encontrar os produtos com preços mais económicos, nos diversos supermercados ao seu redor, oferecendo assim uma maior comodidade na hora de comparar preços sem ser necessário visitar os estabelecimentos, ou consultar folhetos, visando também oferecer uma forma simples de economizar tempo e dinheiro, localizando sempre as melhores oportunidades e ofertas.

Para verificar se a hipótese de que esta aplicação é útil, criou-se um processo de validação/invalidação da mesma, recorrendo a entrevistas, junto de pessoas que fazem compras semanalmente num supermercado. Explicitamente a hipótese é: “As pessoas estão dispostas a pagar por uma aplicação que lhes permite poupar tempo e dinheiro em compras de supermercado, mesmo que isso implique irem a outro supermercado à sua volta?”.

Sendo então este processo constituído pelos quatro passos seguintes:

1. Criar uma lista de pessoas que fazem compras todas as semanas.

---

<sup>1</sup> Trabalho Final de Curso

2. Entrevistar as respetivas pessoas, com perguntas que permitam validar/invalidar a nossa hipótese. Sendo as perguntas as seguintes:
  - a. Qual é a tua maior frustração em relação a ir às compras?
  - b. Imagina que tinhas uma aplicação que te permitia submeter uma lista de compras e te dava o supermercado, à tua volta, onde essa lista é mais barata. Estarias disposto a ir a esse supermercado em vez do teu habitual, mesmo não sabendo onde é que os produtos estão e perder um pouco mais tempo à procura deles, para poupar mais dinheiro?
3. Registrar respostas numa tabela.
4. Registrar as conclusões.

O problema proposto para a realização desta aplicação, consistiu então no dinheiro desnecessário que as pessoas gastam em compras, por não saberem onde encontrar os produtos mais baratos. Desta forma o objetivo é não só eliminar um dos pontos de dor na ótica do consumidor, mas também fazer com que a satisfação deste aumente e naturalmente veja motivos para repetir a nova experiência antes do processo de compra.

Face ao problema exposto, o resultado obtido foi o esperado, contendo apenas algumas diferenças em relação ao proposto inicialmente. Seguindo-se então a respetiva análise comparativa:

Inicialmente os utilizadores iriam ser capazes de fazer uma avaliação dos produtos na aplicação, poderiam estabelecer limites a gastar diariamente/ semanalmente ou mensalmente nas compras, poderiam também averiguar se o produto estava ou não disponível na loja e receber notificações, no entanto, estes pontos não foram concretizados. Em relação ao nome do projeto e às tecnologias utilizadas, também sofreram alterações. Para o primeiro nome sugerido não existia domínio disponível e em relação às tecnologias, tinham outro nível de complexidade.

Quanto ao que foi realizado e que anteriormente nunca foi mencionado, foi o facto de ser possível os utilizadores reportarem erros nos preços dos produtos, a criação de uma tabela (productCopy) que permite a não duplicação de produtos, quando estes existem em vários supermercados, permitindo um menor gasto de recursos, uma maior

rapidez e para não se fazer alterações nos produtos originais. Tal como a existência de uma página de termos e condições, bem como a utilização de um plugin<sup>2</sup> que permite eficazmente fazer a administração da aplicação, de forma facilmente expansível e “user-friendly”.

No capítulo seguinte são descritas as características da solução, sendo feito o levantamento e análise de requisitos.

---

<sup>2</sup> <https://voyager.devdojo.com>

## 2 Levantamento e Análise de Requisitos

De forma a entender as necessidades e desejos dos consumidores, no capítulo anterior foi demonstrado o problema e hipótese de estudo, sendo que neste capítulo são então apresentadas as características da solução, sendo feito um levantamento e análise de requisitos funcionais e não funcionais.

- **RF001 – Criar utilizador:** Deve ser possível o utilizador registar-se e fazer login.  
**Critério de aceitação:** Após o registo o utilizador consegue aceder à sua conta na aplicação.
  
- **RF002 – Pesquisa de Produtos:** O utilizador registado deve conseguir consultar produtos e os supermercados onde estão mais baratos.  
**Critério de Aceitação:** A pesquisa origina produtos, os seus preços e o supermercado (dentro de um raio à volta do utilizador) onde está mais barato. Clicando num produto, o utilizador pode ver detalhes, incluindo os preços deste produto noutros supermercados.
  
- **RF003 – Consultar Lista de Compras:** O utilizador deve conseguir criar, editar e eliminar listas de compras com os produtos registados no sistema.  
**Critério de Aceitação:** O utilizador é capaz de criar e eliminar listas de compras e adicionar produtos às listas criadas.
  
- **RF004 – Visualizar estatísticas de consumo:** O utilizador deve conseguir ver as suas estatísticas de gastos e poupanças mensais/semanais/diários/média por visita ao supermercado.  
**Critério de Aceitação:** Após registar uma ida às compras, o utilizador deve conseguir ver um painel de estatísticas de gastos e poupanças, assim como o seu histórico.

- **RF005 – Visualizar produtos por categoria:** O utilizador deve conseguir aceder aos produtos através de uma categoria.  
**Critério de aceitação:** Ao pesquisar por uma categoria, serão mostrados ao utilizador produtos dessa mesma categoria (Exemplo: Lacticínios mostrará leites, manteigas, queijos, etc.).
  
- **RF006 – Localização:** A aplicação deve ter acesso à localização do utilizador e permitir-lhe definir um raio de inclusão de supermercados (em km).  
**Critério de aceitação:** A aplicação envia um pedido de autorização para aceder à sua localização, que após aceite, grava-a no perfil do utilizador, assim como um raio por omissão de inclusão de supermercados. Esta definição reflete-se na consideração de supermercados.
  
- **RF007- Gerir conta:** Deve ser possível o utilizador gerir os seus dados, alterando/atualizando as suas definições.  
**Critério de aceitação:** Após estar registado o utilizador consegue aceder às suas definições.
  
- **RF008 – Registo de novos produtos:** O utilizador registado deve conseguir fazer o registo de novos produtos.  
**Critério de Aceitação:** A pesquisa origina produtos, os seus preços e o supermercado onde está mais barato, caso o produto ainda não exista, o utilizador pode adicionar o mesmo, com o seu respetivo preço e detalhes.
  
- **RF009 – Edição de preço do produto:** O utilizador registado deve conseguir alterar o preço de um produto.  
**Critério de Aceitação:** A pesquisa origina produtos, os seus preços e o supermercado onde está mais barato, o utilizador pode modificar o preço do produto num determinado supermercado.
  
- **RF010 – Reportar produto:** O utilizador registado deve conseguir reportar um erro existente num produto, tal como o preço, título entre outros.

- **Critério de Aceitação:** Ao efetuar a pesquisa de um produto, são mostrados os preços e o supermercado onde está mais barato. O utilizador ao verificar que existe um erro no produto, pode efetuar um pedido de análise ao administrador.
  
- **RF011 – Consultar valor total na lista de Compras:** O utilizador deve conseguir consultar o preço total dos produtos que adicionou à lista de compras.  
**Critério de Aceitação: Prioridade:** A pesquisa origina produtos, os seus preços e o supermercado onde está mais barato, o utilizador pode acrescentar os mesmos à lista de compras, sendo possível ver o total dos produtos adicionados.
  
- **RF012 – Limite de gastos:** O utilizador deve conseguir definir um limite de dinheiro a gastar mensais/semanais/diários.  
**Critério de aceitação:** Com as listas de compras criadas o utilizador pode ter acesso ao valor já gasto ou a gastar e pode estabelecer um valor limite.
  
- **RF013 – Imprimir lista de compras:** O utilizador deve conseguir imprimir a lista de compras criada.  
**Critério de aceitação:** O utilizador registado consegue aceder às suas listas de compras criadas e pode imprimir as mesmas.
  
- **RF014 – Notificações:** O utilizador deve conseguir receber notificações para ir às compras, dos gastos nas compras etc.  
**Critério de aceitação:** O utilizador registado consegue receber alertas.
  
- **RF015 – Logout:** Deve ser possível o utilizador sair da aplicação.  
**Critério de aceitação:** Após o registo o utilizador consegue aceder à sua conta na aplicação e deve conseguir sair do sistema sempre que quiser.

Requisitos Funcionais			
ID	Requisito	Prioridade	Cumprimento
RF001	Criar utilizador: Deve ser possível o utilizador registar-se e fazer login.	Alta	Integral
RF002	Pesquisa de Produtos: O utilizador registado deve conseguir consultar produtos e os supermercados onde estão mais baratos.	Alta	Integral
RF003	Consultar Lista de Compras: O utilizador deve conseguir criar, editar e eliminar listas de compras com os produtos registados no sistema.	Alta	Integral
RF004	Visualizar estatísticas de consumo: O utilizador deve conseguir ver as suas estatísticas de gastos e poupanças mensais/semanais/diários/média por visita ao supermercado.	Alta	Integral
RF005	Visualizar produtos por categoria: O utilizador deve conseguir aceder aos produtos através de uma categoria.	Alta	Integral
RF006	Localização: A aplicação deve ter acesso à localização do utilizador e permitir-lhe definir um raio de inclusão de supermercados (em km).	Alta	Parcial
RF007	Gerir conta: Deve ser possível o utilizador gerir os seus dados, alterando/atualizando as suas definições.	Alta	Integral
RF008	Registo de novos produtos: O utilizador registado deve conseguir fazer o registo de novos produtos.	Alta	Integral
RF009	Edição de preço do produto: O utilizador registado deve conseguir alterar o preço de um produto.	Alta	Integral
RF010	O utilizador registado deve conseguir reportar um erro existente num produto, tal como o preço, título entre outros.	Alta	Integral
RF011	Consultar valor total na lista de Compras: O utilizador deve conseguir consultar o preço total dos produtos que adicionou à lista de compras.	Alta	Integral
RF012	Limite de gastos: O utilizador deve conseguir definir um limite de dinheiro a gastar mensais/semanais/diários.	Média	Não cumprido
RF013	Imprimir lista de compras: O utilizador deve conseguir imprimir a lista de compras criada.	Média	Integral
RF014	Notificações: O utilizador deve conseguir receber notificações.	Baixa	Não cumprido
RF015	Logout: Deve ser possível o utilizador sair da aplicação.	Alta	Integral

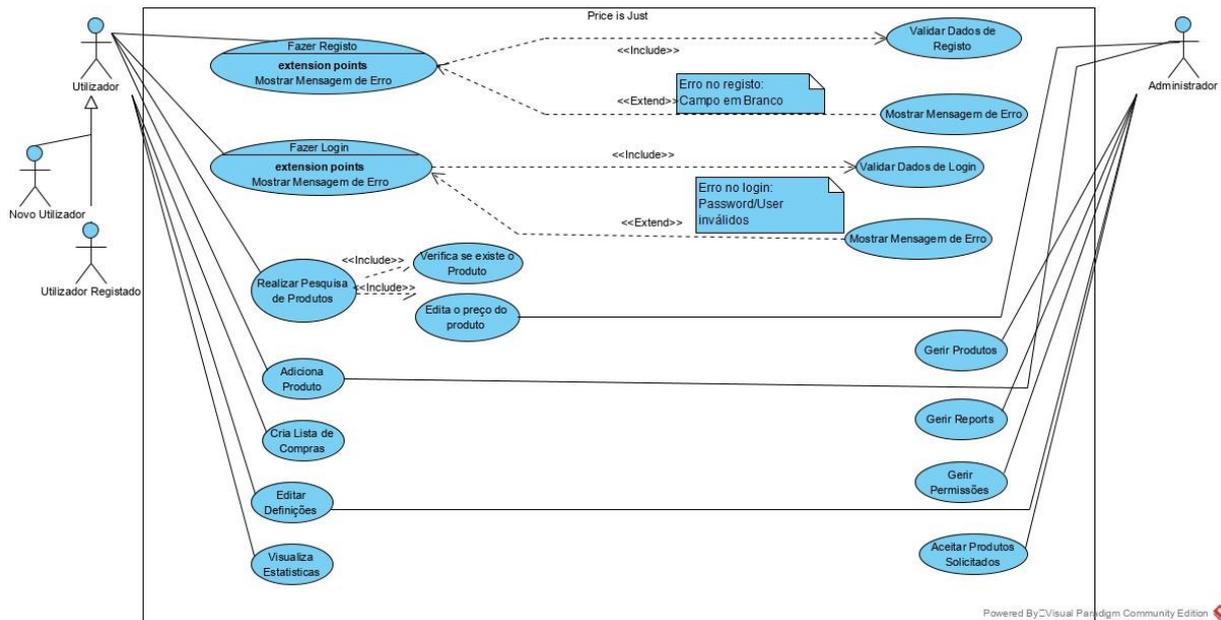
**Tabela 1. Requisitos Funcionais com respetiva prioridade e grau de cumprimento.**

Requisitos Não Funcionais		
ID	Requisito	Prioridade
RFN01	Servidor disponível integralmente	Alta
RFN02	A aplicação deve permitir uma navegação simples, lógica e intuitiva com um menu que disponibilize as informações necessárias ao utilizador.	Alta
RFN03	A aplicação deve disponibilizar igual informação e funcionalidades a todos os utilizadores.	Alta
RFN04	A aplicação apenas deve permitir o acesso às áreas do menu de navegação, a utilizadores previamente registados.	Alta
RFN05	A aplicação deve permitir uma utilização permanente, em qualquer hora do dia e dia da semana.	Alta
RFN06	A aplicação deve permitir que as suas operações sejam efetuadas com rapidez.	Alta

<b>RFN07</b>	A aplicação deve permitir que cada utilizador só tenha acesso aos seus dados e todas as palavras-passe devem estar cifradas.	Alta
<b>RFN08</b>	A base de dados deve estar protegida de acessos não-autorizados.	Alta

**Tabela 2. Requisitos Não Funcionais com respetivo grau de prioridade.**

Definidos os requisitos é necessária uma representação das funcionalidades do sistema, assim como as ações que o utilizador tem com o mesmo. Como tal, abaixo são então apresentados os diagramas de caso de uso das funcionalidades do sistema.



**Figura 1. Diagramas dos Casos de Uso das funcionalidades do sistema.**

Após a definição dos casos de uso é possível através dos Diagramas de Entidade Relação, identificar as entidades, atributos e respetivas relações.

- **Users** – Classe representativa de um utilizador, sendo que a cada um, estão associadas várias permissões.
- **Products**- Classe representativa de um produto base, que contem apenas as informações básicas de um produto, como o id, título, descrição e imagem.
- **Product Copy** – Classe representativa de um produto base que permite a existência do mesmo produto em vários supermercados sem a necessidade de produtos duplicados na base de dados.
- **Categories** - Classe representativa de uma categoria.

- **Supermarkets** – Classe representativa de um supermercado.
- **User Has Supermarkets** – Classe representativa dos supermercados favoritos do utilizador.
- **Shopping List** - Classe representativa de uma lista de compras.
- **Shopping Items** – Classe representativa de um produto na lista de compras.

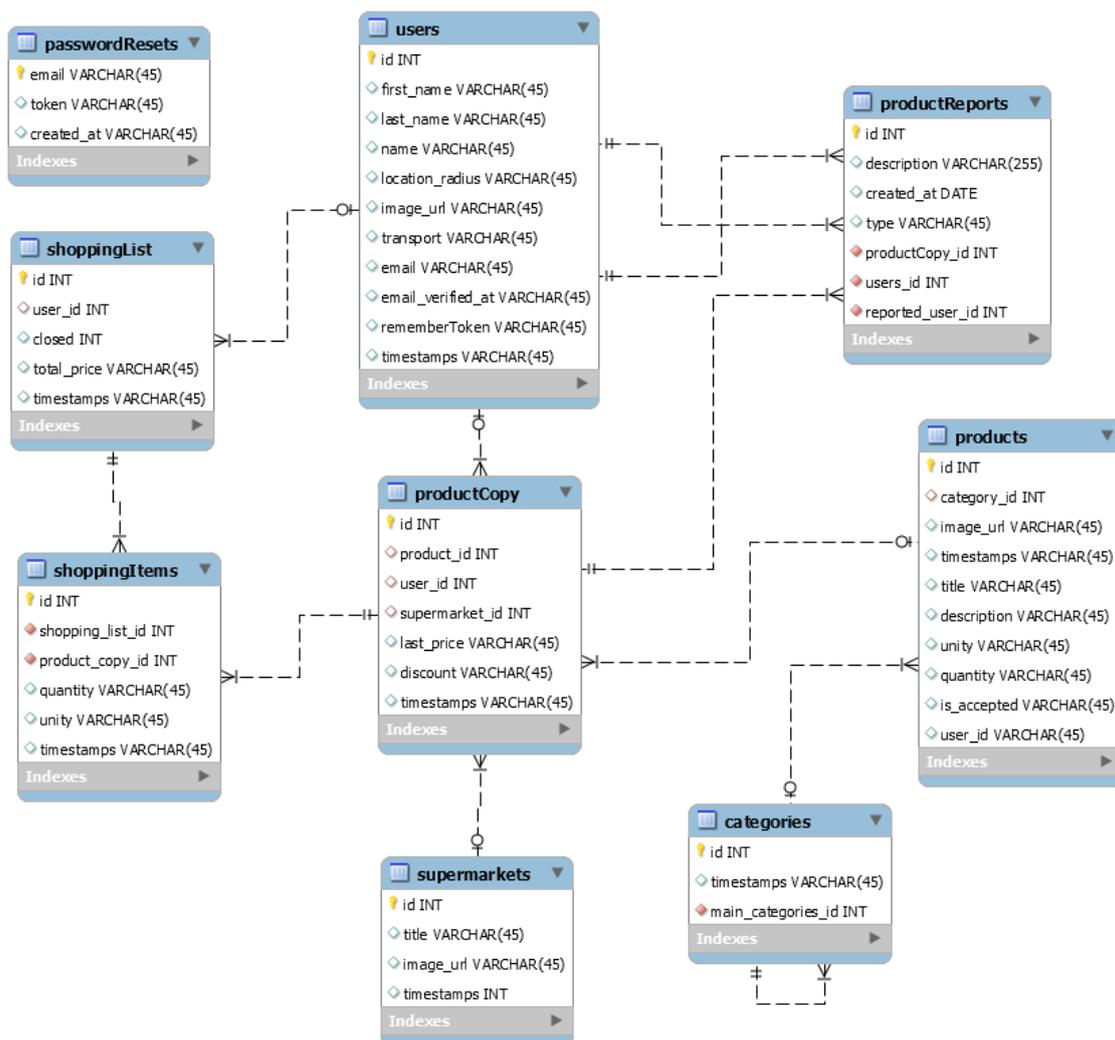


Figura 2. Desenho da Base de Dados.

A base de dados foi desenhada e desenvolvida em paralelo com os modelos/entidades garantindo a consistência necessária para o sucesso das migrações das tabelas de dados.

No capítulo seguinte são apresentadas as respostas das entrevistas bem como as conclusões tiradas, de forma a comprovar a viabilidade e pertinência deste projeto.

### 3 Viabilidade e Pertinência

No capítulo anterior foram apresentadas as características da solução e respetivo levantamento e análise de requisitos funcionais e não funcionais. Sendo que neste capítulo são então apresentadas as respetivas conclusões das entrevistas, é feita uma análise SWOT<sup>3</sup>, de forma a recolher informações pertinentes para que a solução desenvolvida seja viável, tal como é apresentado o resultado dos testes realizados.

Foram entrevistadas 42 pessoas, cujas respostas se encontram no [Apêndice A](#). Relativamente à primeira questão, podemos dividir as frustrações dos entrevistados em 4 categorias, sendo elas as seguintes:

- Tempo perdido;
- Dinheiro mal gasto;
- Falta de informação;
- Falta de organização.

A partir de cada categoria é possível analisar a quantidade de pessoas que sente tal frustração:

**Tabela 3. Análise quantitativa das frustrações dos entrevistados.**

Frustrações	Número de Respostas
<b>Tempo Perdido</b>	
Filas de espera	19
Procura de produtos	11
Procura do produto mais barato	1
<b>Dinheiro mal gasto</b>	
Não saber onde estão os produtos mais baratos	8
Não saber quando é que o produto está em promoção	5
<b>Falta de informação</b>	

---

<sup>3</sup> Análise SWOT - [https://www.mindtools.com/pages/article/newTMC\\_05.htm](https://www.mindtools.com/pages/article/newTMC_05.htm)

Não saber se o produto existe/ está esgotado	6
Falta de informação qualitativa dos produtos	1
Falta de preço/preço incorreto	1
Falta de critério de escolha de produto explícito	1
<b>Falta de organização</b>	
Falta de lista de compras	1

Podemos então concluir, com base nesta amostra, que as maiores frustrações dos entrevistados, são:

- Perca de tempo em filas de espera;
- Perca de tempo à procura dos produtos no supermercado;
- Não saber onde encontrar os produtos mais baratos;
- Não saber se os produtos estão em promoção;
- Não saber se os produtos existem em stock no respetivo supermercado.

Relativamente à segunda questão colocada, podemos concluir que 32 entrevistados estariam dispostos a utilizar uma aplicação que lhes que permitisse submeter uma lista de compras e ter acesso a preços mais vantajosos num supermercado ao seu redor, 8 pessoas não estariam interessadas e apenas 2 pessoas ficaram reticentes.

**Tabela 4. Análise quantitativa em relação à utilização da aplicação.**

Opção	Número de Respostas
Sim	32
Não	8
Talvez	2

As novas tecnologias estão em constante evolução, fazendo com que a comunicação seja mais fluída e que a informação seja partilhada quase instantaneamente. Como tal, e tendo um grande impacto na sociedade, as empresas

procuram cada vez mais projetar os seus negócios através destes meios, sendo necessário existir novas e modernas ideias de forma a potenciar os produtos ou serviços.

As aplicações móveis/web vieram permitir ao utilizador estar em contacto com uma determinada empresa a qualquer momento. Ou seja, podemos estar a um clique de sabermos informações sobre um produto ou serviço, podemos usufruir de promoções e serviços não disponíveis noutras plataformas etc. Assim sendo, e pelo facto das aplicações estarem a crescer e a imporem-se no mercado, é importante que estas sejam estudadas e analisadas de forma a que consigam chegar aos consumidores/clientes finais. Como tal, é necessária uma gestão estratégica para que sejam tomadas decisões que determinem o bom ou mau desempenho. Através de uma análise SWOT (“*Strenghts*” - Pontos Fortes; “*Weaknesses*” - Pontos Fracos; “*Opportunities*” - Oportunidades e “*Threats*” - Ameaças) procurou-se identificar os principais aspetos que caracterizam a posição estratégica, tanto a nível interno como externo da aplicação “Price is Just”.



Figura 3. Análise Swot

Após uma detalhada análise das entrevistas e análise SWOT, verifica-se que estamos perante um ambiente favorável para a implementação de uma aplicação que permita às pessoas poupar dinheiro nas suas compras. Verifica-se também que existe uma predominância de pontos fortes em relação aos pontos fracos o que se revela como uma mais-valia. Contudo, será necessário ter em conta o grau de ameaças detetadas e estabelecer algumas estratégias para as contornar ou minimizar.

No caso da aplicação “Price is Just<sup>4</sup>”, visto ser uma aplicação nova, a estratégia a adotar é a de inserção no mercado existente, tendo como objetivo angariar novos clientes e convencer os não utilizadores a adquirirem o produto. Esta aplicação insere-se no mercado nacional, tendo como público-alvo todas as pessoas que efetuem compras nos supermercados e tencionem poupar dinheiro, tendo então como missão o bem-estar e qualidade de vida das pessoas, visto estarem a poupar dinheiro.

De forma a detetar erros e defeitos na aplicação, foram também realizados alguns testes, sendo estes elementos indispensáveis para a garantia da qualidade do software e representam uma revisão final das especificações.

Neste trabalho foram realizados os seguintes testes:

- **Testes de Validação (Funcional e Não Funcional)** – De forma a comprovar se a aplicação desenvolvida satisfaz os requisitos funcionais e não funcionais:

1	<b>Tipo de teste</b>	Funcional
	<b>Subtipo de teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de login dos utilizadores.
	<b>Requisito</b>	RF001 – Criar utilizador
	<b>Test Case</b>	1) O sistema apresenta um formulário para registo do utilizador: nome, e-mail e palavra-passe; 2) Utilizador informa os seus dados; 2.1) Validação dos dados introduzidos pelo utilizador; 3) Sistema guarda na base de dados; 4) Criação de novo utilizador; 5) Utilizador adiciona credenciais válidas -> entra na aplicação; 6) Utilizador adiciona credenciais erradas -> bloqueia acesso.

<sup>4</sup> <http://priceisjust.com>

2	<b>Tipo de teste</b>	Funcional
	<b>Subtipo de teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de pesquisa de produtos.
	<b>Requisito</b>	RF002 – Pesquisa de Produtos
	<b>Test Case</b>	1) O sistema apresenta um menu de pesquisa; 2) Utilizador informa os produtos que quer pesquisar; 3) Se produtos existirem -> aparece ao utilizador; 4) Se produtos não existir -> aparece mensagem que produto não existe.
3	<b>Tipo de teste</b>	Funcional
	<b>Subtipo de teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade da Lista de Compras
	<b>Requisito</b>	RF003 - Consultar Lista de Compras
	<b>Test Case</b>	1) O utilizador pesquisa por um produto; 2) O utilizador adiciona produto à lista de compras; 3) O utilizador consulta o produto adicionado à lista de compras; 4) O sistema apresenta todas as listas de compras criadas pelo utilizador.
4	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade da opção de Estatísticas
	<b>Requisito</b>	RF004 – Visualizar estatísticas de consumo
	<b>Test Case</b>	1) O utilizador tem uma ou várias listas de compras; 2) O sistema apresenta os gastos do utilizador/ valores das listas de compras num determinado período.
5	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade da pesquisa de produtos por categoria
	<b>Requisito</b>	RF005 – Visualizar produtos por categoria
	<b>Test Case</b>	1) O sistema apresenta um menu de pesquisa por categoria de produto.
6	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de localização do utilizador e respetivos supermercados à volta.
	<b>Requisito</b>	RF006 – Localização
	<b>Test Case</b>	1) O utilizador insere a sua localização; 2) O sistema apresenta supermercados na zona do utilizador.
7	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de alteração / atualização de definições do utilizador.
	<b>Requisito</b>	RF007- Gerir conta
	<b>Test Case</b>	1) O utilizador acede à sua área privada; 2) O utilizador altera/atualiza definições permitidas;
8	<b>Tipo de Teste</b>	Funcional

	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de criação de um novo produto.
	<b>Requisito</b>	RF008 – Registo de novos produtos
	<b>Test Case</b>	1) O sistema apresenta um menu de pesquisa; 2) Utilizador informa o produto que quer pesquisar; 2.1) Se produto existir -> aparece ao utilizador; 2.2) Se produto não existir -> não aparece ao utilizador. 3) Caso produto não exista, utilizador cria novo produto 3.1) Utilizador insere detalhes do produto-> nome, categoria, preço, supermercado, quantidade.
9	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de alteração de preço de um produto
	<b>Requisito</b>	RF009 – Edição de preço do produto
	<b>Test Case</b>	1) O utilizador pesquisa por um produto; 2) O utilizador edita o preço do produto.
10	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de reportar um erro num produto.
	<b>Requisito</b>	RF010 – Reportar erro de preço
	<b>Test Case</b>	1) O utilizador pesquisa por um produto; 2) O utilizador clica em reportar erro; 3) O utilizador indica qual o motivo do erro.
11	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de consultar o total do preço dos produtos adicionados à lista de compras.
	<b>Requisito</b>	RF010 – Consultar valor total na lista de Compras
	<b>Test Case</b>	1) O utilizador adiciona produtos à lista de compras; 2) O sistema devolve o total dos produtos adicionados à respetiva lista.
12	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de alerta de limite de gastos
	<b>Requisito</b>	RF011 – Limite de gastos
	<b>Test Case</b>	1) O utilizador define um limite a gastar nas suas compras; 2) O utilizador adiciona produtos à lista; 3) O sistema notifica quando o limite é atingido.
13	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de impressão da lista de compras do utilizador
	<b>Requisito</b>	RF012 – Imprimir lista de compras:
	<b>Test Case</b>	1) O utilizador adiciona produtos à lista; 2) O utilizador imprime a sua lista de compras.

14	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade alertas ao utilizador
	<b>Requisito</b>	RF013 – Notificações
	<b>Test Case</b>	1) O sistema notifica o utilizador consoante respetivos gastos, idas às compras.
15	<b>Tipo de Teste</b>	Funcional
	<b>Subtipo de Teste</b>	Requisitos
	<b>Objetivo</b>	Testar a funcionalidade de sair da aplicação
	<b>Requisito</b>	RF014 – Logout
	<b>Test Case</b>	1) O utilizador registado clica no botão de sair; 2) O sistema devolve a página inicial com sessão do utilizador encerrada.

Tabela 5. Testes Funcionais.

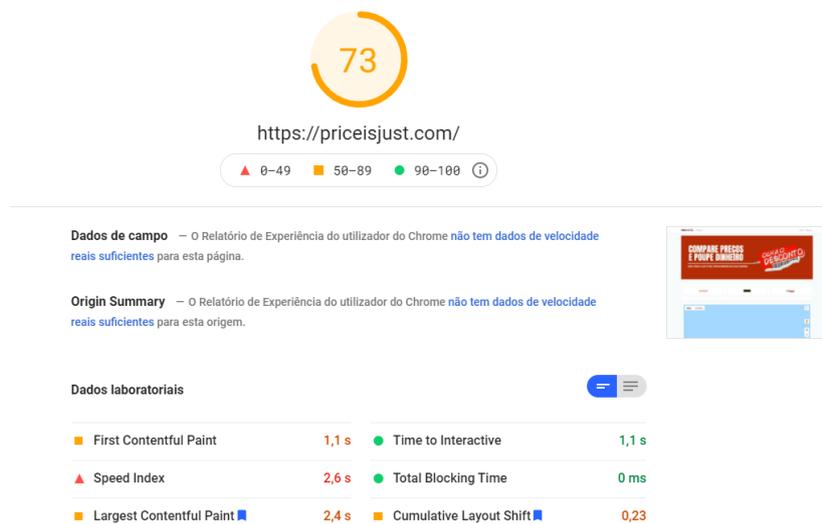
1	<b>Tipo de Teste</b>	Não-Funcional
	<b>Subtipo de Teste</b>	Tecnologia / Implementação
	<b>Objetivo</b>	Testar o Servidor e utilização correta da base de dados
	<b>Requisitos</b>	RFN01- Servidor disponível integralmente; RFN05 - A aplicação deve permitir uma utilização permanente, em qualquer hora do dia e dia da semana.
2	<b>Tipo de Teste</b>	Não-Funcional
	<b>Subtipo de Teste</b>	Usabilidade
	<b>Objetivo</b>	Testar usabilidade da aplicação
	<b>Requisito</b>	RFN02 -A aplicação deve permitir uma navegação simples, lógica e intuitiva com um menu que disponibilize as informações necessárias ao utilizador; RFN03 - A aplicação deve permitir uma utilização permanente, em qualquer hora do dia e dia da semana.
3	<b>Tipo de Teste</b>	Não-Funcional
	<b>Subtipo de Teste</b>	Usabilidade
	<b>Objetivo</b>	Testar se o sistema fica adaptado às diferentes multiplataformas
	<b>Requisito</b>	RFN09 – A aplicação deve funcionar em multiplataformas.
4	<b>Tipo de Teste</b>	Não-Funcional
	<b>Subtipo de Teste</b>	Segurança
	<b>Objetivo</b>	Testar segurança dos dados do utilizador
	<b>Requisito</b>	RFN07- A aplicação deve permitir que cada utilizador só tenha acesso aos seus dados e todas as palavras-passe devem estar cifradas.
5	<b>Tipo de Teste</b>	Não-Funcional
	<b>Subtipo de Teste</b>	Performance
	<b>Objetivo</b>	Testar velocidade da aplicação.
	<b>Requisito</b>	RFN06- A aplicação deve permitir que as suas operações sejam efetuadas com rapidez.

Tabela 6. Testes Não Funcionais.

- **Testes de Desempenho** – De forma a incluir testes de performance, medindo o tempo que o servidor demora a responder ao pedido do utilizador, incluindo também a velocidade da aplicação e experiência do utilizador, a página web foi analisada com a ferramenta gratuita do Google - Page Speed Insights<sup>5</sup>.

Relativamente à velocidade, foram dados os seguintes aspetos:

- 1- Reduzir o tempo de resposta do servidor inicial;
- 2- Eliminar recursos que bloqueiam o processamento;
- 3- Ativar compressão de texto;
- 4- Dimensionar adequadamente as imagens;
- 5- Reduzir JavaScript não utilizado;
- 6- Reduzir CSS;



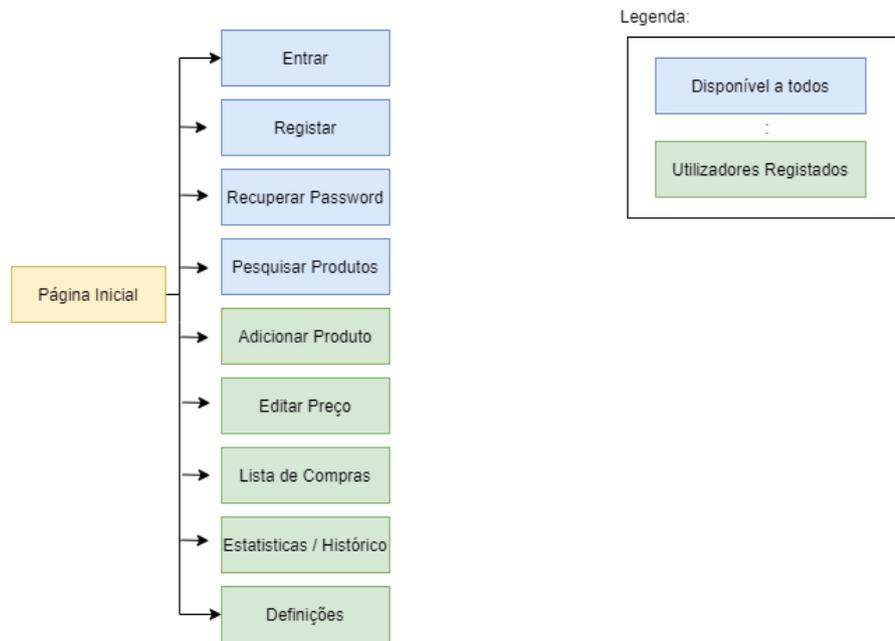
**Figura 4. Exemplo do teste utilizando a ferramenta Page Speed Insights.**

- **Teste de Usabilidade** – Sendo este teste focado na experiência do utilizador, na consistência do layout e acesso às funcionalidades, de forma a testar se a aplicação é ou não intuitiva. Para a realização destes testes, foram seleccionadas pessoas das que foram entrevistadas anteriormente e na qual disseram que estavam dispostas a utilizar a aplicação, de forma a conseguir resultados mais próximos da realidade. Os testes foram realizados tanto no website como no dispositivo móvel. Os objetivos e questões foram quanto ao grau de satisfação

<sup>5</sup> <https://developers.google.com/speed/pagespeed/insights/?hl=pt-PT>

do utilizador e quais as dificuldades que tiveram na utilização do sistema. As tarefas que desenvolveram encontram-se no [Apêndice B](#).

Ainda neste tipo de teste houve uma preocupação com as funcionalidades que são acedidas pelos tipos de utilizador:



**Figura 5. Mapa de acessibilidade consoante tipo de utilizador.**

Os utilizadores conseguiram realizar todas as tarefas apresentadas. Após realizados os testes de usabilidade, foi feito um breve questionário, que se encontra no [Apêndice D](#), para anotar o grau de satisfação, as dificuldades sentidas e possíveis sugestões para melhoria da aplicação. No [Apêndice E](#), encontram-se também as tabelas com os resultados obtidos.

Foram também realizados também:

- **Testes de Integração-** Para verificar se o sistema funciona corretamente, quando do momento de integração da plataforma online, foram efetuados vários testes, mais precisamente, a nível do domínio, do “host”, da integração dos protocolos necessários (SSL<sup>6</sup>/TLS<sup>7</sup>) e da associação da base de dados e do sistema de ficheiros corretamente.

---

<sup>6</sup> Secure Socket Layers

<sup>7</sup> Transport Layer Security

- **Teste de Segurança** – De forma a analisar se os dados são acedidos de forma segura, foram utilizados e testados em vários ambientes e browser, os protocolos de segurança SSL e TSL, que permitem visitar o website de forma segura, substituindo o URL<sup>8</sup> de HTTP<sup>9</sup> para um protocolo de transferência mais segura (HTTPS<sup>10</sup>).
- **Testes Unitários** – De forma a testar componentes isoladas, e verificar se os dados introduzidos têm o comportamento esperado, foram feitos vários testes, onde se incluíram utilizadores externos, nos vários módulos existentes na plataforma, realçando os de gestão de produtos (criar/editar/pesquisar) e as funcionalidades das listas de compras.

Não foram encontrados quaisquer erros a nível de programação. Todas as funcionalidades cumpriram os objetivos estabelecidos. Após uma análise das várias respostas dadas pelos utilizadores, foi possível concluir que a aplicação tem uma navegação intuitiva, que os utilizadores preferiram utilizar na aplicação móvel e estão dispostos a utilizar a mesma.

No próximo capítulo é descrita a solução desenvolvida.

---

<sup>8</sup> *Uniform Resource Locator*

<sup>9</sup> *Hyper Text Transfer Protocol*

<sup>10</sup> *Hyper Text Transfer Protocol Secure*

## 4 Solução Desenvolvida

Nos capítulos anteriores foram já identificados os problemas e frustrações dos utilizadores num processo de realização de compras, e como tal, neste capítulo é então apresentada como foi desenvolvida a solução para resolver os problemas em questão.

A arquitetura geral é constituída por duas camadas: “*Front-End*” e “*Back-End*”, sendo estas que descrevem o funcionamento lógico do software. Já o ambiente de desenvolvimento está assente no *PHP*<sup>11</sup> mais propriamente na framework *Laravel*<sup>12</sup> com estrutura MVC.<sup>13</sup>

O padrão de uma arquitetura MVC divide uma aplicação em três componentes principais que interagem uns com os outros: o “*Model*”, a “*View*” e o “*Controller*”, onde em suma, o “*Controller*” recebe todos os pedidos feitos pelo utilizador e de seguida trabalha com o “*Model*” para preparar os dados que são necessários para a “*View*”, já esta, define e gere o modo como os dados são apresentados ao utilizador.

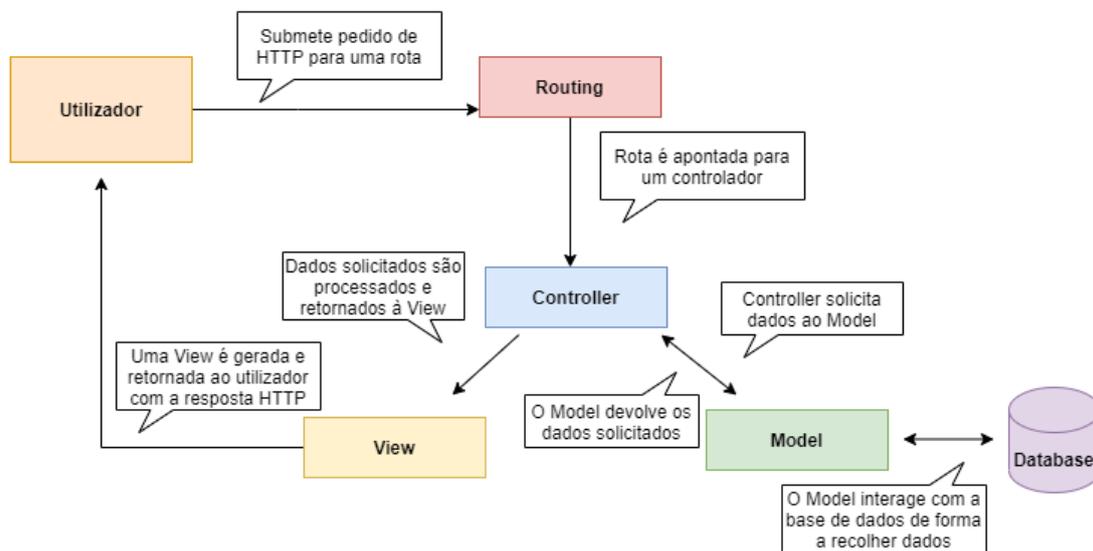


Figura 6. Arquitetura MVC implementada por Laravel

Escolher a tecnologia a utilizar é fundamental no processo de desenvolvimento, como tal nesta aplicação foi utilizada a linguagem *PHP*, pois é uma linguagem mais

<sup>11</sup> <https://www.php.net>

<sup>12</sup> <https://laravel.com>

<sup>13</sup> *Model-View-Controller* - [https://www.tutorialspoint.com/struts\\_2/basic\\_mvc\\_architecture.htm](https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm)

acessível e possui “*frameworks*” que auxiliam no seu desenvolvimento, tornando-o mais rápido com soluções flexíveis, promovendo práticas de desenvolvimento web modernas e com grande documentação e apoio comunitário.

Como já referido, a *Laravel* é então uma “*framework*” PHP utilizada para o desenvolvimento web, que utiliza a arquitetura MVC (“*Model-View-Controller*”). Tem como principal característica ajudar a desenvolver aplicações seguras, com código limpo e simples. Requer a instalação do gestor de dependências PHP (*Composer*<sup>14</sup>), bem como, para simular um servidor local, a instalação de um ambiente de desenvolvimento PHP (*Mamp*<sup>15</sup>, ou *Xampp*<sup>16</sup>) que irá criar um servidor local, tratar das conexões às bases de dados e habilitar a utilização da linguagem PHP. A *Laravel* disponibiliza ainda uma linha de comandos (*Artisan*<sup>17</sup>) com um conjunto de comandos úteis que permitem automatizar tarefas associadas ao desenvolvimento da aplicação.

Comandos Terminal	Descrição
<code>composer create-project laravel/laravel myappname</code>	Criar projeto Laravel.
<code>php artisan serve</code>	Comando que possibilita a ligação a um servidor remoto.
<code>php artisan make:controller</code>	Gerar um Controller com um determinado nome.
<code>php artisan make:model</code>	Gerar um Model com um determinado nome.
<code>php artisan make:migration</code>	Gerar uma classe que representa a tabela de dados que irá estar na base de dados após a migração
<code>php artisan make:auth</code>	Gerar controladores e vistas para autenticação e manutenção das contas dos utilizadores.
<code>php artisan migrate</code>	Migração de todas as classes que representam a tabela de dados.

**Tabela 7. Lista de comandos mais utilizados.**

<sup>14</sup> <https://getcomposer.org>

<sup>15</sup> <https://www.mamp.info/en/windows/>

<sup>16</sup> <https://www.apachefriends.org/index.html>

<sup>17</sup> <https://laravel.com/docs/8.x/artisan>

### **“Routes e Controllers”**

Um “Route”, é um link/url na aplicação. São utilizados para as várias páginas ou retorno de dados da aplicação. Não só indica o link, mas também a sua função, por norma, executando uma função específica no Controller.

```
Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])
```

**Figura 7. Exemplo de uma Route.**

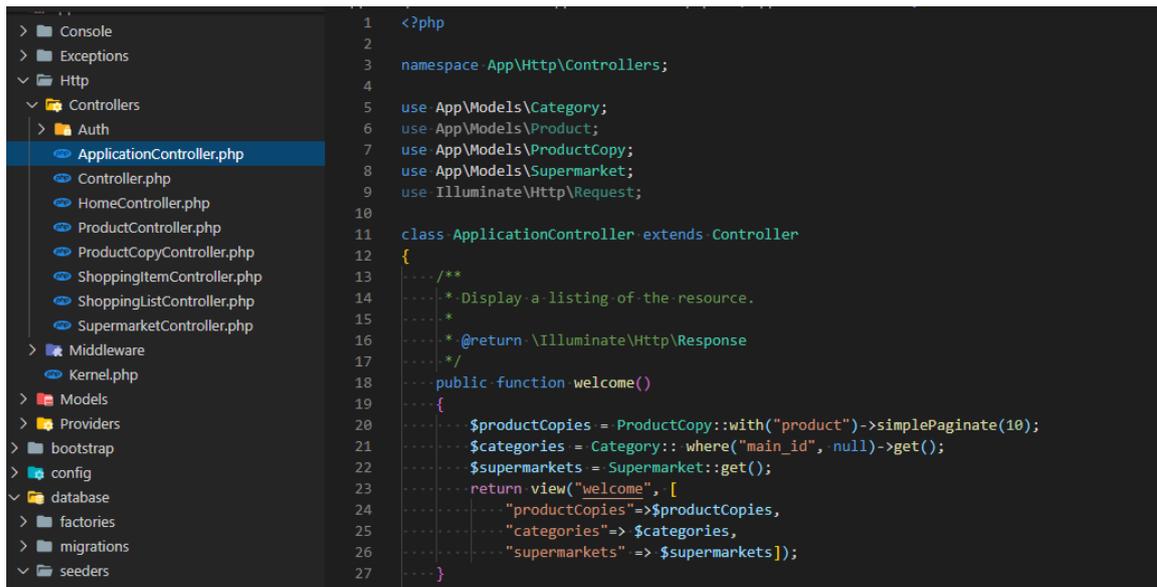
A aplicação verifica sempre se existe alguma regra definida, caso se verifique, é efetuado o mapeamento para o método do controlador correspondente, neste exemplo em cima referido é o “HomeController” e o método é o “index”.

Método HTTP	Caminho URL (Rout)	Ação (Método)	Nome da Rota
GET	/products	index	products.index
GET	/ products /create	create	products.create
POST	/ products	store	products.store
GET	/ products /{product}	show	products.show
GET	/ products /{product}/edit	edit	products.edit
PUT/PATCH	/ products /{product}	update	products.update
DELETE	products /{product}	destroy	products.destroy

**Tabela 8. Lista de Métodos disponibilizados e respetivas relações.**

Um “Controller” é então uma classe com o nome do objeto incluindo a palavra “Controller” no final, por exemplo, a tabela “Products”, com o model “Product” tem um “Controller” com o nome “ProductController”, onde nele, estão as funções para buscar a listagem de dados, mostrar um dado específico e apagar dados.

Após ser criado um “Controller” é necessário criar as respectivas “Views”.



```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Category;
6 use App\Models\Product;
7 use App\Models\ProductCopy;
8 use App\Models\Supermarket;
9 use Illuminate\Http\Request;
10
11 class ApplicationController extends Controller
12 {
13     /**
14      * Display a listing of the resource.
15      *
16      * @return \Illuminate\Http\Response
17      */
18     public function welcome()
19     {
20         $productCopies = ProductCopy::with("product")->simplePaginate(10);
21         $categories = Category::where("main_id", null)->get();
22         $supermarkets = Supermarket::get();
23         return view("welcome", [
24             "productCopies" => $productCopies,
25             "categories" => $categories,
26             "supermarkets" => $supermarkets]);
27     }
28 }

```

Figura 8. Exemplo de um Controller.



```

<div class="form-row">
  <div class="col-6 col-lg-3"><label for="quantity">Quantidade</label></div>
  <div class="col-6 col-lg-4"><label for="unity">Peso / Volume</label></div>
</div>
<div class="form-row">
  <div class="col-6 col-lg-3">
    <div class="form-group">
      <input type="number" name="quantity" class="form-control " value="1" min="1" required>
    </div>
  </div> <!-- col.// -->
  <div class="col-6 col-lg-4">
    <select name="unity" class="form-control" required>
      <option @if($productCopy->product->unity == "un") selected @endif value="un"> Unidade (un) </option>
      <option @if($productCopy->product->unity == "kg") selected @endif value="kg"> Kilograma (kg) </option>
      <option @if($productCopy->product->unity == "g") selected @endif value="g"> Grama (g) </option>
      <option @if($productCopy->product->unity == "l") selected @endif value="l"> Litro (l) </option>
      <option @if($productCopy->product->unity == "cl") selected @endif value="cl"> Centilitro (cl) </option>
    </select>
  </div> <!-- col.// -->
</div> <!-- col.// -->
<div class="col">
  <button type="submit" class="btn btn-primary w-100"> Adicionar à Lista <i class="fas fa-shopping-cart"></i></a>
</div> <!-- col.// -->
</div> <!-- row.// -->
</form>

```

Figura 9. Exemplo de uma View.

## “Migrations”

As “Migrations” são a designação da estrutura das tabelas, isto é, em vez de se escrever diretamente na base de dados, a *Laravel* usa estas, que possibilitam uma forma fácil de indicar os campos da tabela e algumas funções adicionais que ajudam no desenvolvimento. Estas permitem modificar o esquema da base de dados, mantendo todas as cópias da aplicação atualizadas.

```
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateProductsTable extends Migration
8 {
9     /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('products', function (Blueprint $table) {
17             $table->id();
18             $table->unsignedBigInteger("category_id");
19             $table->string("image_url")->nullable();
20             $table->enum("unity", ["un", "kg", "g", "l", "dl", "cl"]);
21             $table->timestamps();
22             $table->string("title")->nullable();
23             $table->string("description")->nullable();
24             $table->double("quantity");
25
26             //Adicionado pelo user
27             $table->boolean("is_accepted")->default(false); // Administrador tem de aprovar
28             $table->unsignedBigInteger("user_id")->nullable();
29         });
30     }
31 }
```

Figura 10. Exemplo de Migration

### “Models”

Os “Models” são o rosto das “Migrations”. Estes indicam que campos são modificados/criados. Os casos em que podem ser modificados, indicam as relações entre outras migrações e até mesmo, funções que sejam próprias das tabelas, ou seja, quando o utilizador não tem uma imagem, existe uma função que traz uma imagem indicada pelo programa.

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    use HasFactory;

    protected $fillable = [
        'title',
        'description',
        'image_url',
        'is_accepted',
        'user_id',
        'category_id',
        'unity',
        'quantity',
    ];

    public function user(){
        return $this->belongsTo(User::class); // Este produto pertence a uma pessoa
    }

    public function category(){

```

Figura 11. Exemplo de Model

## “Factory e Seeds”

Uma “Factory”, é uma classe que permite criar dados falsos que ajudam a testar e desenvolver o sistema sem ter que preocupar com a criação de dados de teste. Para adicionar estes dados à base de dados, é necessário chamar a “Factory” através de uma “Seed”.

Uma “Seed” é uma classe que permite instanciar dados predefinidos de tabelas, que serão sempre os mesmos, por exemplo as categorias dos produtos. Estes dados podem ser indicados manualmente ou utilizando “Factories”.

```

database > factories > UserFactory.php > ...
 9 class UserFactory extends Factory
10 {
11     /**
12     * The name of the factory's corresponding model.
13     *
14     * @var string
15     */
16     protected $model = User::class;
17
18     /**
19     * Define the model's default state.
20     *
21     * @return array
22     */
23     public function definition()
24     {
25         return [
26             'first_name' => $this->faker->name,
27             'last_name' => $this->faker->name,
28             'email' => $this->faker->unique()->safeEmail,
29             'email_verified_at' => now(),
30             'password' => '$2y$10$92IXUNpkj0r0Q5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password
31             'remember_token' => Str::random(10),
32         ];
33     }
34

```

Figura 12. Exemplo de uma Factory.

```

database > seeders > CategorySeeder.php > CategorySeeder > run
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6
7 class CategorySeeder extends Seeder
8 {
9     /**
10     * Run the database seeds.
11     *
12     * @return void
13     */
14     public function run()
15     {
16         //*****
17         //** Mercadoria Doce (1)
18         //*****
19         \App\Models\Category::create(["id"=>"1"]); //Mercadoria Doce
20         \App\Models\Category::create(["id"=>"2", "main_category_id"=>1]); // Sub: Tartes e Bolos
21         \App\Models\Category::create(["id"=>"3", "main_category_id"=>1]); // Sub: Bolachas
22         \App\Models\Category::create(["id"=>"4", "main_category_id"=>1]);
23         \App\Models\Category::create(["id"=>"5", "main_category_id"=>1]);
24         \App\Models\Category::create(["id"=>"6", "main_category_id"=>1]);
25         \App\Models\Category::create(["id"=>"7", "main_category_id"=>1]);
26         \App\Models\Category::create(["id"=>"8", "main_category_id"=>1]);
27         \App\Models\Category::create(["id"=>"9", "main_category_id"=>1]);

```

Figura 13. Exemplo de uma Seeder.

## Base de Dados

A ligação à base de dados é parametrizada em dois ficheiros de configuração existentes dentro da própria aplicação: “*database.php*” e o “*.env*”. Sendo que no primeiro é configurado o driver de configuração que pode ser *MySQL*<sup>18</sup>, *SQLite*<sup>19</sup> entre outras. Já no segundo ficheiro é onde estão guardadas as credenciais de acesso, permitindo assim a conexão de forma correta à base de dados.

```
config > database.php
33
34
35
36 ..... 'connections' => [
37
38 ..... 'sqlite' => [
39 ..... 'driver' => 'sqlite',
40 ..... 'url' => env('DATABASE_URL'),
41 ..... 'database' => env('DB_DATABASE', database_path('database.sqlite')),
42 ..... 'prefix' => '',
43 ..... 'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),
44 ..... ],
45
46 ..... 'mysql' => [
47 ..... 'driver' => 'mysql',
48 ..... 'url' => env('DATABASE_URL'),
49 ..... 'host' => env('DB_HOST', '127.0.0.1'),
50 ..... 'port' => env('DB_PORT', '3306'),
51 ..... 'database' => env('DB_DATABASE', 'forge'),
52 ..... 'username' => env('DB_USERNAME', 'forge'),
53 ..... 'password' => env('DB_PASSWORD', ''),
54 ..... 'unix_socket' => env('DB_SOCKET', ''),
55 ..... 'charset' => 'utf8mb4',
56 ..... 'collation' => 'utf8mb4_unicode_ci',
57 ..... 'prefix' => '',
58 ..... 'prefix_indexes' => true,
59 ..... 'strict' => true,
60 ..... 'engine' => null,
61 ..... 'options' => extension_loaded('pdo_mysql') ? array_filter([
62 ..... PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
63 ..... ]) : [],
64 ..... ],
65
```

**Figura 14. Configuração da base de dados.**

```

1 APP_NAME=PriceIsJust
2 APP_ENV=local
3 APP_KEY=base64:b1II2dcsELLNoCRR7fYoVkdGZXNZ8m7fFsJLbQTNMs=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=sqlite
11 |
12 BROADCAST_DRIVER=log
13 CACHE_DRIVER=file
14 QUEUE_CONNECTION=sync
15 SESSION_DRIVER=file
```

**Figura 15. Ficheiro .env de configuração.**

---

<sup>18</sup> <https://www.mysql.com>

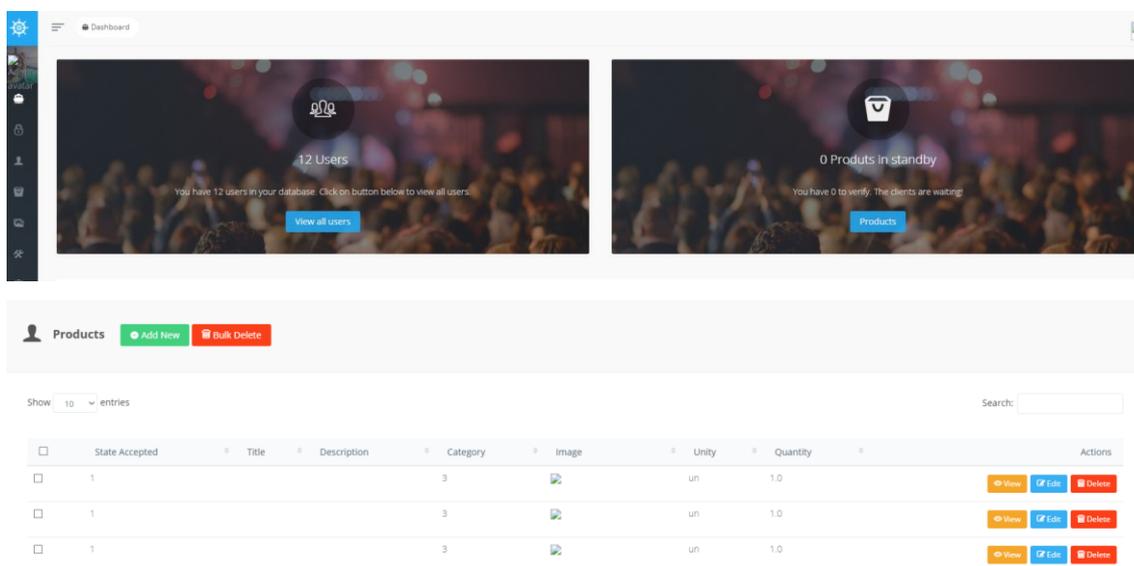
<sup>19</sup> <https://www.sqlite.org/index.html>

Após o desenho de arquitetura e desenvolvimento dos modelos que implementam a lógica e respetiva configuração, é necessário migrar toda a informação através do comando “*php artisan migrate*”<sup>20</sup>. Ao correr esta linha de comando, todas as migrações do modelo de dados são executadas implementando a lógica na base de dados.

```
Migrating: 2021_03_30_213407_create_categories_table
Migrated:  2021_03_30_213407_create_categories_table (5.13ms)
Migrating: 2021_03_30_213532_create_supermarkets_table
Migrated:  2021_03_30_213532_create_supermarkets_table (5.97ms)
Migrating: 2021_03_30_213925_create_shopping_lists_table
Migrated:  2021_03_30_213925_create_shopping_lists_table (5.02ms)
Migrating: 2021_04_02_201958_create_shopping_items_table
```

Figura 16. - Execução do comando para migrar o modelo de dados.

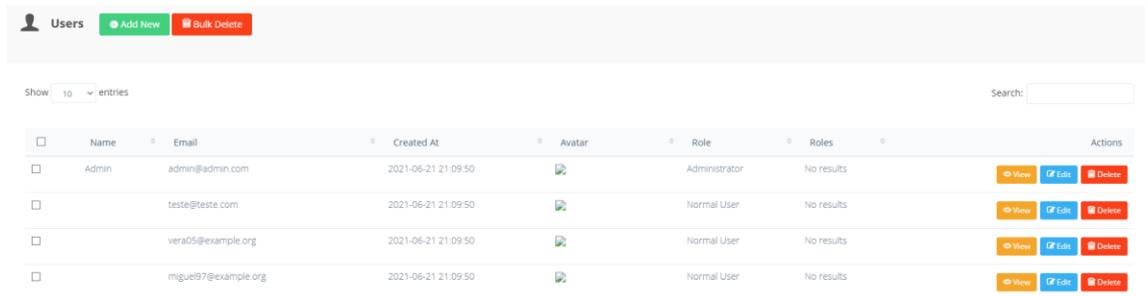
Como já mencionado anteriormente, para ajudar na parte administrativa, foi utilizado o “Voyager”, um plugin gratuito do Laravel, que permite fazer toda a gestão de qualquer tabela existente no sistema, utilizando um módulo do Voyager chamado BREAD. Este, permite adicionar, editar, gerir e remover áreas de gestão das tabelas existentes, na qual foi utilizado para a gestão de produtos, utilizadores e casos reportados de produtos.



The screenshot displays the Voyager admin interface. At the top, there are two summary cards: '12 Users' with a 'View all users' button and '0 Products in standby' with a 'Products' button. Below these is a 'Products' section with 'Add New' and 'Bulk Delete' buttons. A table lists products with columns for State Accepted, Title, Description, Category, Image, Unity, and Quantity. Each row includes 'View', 'Edit', and 'Delete' action buttons.

State Accepted	Title	Description	Category	Image	Unity	Quantity	Actions
1			3		un	1.0	View Edit Delete
1			3		un	1.0	View Edit Delete
1			3		un	1.0	View Edit Delete

<sup>20</sup> <https://laravel.com/docs/8.x/migrations>



<input type="checkbox"/>	Name	Email	Created At	Avatar	Role	Roles	Actions
<input type="checkbox"/>	Admin	admin@admin.com	2021-06-21 21:09:50		Administrator	No results	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	teste	teste@teste.com	2021-06-21 21:09:50		Normal User	No results	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	vera05	vera05@example.org	2021-06-21 21:09:50		Normal User	No results	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	miguel97	miguel97@example.org	2021-06-21 21:09:50		Normal User	No results	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

**Figura 17. Exemplo de Funcionalidades do Voyager.**

Para além das tecnologias acima mencionadas, foram também utilizadas outras tecnologias:

- **HTML**<sup>21</sup> – Linguagem de marcação e que serve para criar e representar visualmente as páginas *web*.
- **CSS**<sup>22</sup> - Mecanismo para adicionar estilo (cores, fontes, espaçamento, etc.) a um documento *web*.
- **JavaScript**<sup>23</sup> - Linguagem de programação interpretada, que é utilizada do lado do cliente e que permite ao utilizador interagir com a página *web*. Juntamente com o HTML e o CSS é uma das três tecnologias essenciais para a produção de conteúdo *web*.
- **Bootstrap**<sup>24</sup>- Framework que permite criar interfaces *web* com CSS e *JavaScript*, cuja particularidade é adaptar a interface do *site* para o tamanho do dispositivo que é exibido (“*Web-Responsive*”).
- **PhpMyAdmin**<sup>25</sup> - Permite facilmente gerir bases de dados através de uma interface gráfica.
- **Filezilla**<sup>26</sup> – Permite facilmente transferir dados do computador para o servidor.

A aplicação “*Price is Just*” está disponível em multiplataformas: *Web*, *iOS App* e *Android App*. Os seus protótipos não funcionais encontram-se no [Apêndice C](#).

---

<sup>21</sup> <https://www.w3schools.com/html/>

<sup>22</sup> <https://www.w3schools.com/css/>

<sup>23</sup> <https://www.w3schools.com/js>

<sup>24</sup> <https://getbootstrap.com>

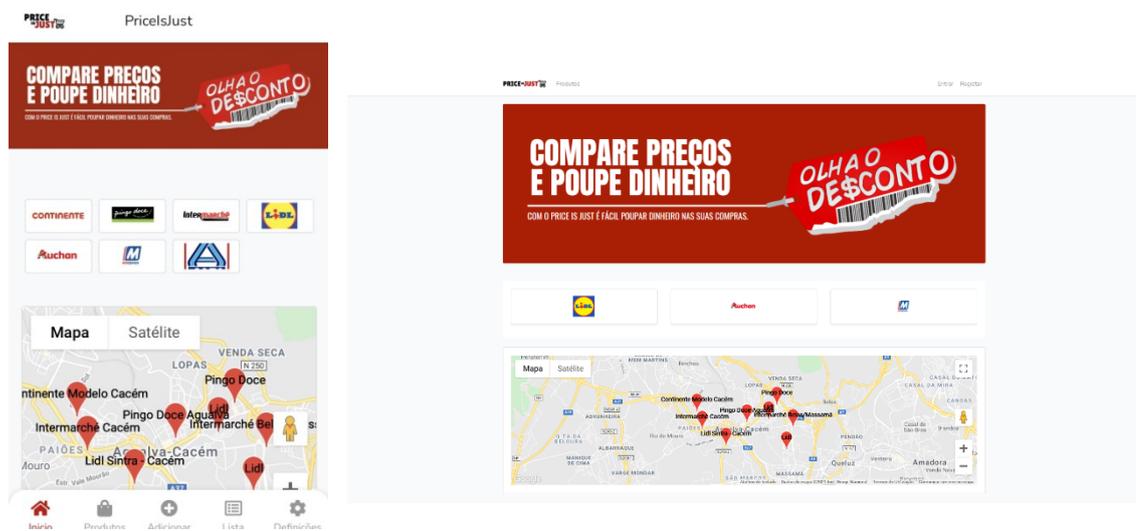
<sup>25</sup> <https://www.phpmyadmin.net>

<sup>26</sup> <https://filezilla-project.org>



**Figura 18. Representação genérica de Aplicação Multiplataformas.**

Para utilizar os serviços que permitem a disponibilização dos mapas do *Google*, foi necessário criar uma conta no *site* do “*Google Developers*”<sup>27</sup>. Sendo que ao efetuar o registo de conta, foi atribuída uma chave que fornece o acesso a estes serviços. Através desta chave, foi disponibilizada uma vasta gama de funcionalidades como a criação do mapa, permitindo gerar a localização do utilizador e supermercados.



**Figura 19. Funcionalidade do Google Maps em Mobile e Web.**

Ao iniciar a aplicação, o mapa é gerado e mostrado vazio ao utilizador, onde de seguida, é solicitado a permissão do acesso à sua localização. No decorrer desta ação, é efetuado um pedido ao servidor (*priceisjust*), pela lista de todos os supermercados existentes na base de dados, que servirá para filtrar através do seu nome, todos os estabelecimentos que foram recebidos do *Google Maps*. Desta forma, é possível obter uma lista abrangente da localização dos supermercados existentes no mundo inteiro, sem a necessidade de as gerir internamente, isto é, criar as tabelas das localizações dos

<sup>27</sup> <https://developers.google.com>

supermercados e gerir as mesmas. Foi criada uma funcionalidade onde ao clicar no ícone de localização do supermercado, é feito zoom no mesmo. Ao clicar novamente no mesmo supermercado, a plataforma remete-nos para a página de pesquisa de produtos desse supermercado.



```
function init() {
  createMap(0, 0);

  var xhttp = new XMLHttpRequest();
  var metas = document.getElementsByTagName('meta');

  xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) {
      supermarkets = JSON.parse(this.responseText)
    }
    localizarUsuario();
  };
  xhttp.open("POST", "/supermarkets", true);
  for (i = 0; i < metas.length; i++) {
    if (metas[i].getAttribute("name") == "csrf-token") {
      xhttp.setRequestHeader("X-CSRF-Token",
        metas[i].getAttribute("content"));
    }
  }
  xhttp.send();
}

function inSupermarkets(place) {
  for (var i = 0; i < supermarkets.length; i++) {
    if (place.name.includes(supermarkets[i].name)) {
      return supermarkets[i].link;
    }
  }
  return false;
}
```

Figura 20. Implementação do filtro de supermercados.

Relativamente às tecnologias referidas nas entregas anteriores, onde seria usado Gatsby<sup>28</sup>, React<sup>29</sup>, GraphQL<sup>30</sup> e Electron<sup>31</sup>, após pesquisas feitas e com a ajuda da unidade curricular de Computação Móvel percebi que podia escolher outras tecnologias mais acessíveis e eficazes. Desenvolver em Angular<sup>32</sup>, tal como dado na unidade curricular de Programação Web, também foi uma hipótese, no entanto tornava-se mais pesado do que as *“frameworks”* utilizadas.

A razão pela escolha destas tecnologias deve-se ao facto de serem ferramentas sempre em constante evolução, o que também se tornou um desafio pessoal dominá-las, tal como me irão permitir ter outros conhecimentos mais aprofundados para enfrentar o mercado de trabalho nesta área que é o meu objetivo.

A mudança de nome de *“The Price is Right”* para *“Price is Just”* deveu-se ao facto de já não existir domínio<sup>33</sup> disponível para a primeira escolha.

No capítulo seguinte é feito um estudo de mercado, de forma a analisar os potenciais concorrentes.

---

<sup>28</sup> <https://www.gatsbyjs.com>

<sup>29</sup> <https://reactjs.org>

<sup>30</sup> <https://graphql.org>

<sup>31</sup> <https://www.electronjs.org>

<sup>32</sup> <https://angular.io>

<sup>33</sup> <https://www.2mhost.com>

## 5 Benchmarking

Após identificação do problema e apresentada a solução desenvolvida, neste capítulo pretendeu-se estudar o mercado atual, os possíveis concorrentes, de forma a analisar e estruturar uma estratégia para inserção no mercado e conseqüente sucesso.

Atualmente estamos perante uma constante e iminente evolução tecnológica acompanhada de um crescimento exponencial das potencialidades das mesmas, onde assistimos a uma era cada vez mais digital e voltada para o consumidor, sendo que os avanços tecnológicos servem não só na visão das empresas, mas também, dos consumidores, que são cada vez mais exigentes e sensíveis em determinados aspetos, e sempre que estiver à sua disposição uma tecnologia que lhe acelere um processo ou evite outros, este não terá grandes dúvidas na sua utilização.

No mercado atual, existem algumas aplicações que comparam preços, algumas com capacidade de oferecer maior diversidade de produtos do que outras, interfaces alternativas, apresentações diferentes para os produtos comparados, no entanto o critério mais importante para o consumidor na escolha de um produto é sempre o preço. Nestas aplicações, a utilização das mesmas é semelhante, sendo que o utilizador escolhe um produto, pesquisando numa caixa de texto ou numa seleção de categorias até atingir o mesmo.

Com base em diversas pesquisas feitas na internet, entre elas artigos que avaliam a popularidade destas aplicações, podemos considerar para o presente estudo, como potenciais concorrentes, as seguintes plataformas: PriceGrabber<sup>34</sup>, Buscapé<sup>35</sup>, Kuantokusta<sup>36</sup>, Kuantokusta Supermercados<sup>37</sup> e Save Cook<sup>38</sup>.

O PriceGrabber é um comparador de preços que existe na América do Norte e Reino Unido e que compara milhares de produtos e serviços de diferentes categorias. O site oferece aos clientes, a possibilidade de comparar preços e características dos produtos em diversas lojas online, permitindo assim aos utilizadores, encontrarem o artigo desejado. Inclui sistemas de classificações, opiniões de comerciantes, comentários

---

<sup>34</sup> <http://www.pricegrabber.com>

<sup>35</sup> <https://www.buscape.com.br>

<sup>36</sup> <https://www.kuantokusta.pt>

<sup>37</sup> <https://www.kuantokusta-supermercados.pt>

<sup>38</sup> <https://save-cook.com>

detalhados sobre o produto, comparações de produtos lado a lado, e notificações de e-mail com os melhores preços e disponibilidade na Internet.

O BuscaPé é um site brasileiro, líder na decisão de compra, na América Latina, que permite também a pesquisa de produtos e posterior comparação de preços.

O Kuantokusta é um site líder em Portugal de comparador de preços. Este ajuda os consumidores a comparar os produtos expostos pelos comerciantes, para assim encontrarem as melhores ofertas de produtos e de serviços disponíveis. O resultado de uma pesquisa, apresenta uma listagem com produtos do mesmo género, ordenados pelo melhor preço, com o logotipo e o link da loja que vende o produto.

Desde 2016 que o Kuantokusta lançou uma vertente para supermercados, o Kuantokusta Supermercados, onde se destacam as promoções do dia, permitindo acompanhar, ao momento, as campanhas de todos os supermercados. Para cada produto, é ainda disponibilizada informação sobre o preço noutras lojas, de forma a informar o consumidor se a promoção é de real. O Kuantokusta Supermercados compara preços de 6 lojas em Portugal: Continente, Intermarché, Jumbo, Supercor (El Corte Inglés), Froiz e Amazonite. Para além disso, este site está disponível para comparar e poupar de forma gratuita, onde o consumidor passa a ter vários supermercados numa única plataforma e informação atualizada ao minuto sobre preços e promoções. Existe também a opção de criar um carrinho de compras personalizado, existindo mais de 55 mil produtos disponíveis. Ao longo desta seleção, o sistema vai identificando os preços dos produtos escolhidos noutros supermercados, convidando o utilizador a trocar de loja caso exista uma situação de poupança. No final, pode ver-se o preço total em cada supermercado, mostrando qual é o mais barato.

A SaveCook é uma plataforma que ajuda os utilizadores a poupar nos supermercados e a organizar as suas refeições, com a disponibilização de receitas. É possível encontrar inúmeras receitas, com o respetivo grau de dificuldade, tempo de preparação e informação nutricional, sendo que ao escolher a receita pretendida é automaticamente gerada uma “Lista de Compras” com a indicação do preço dos

produtos em cada um dos principais supermercados. Deste modo o consumidor pode não só fazer a escolha mais barata, como ainda otimiza as suas idas ao supermercado.

Análise comparativa da concorrência						
Compara preços dos mesmos produtos em diferentes lojas?	✓	✓	✓	✓	✓	✓
Compara produtos de supermercado?	X	✓	✓	✓	✓	✓
Possibilidade de criar lista de compras?	X	X	X	✓	✓	✓
Guardar lista de compras?	X	X	X	✓	✓	✓
Imprimir lista de compras?	X	X	X	✓	X	✓
Utilizador pode registar novos produtos/preços?	X	X	X	X	X	✓
Inclui sistemas de classificações/ comentários?	✓	X	✓	X	X	✓
Utilizador tem acesso a estatísticas de consumo?	X	X	X	X	X	✓
Acesso a localização?	X	X	X	✓	✓	✓
Estatística de preço consoante distância à loja?	X	X	X	X	X	✓
Scan de código de barras com a câmara?	X	X	X	X	X	✓
Acesso em multiplataformas?	X	X	X	X	✓	✓
Disponível em Portugal?	X	X	✓	✓	✓	✓

**Tabela 9. Análise do Mercado - Comparação Inicial.**

Análise comparativa da concorrência						
Compara preços dos mesmos produtos em diferentes lojas?	✓	✓	✓	✓	✓	✓
Compara produtos de supermercados?	X	✓	✓	✓	✓	✓
Possibilidade de criar lista de compras?	X	X	X	✓	✓	✓
Guardar lista de compras?	X	X	X	✓	✓	✓
Imprimir lista de compras?	X	X	X	✓	X	✓
Utilizador pode registar novos produtos/preços?	X	X	X	X	X	✓
Utilizador pode reportar erros nos preços?	X	X	X	X	X	✓
Inclui sistema de classificações/ comentários?	✓	X	✓	X	X	
Utilizador tem acesso a estatísticas de consumo?	X	X	X	X	X	✓
Utilizador estabelece limite de gastos?	X	X	X	X	X	✓
Acesso à localização?	X	X	X	✓	✓	✓
Estatística de preço consoante a distância à loja?	X	X	X	X	X	
Utilizador recebe alertas / notificações?	X	X	X	X	X	
Scan de código de barras com a câmara?	X	X	X	X	X	
Acesso em multiplataformas?	X	X	X	X	X	✓
Disponível em Portugal?	X	X	✓	✓	✓	✓

**Tabela 10. Análise do Mercado - Comparação Final.**

Perante esta análise de mercado, pode-se concluir que o único concorrente direto é o KuntoKusta Supermercados, no entanto, este apenas está disponível em website e nem sempre está a funcionar corretamente, talvez devido ao facto de que, os preços dos produtos são geridos pela própria plataforma, fazendo com que estes estejam desatualizados ou inexistentes, como é possível verificar na figura abaixo (Figura.20). Sendo um dos possíveis pontos que levam a favorecer a “Price is Just” pois é uma aplicação presente em multiplataformas e com o servidor disponível integralmente.



Figura 21. Website KuntoKusta Supermercados sem resultados.

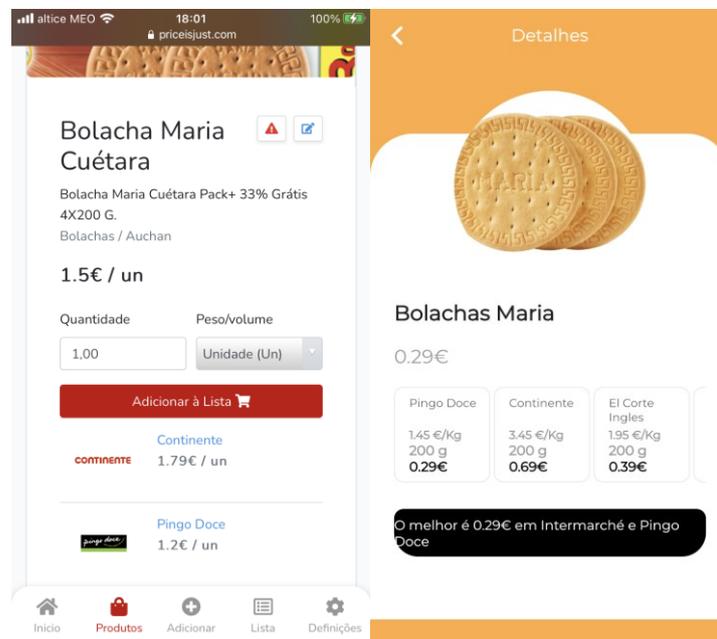


Figura 22. Comparação de produto: Price is Just vs Save Cook

No capítulo seguinte é demonstrado o calendário e respetivo método de trabalho utilizado no desenvolvimento do projeto.

## 6 Método e Planeamento

Nos capítulos anteriores foram descritos todos os procedimentos para a realização deste projeto, sendo que apenas falta apresentar o calendário e respetivo o planeamento.

O desenvolvimento deste Trabalho Final de Curso teve por base uma metodologia Agile<sup>39</sup>, em que se valoriza a permissão de efetuar mudanças tardias no projeto sem grandes problemas, software funcional e simplicidade.

Esta metodologia está dividida em sprints, sendo que a cada sprint existe um conjunto de atividades que acontecem de forma consecutiva, isto é, uma após a outra.

O plano de desenvolvimento foi o seguinte:

- **Sprint 1** - Identificar e compreender a construção de uma aplicação utilizando PHP, Laravel, bem como a sua arquitetura.
- **Sprint 2** - Desenvolver diagrama arquitetural da aplicação / Protótipo.
- **Sprint 3** - Desenvolver API do servidor.
- **Sprint 4** - Desenvolver registo, login, logout e homepage.
- **Sprint 5** - Desenvolver funcionalidade principal: CRUD dos produtos.
- **Sprint 6** - Desenvolver funcionalidade que permite aos utilizadores saber o supermercado mais barato, dentro de um raio.
- **Sprint 7** - Testes / Reavaliação e refinamento do projeto.

Nos cronogramas seguintes podemos fazer uma análise comparativa do que foi inicialmente estabelecido e do que foi realmente feito, onde podemos observar que todo o planeamento, todos os prazos e objetivos estipulados foram cumpridos com sucesso nas respetivas datas indicadas.

---

<sup>39</sup> <https://www.guru99.com/agile-scrum-extreme-testing.html>



## 7 Resultados

No capítulo anterior foi apresentado todo o planeamento e método de trabalho, sendo que neste capítulo são então apresentados os resultados finais.

Esta plataforma foi desenvolvida de forma responsiva, otimizada e visualmente agradável, como é possível verificar pelos testes efetuados e resultados obtidos por terceiros, com uma taxa de apreciação de 100% conforme se pode comprovar no [Apêndice E](#).

Foi desenvolvido e testado com sucesso, toda a gestão de produtos no qual foi dado ênfase à criação de novos produtos não existentes na plataforma, a inclusão dos mesmos em diferentes supermercados e a atualização de preços. Estes casos foram testados internamente e por terceiros, como já referido anteriormente.

Toda a componente foi desenvolvida recorrendo ao Laravel, que permitiu um desenvolvimento mais rápido e eficaz, garantindo segurança para o utilizador e para o sistema. No [Capítulo 4](#) foi apresentada a solução desenvolvida e no [Apêndice F](#) seguem-se os resultados finais tanto para Web como para Mobile.

De acordo com o [Capítulo 2](#), podemos constatar o cumprimento da maioria dos requisitos, sendo que os que não foram concretizados irão ser desenvolvidos no futuro. Os resultados dos testes em conjunto com a avaliação dos utilizadores que participaram nos mesmos, foram positivos, garantindo o funcionamento correto e exigido pelos mesmos requisitos.

No capítulo seguinte é feita a conclusão.

## 8 Conclusão e Trabalhos Futuros

Neste capítulo é então feita a conclusão e análise de trabalho futuro.

No início do desenvolvimento deste projeto, não se tem uma ideia clara do que se quer, nem uma noção do tempo e daquilo que se consegue fazer. A decisão de realizar esta aplicação foi sem dúvida a pensar na minha carreira profissional, sendo esta uma área que pretendo seguir. Com a realização deste projeto adquiri imensas competências, conhecimento e experiência que não tinha em diversas tecnologias.

No entanto, no decorrer dos semestres, fui confrontada com algumas dificuldades, a maior, foi logo no início, na escolha tecnologias a utilizar, onde perdi mais tempo para estudar a documentação das tecnologias, pesquisas online e a assistir a vídeos no Youtube, que me ajudaram bastante na realização deste. Outra dificuldade a referir, foi o tempo disponível, uma vez que me encontrava a trabalhar e estando a desenvolver este trabalho sozinha, tornou-se difícil a conciliação e cumprimento dos prazos e metas a atingir, no entanto, consegui cumprir sempre com todos os prazos e objetivos.

Com as dificuldades apresentadas, acabei também por adquirir um vasto conhecimento, desenvolvi a capacidade de resolver problemas, tornando-me mais autónoma e independente.

Como mencionado em cima, no início não conseguimos ter a noção do tempo, como tal, existiram funcionalidades propostas por mim, que não foram implementadas, sendo uma das perspetivas futuras continuar a desenvolver a aplicação, realizar melhorias no sistema, implementar o que faltou e novos requisitos e como o projeto foi feito de raiz por mim, poderei, eventualmente, utilizá-lo para fins comerciais.

No ponto de vista pessoal, depois desta experiência produtiva, e dos conhecimentos adquiridos ao longo da licenciatura, quero continuar a aprofundar os meus conhecimentos em tecnologias do desenvolvimento de *software*, procurar mais desafios e continuar a crescer do ponto de vista profissional.

## Bibliografia

- [1] I. PriceGrabber.com, “Pricegrabber,” 2020, <http://www.pricegrabber.com>
- [2] B. Company, “Buscapé,” 2020, <http://www.buscape.pt>
- [3] K. Kusta, “Kunto kusta,” 2020, <http://www.kuantokusta.pt>
- [4] K. Kusta, “Kunto kusta Supermercados,” 2020, <https://www.kuantokusta-supermercados.pt>
- [5] S. Cook, “Save Cook,” 2020, [https://save-cook.com/index\\_pt.html](https://save-cook.com/index_pt.html)
- [6] Artisan Console – “Laravel - The PHP Framework For Web Artisans”, 2021, <https://laravel.com/docs/8.x/artisan>
- [7] Basic MVC Architecture – Tutorialspoint, 2021, [https://www.tutorialspoint.com/struts\\_2/basic\\_mvc\\_architecture.htm](https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm)
- [8] “Google Developers” Google Developers, 2021, <https://developers.google.com/?hl=pt>
- [9] Composer, 2021, <https://getcomposer.org/>
- [10] About the XAMPP project, 2021, <https://www.apachefriends.org/about.html>
- [11] contributors, phpMyAdmin. “PhpMyAdmin”. PhpMyAdmin, 2021, <https://www.phpmyadmin.net/>
- [12] “Postman | The Collaboration Platform for API Development”. Postman, 2021, <https://www.postman.com/>
- [13] contributors, Mark Otto, Jacob Thornton, and Bootstrap. Bootstrap, 2021, <https://getbootstrap.com>
- [14] Laravel PHP Framework Tutorial - Full Course for Beginners. www.youtube.com, 2021, <https://www.youtube.com/watch?v=ImtZ5yENzgE>
- [15] Laravel Admin Package – Voyager, [https://www.youtube.com/watch?v=\\_\\_gp4E7tC-Vs](https://www.youtube.com/watch?v=__gp4E7tC-Vs)
- [16] Google Maps JavaScript API Tutorial, <https://www.youtube.com/watch?v=Zxf1mnP5zwc&t=1427s>
- [17] Laravel E-Commerce – Admin Backend w/ Laravel Voyager – Part 10, <https://www.youtube.com/watch?v=gN3BCi0emVA>
- [18] *MySQL: MySQL Workbench Manual : 9.3.1 Creating a Model*  
<https://dev.mysql.com/doc/workbench/en/wb-getting-started-tutorial-creating-a-model.html>
- [19] XAMPP Installers and Downloads for Apache Friends, 2021, <https://www.apachefriends.org/index.html>
- [20] Bootstrap 3 Tutorial, 2021, <https://www.w3schools.com/bootstrap/>

[21]WebPageTest - Website Performance and Optimization Test, 2021,

<https://www.webpagetest.org/>

[22]PageSpeed Insights, 2021,

<https://developers.google.com/speed/pagespeed/insights/?hl=pt-PT>

[23] SWOT Analysis: – How to Develop a Strategy For Success,2020,

[http://www.mindtools.com/pages/article/newTMC\\_05.htm](http://www.mindtools.com/pages/article/newTMC_05.htm)

[24] \$2.75/mo Cheap Hosting with Free SSL certificate | 2MHost.com, 2021,

<https://www.2mhost.com>

[25] Laravel voyager, 2021, <https://voyager.devdojo.com>

[26] FileZilla - The free FTP solution, 2021, <https://filezilla-project.org>

## Apêndice A – Questionário e Respetivas Respostas

### Questionário:

- a. Qual é a tua maior frustração em relação a ir às compras?
- b. Imagina que tinhas uma aplicação que te permitia submeter uma lista de compras e te dava o supermercado, à tua volta, onde essa lista é mais barata. Estarias disposto a ir a esse supermercado em vez do teu habitual, mesmo não sabendo onde é que os produtos estão e perder um pouco mais tempo à procura deles, para poupar mais dinheiro?

### Respostas:

Entrevistado	Resposta a)	Resposta b)
<b>Raquel</b>	Quando vou para a frente e para trás em corredores que já estive; Não tenho a minha lista organizada de acordo com os corredores do continente; Não consigo analisar os rótulos nutricionais, não é fácil decidir quanto à compra de alguns alimentos em função do preço vs. boa tabela nutricional; Quando os produtos não têm o preço afixado ou estão mal afixados.	Não, por vezes os preços mais baratos dos produtos não compensam em tempo à procura dos produtos na nova loja que fui, ou o que gasto em gasolina por ter ido mais longe, se as recomendações vão ser sempre mutáveis, não é um investimento. Prefiro acompanhar os preços na loja que vou habitualmente.
<b>Pedro</b>	Ir de propósito comprar algo e não haver.	Não, prefiro ir ao habitual, que me é mais confortável.
<b>Bruno</b>	Não encontrar o produto que procuro de alguma marca específica.	Estaria disposto a ir a esse supermercado se a distância não fosse longa e dependeria do produto. Por exemplo se fosse fruta/legumes apesar de ser mais barato prefiro ir ao que vou normalmente porque sei que a qualidade é boa, enquanto se fosse algum que sei que a qualidade não é tão boa não iria. Acho que só iria se fosse por alguma marca em que o produto não muda a qualidade de supermercado para supermercado.
<b>Fábio</b>	Fico frustrado quando mais perto é o mais caro e perco tempo a ver os preços.	Sim usaria essa aplicação

<b>Dora</b>	Os produtos que quero comprar às vezes não estão em promoção.	Se a totalidade da poupança compensar, sim usava a aplicação.
<b>João</b>	Não poder comprar tudo o que quero.	Sim, para comprar os mesmos produtos mais baratos, estava disposto a mudar de supermercado.
<b>Rita. A</b>	Procura pelos produtos que quero.	Posto como está, podia não estar disposta (tempo, deslocação e espaço também têm um preço); mas para quem está por dentro destas coisas da transformação digital, se uma app fizesse isso também devia poder submeter essa lista de compras online do supermercado em causa. Aí sim eu ia utilizar de certeza.
<b>António. T</b>	Tempo e dinheiro.	Sim com certeza.
<b>Pedro. A</b>	Haver filas para entrar e as pessoas andarem a passear nas compras.	Se a diferença de preço se justificasse, sim.
<b>Isabel</b>	Nenhuma.	Eventualmente, desde que efetivamente mais próximo de casa e se associado ao “preço mais baixo” fosse possível verificar a “origem do produto”.
<b>Catarina</b>	Não me chateia ir às compras.	Só se a diferença de preço for muito significativa ou vou acabar por comprar mais coisas enquanto procuro o que quero e não sei onde está.
<b>Duarte</b>	Nesta fase, haver filas em todo o lado.	Sim se a diferença de preços fosse significativa.
<b>Vera</b>	Não gosto de ter de comprar algo que preciso. Nunca gostei. Agora, acresce o facto de querer terminar o mais rapidamente possível.	Na linha da resposta ao 1, só concordaria se o preço valesse bem a diferença (não gosto de ter de procurar as coisas que preciso porque não sei onde estão; não me agrada a ideia de demorar ainda mais tempo).

<b>Rosa</b>	Tempo despendido a fazer compras.	Não, tendo em conta o mundo de promoções onde vivemos é difícil de ter só um sítio onde as coisas sejam no geral mais baratas, e na minha opinião, tempo é dinheiro, por isso perder tempo a percorrer corredores, à procura de algo, poderá (ou não) compensar a diferença de preço. No entanto ter uma app que avaliasse isso (preço vs. qualidade) e depois transformasse em lista de compras online (no supermercado realmente mais barato) e fazer entregas em casa, seria excelente.
<b>Miguel</b>	Poupança- evitar filas.	Com certeza.
<b>Inês. A</b>	O tempo que demora.	Era uma App útil, mas não sei se mudava onde já vou.
<b>Sara. C</b>	Não ter o produto e o tempo de espera.	Não, porque não tenho paciência para estar à procura dos produtos.
<b>Ana. A</b>	Não ter dinheiro suficiente para as compras necessárias e ter de deixar coisas por comprar ou comprar produtos não tanto do meu agrado por isso.	Obvio que sim.
<b>Joana. P</b>	Tempo de espera nas caixas para pagar porque nunca todas estão abertas, mesmo com a loja com muita gente.	Estaria disposta.
<b>Luísa</b>	Eu gosto de ir às compras e por exemplo, ia sempre ao pingo doce porque gostava efetivamente dos produtos marca branca deles. Mas nunca mais lá fui depois de perceber que é um caos...e que não há produtos vegetarianos. Não aprecio continente. Vou ao Jumbo e Aldi. Um pelo pão e supérfluos. O outro pelos frescos e pela diversidade de alimentos vegetarianos.	Não.

<b>Ana. P</b>	Número de vezes que tenho de ir, estar na fila.	Depende, considerando se forem produtos de marca branca ou não.
<b>Andreia</b>	Falta de dinheiro.	Sim.
<b>Filipa</b>	Estar na fila para entrar.	Idem, não me importaria.
<b>Sandra</b>	Atualmente, o facto de querer vir embora o mais rapidamente possível por causa da máscara.	Só se a diferença de preço justificasse mesmo, caso contrário não, então nos hipermercados perde-se muito tempo e energia à procura das coisas.
<b>Elsa</b>	Fila para entrar (novos tempos).	Sim, sem dúvida.
<b>Pedro</b>	Ter de estar à espera.	Dependia da distância e da poupança, mas à partida não.
<b>Marta</b>	O artigo não estar disponível	Sim estaria.
<b>Maria</b>	Fazê-las a toque de caixa (muito rapidamente).	Talvez, sim.
<b>Vera</b>	Ir às compras já é frustrante. Mas nesta fase diria que é a acumulação de pessoas e usar máscara.	Sim, o facto de ter produtos mais baratos teria peso. a perda de tempo a procurar os produtos seria só na primeira/segunda vez.
<b>Mafalda</b>	O tempo de espera nas filas.	Sim se são mais baratos sim.
<b>José</b>	Ter de esperar que alguém se decida em relação a determinado produto para depois se desviar e eu poder tirar o que quero, principalmente em tempos de Covid.	Sim
<b>Miriam</b>	O tempo que se perde e não encontrar o que se procura.	Sim.
<b>João F</b>	Pessoas que vão às compras para passear e empatam os outros.	Sim.
<b>Rui. P</b>	As pessoas saberem que existem outros à espera e fazerem como se nada fosse.	Sim.
<b>Alexandra</b>	O tempo que se perde e dinheiro que se gasta.	Sem dúvida.

<b>Patrícia</b>	Odeio pessoas, as luzes do supermercado, o ter de escolher entre milhares de marcas dos produtos (arroz, detergente roupa etc.)	Em princípio não, vou ao Lidl porque fica mais perto de casa e sei onde estão todos os produtos. Só se fosse uma poupança enorme, mas como gasto pouco no supermercado, não me iria fazer muita diferença.
<b>Sílvia. C</b>	Ter de esperar nas filas, lidar com pessoas e não saber o local dos produtos.	Sim estaria disposta por uma questão de poupança. O tempo perde-se, mas a carteira não estende e é preferível poupar.
<b>Sara .B</b>	A frustração é querer comprar uma coisa e já não existir aquilo com que vou na ideia ou não haver o meu número tamanho.	Sim estaria disponível se eram os mesmos produtos que uso, mas mais baratos, podia ir a outro supermercado.
<b>Paula</b>	Preços elevados.	Sim estaria disposta mesmo que se perdesse um pouco mais de tempo.
<b>Andreia. A</b>	Filas enormes, as pessoas que não sabem para onde estão a ir e não respeitam os outros, não haver várias coisas da minha lista ou haverem, mas estarem a preços muito altos.	Sim sem dúvida.
<b>Cláudio</b>	Se compras presenciais é ter demasiada gente na loja ou muito barulho.	Sim no final o que me interessa é mesmo poupar dinheiro, mesmo que não seja com os produtos da mesma marca, mas que sejam de qualidade semelhante.
<b>Alice</b>	Quando a loja está muito desorganizada e confusa, não percebo a logica das coisas.	Sim se tivesse tempo e não fosse algo que tivesse de comprar o mais rápido possível fazia esse esforço

**Tabela 11. Respostas às entrevistas**

## **Apêndice B - Tarefas dos Testes de Usabilidade**

**Tarefa 1** – Adicionar um produto ao sistema.

- 1- Fazer login no sistema com o email teste@teste.com e a palavra-passe “teste”;
- 2- Ir à opção de adicionar novo produto;
- 3- Adicionar os detalhes do produto (nome, preço, imagem)
- 4- Submeter e verificar se aparece mensagem de feedback com a conclusão da tarefa com sucesso.

**Tarefa 2** – Alterar preço de um produto.

- 1- Fazer login no sistema com o email teste@teste.com e a palavra-passe “teste”;
- 2- Pesquisar um produto de qualquer supermercado;
- 3- Editar o preço;
- 4- Submeter e verificar se aparece mensagem de feedback com a conclusão da tarefa com sucesso.

**Tarefa 3** – Reportar erro de preço.

- 1- Fazer login no sistema com o email teste@teste.com e a palavra-passe “teste”;
- 2- Pesquisar um produto de qualquer supermercado;
- 3- Reportar o erro de preço;
- 4- Submeter e verificar se aparece mensagem de feedback com o envio para aprovação do Administrador.

**Tarefa 4** – Adicionar produto à lista de compras.

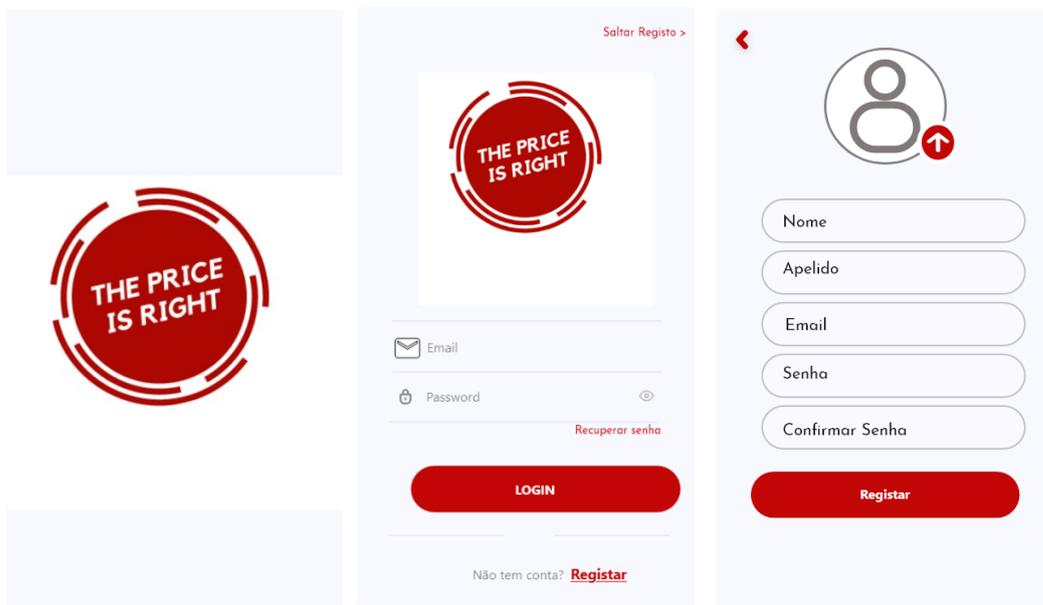
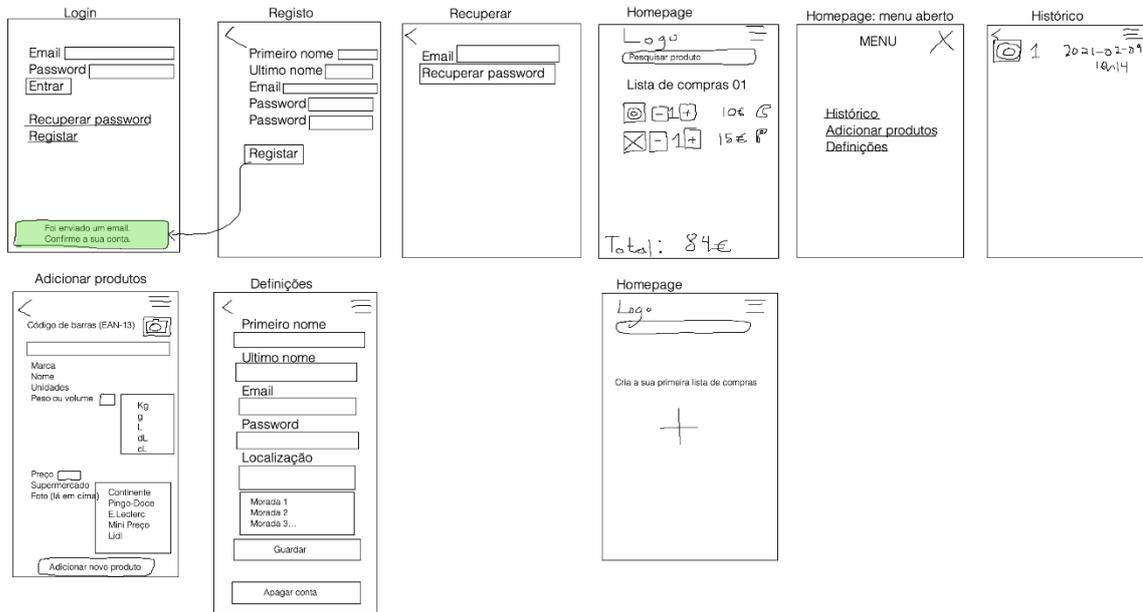
- 1- Fazer login no sistema com o email teste@teste.com e a palavra-passe “teste”;
- 2- Pesquisar um produto de qualquer supermercado;
- 3- Adicionar produto à lista de compras;
- 4- Submeter e verificar se produto ficou na lista com sucesso.

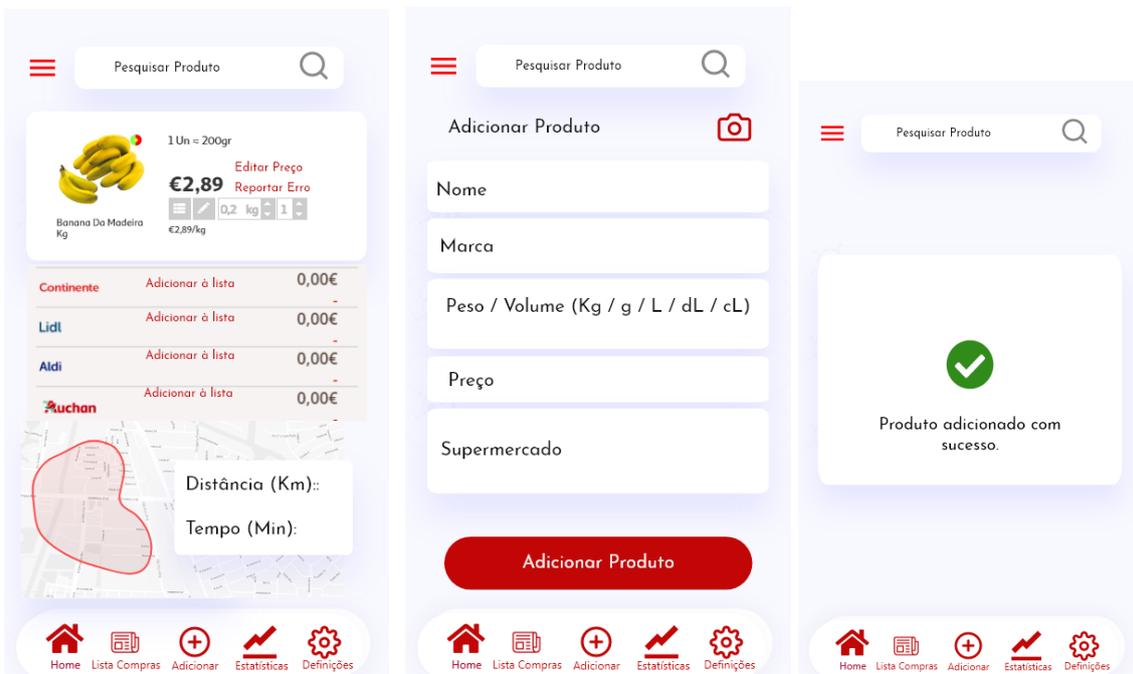
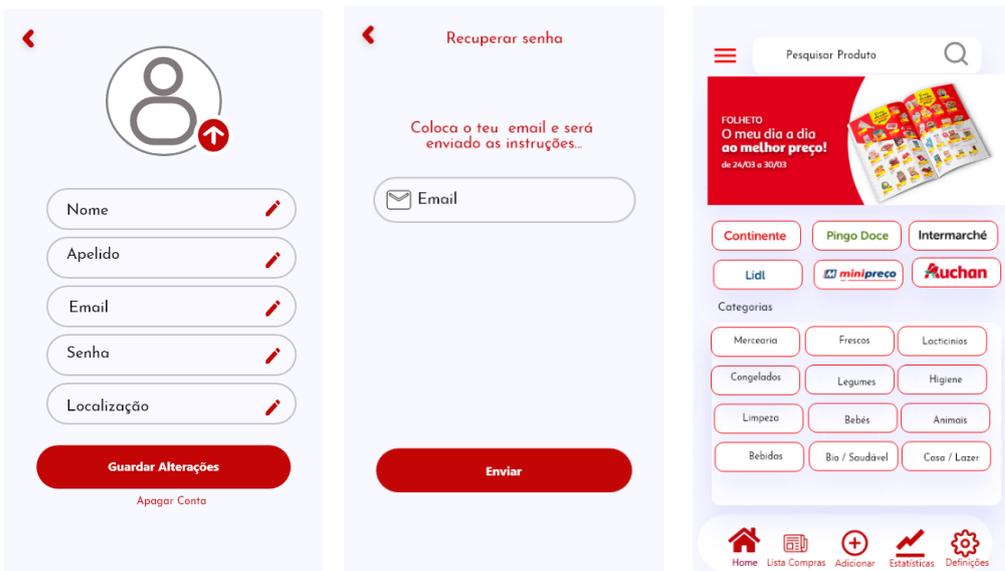
**Tarefa 5** – Verificar supermercados que parecem no mapa se correspondem à localização.

- 1- Entrar na homepage;
- 2- Permitir acesso à localização;
- 3- Confirmar que aparecem os supermercados à sua volta.

## Apêndice C - Protótipos Não Funcionais

### Mobile:





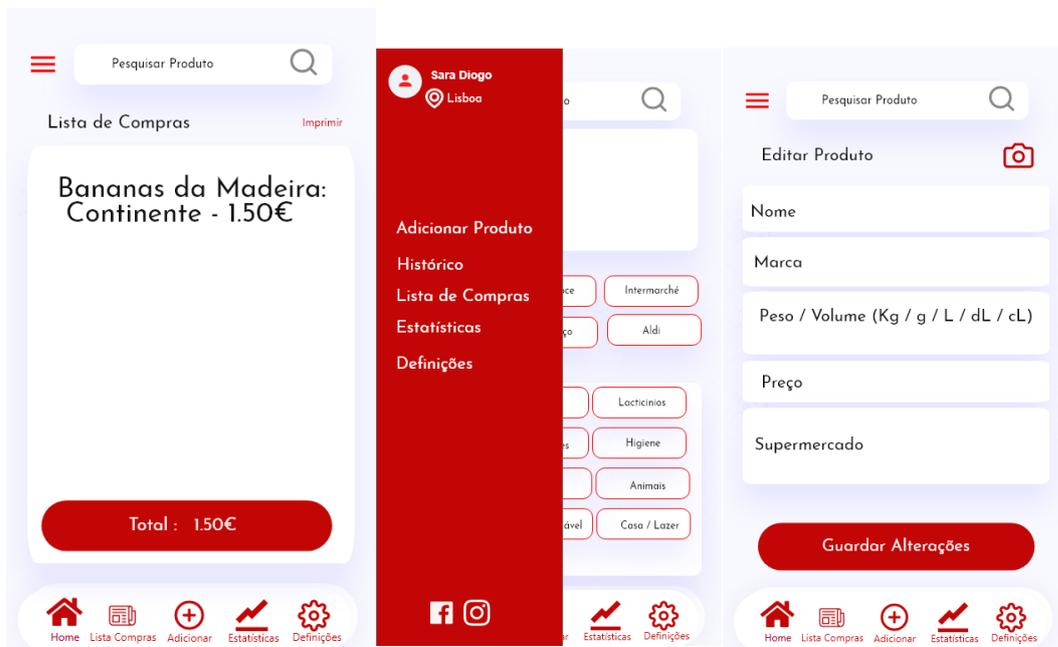
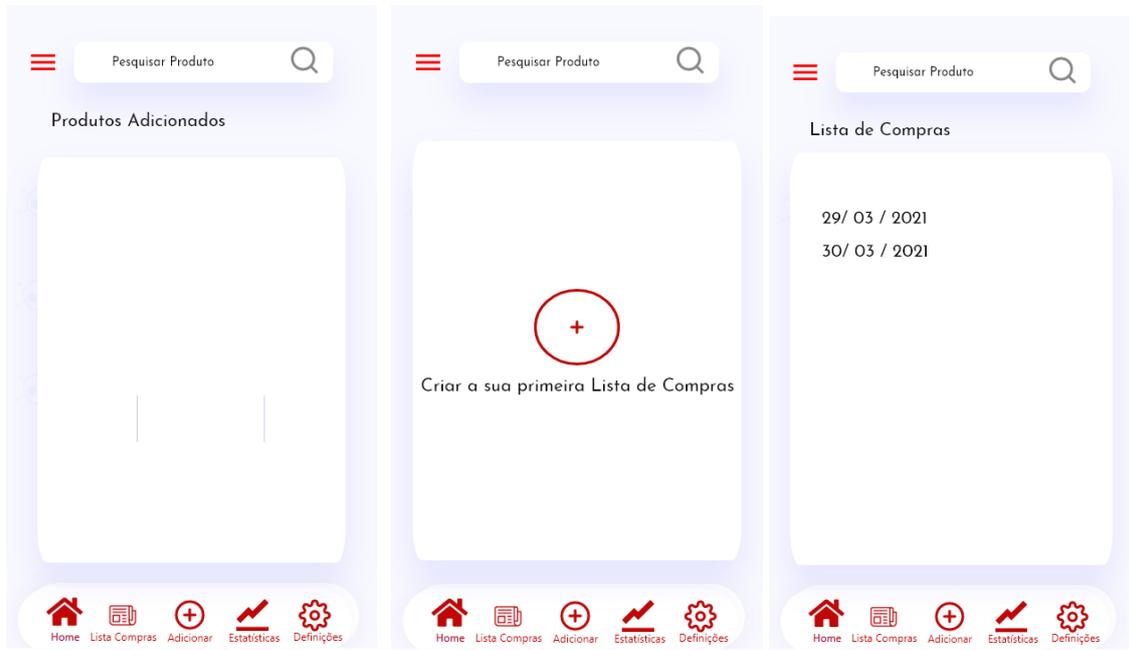
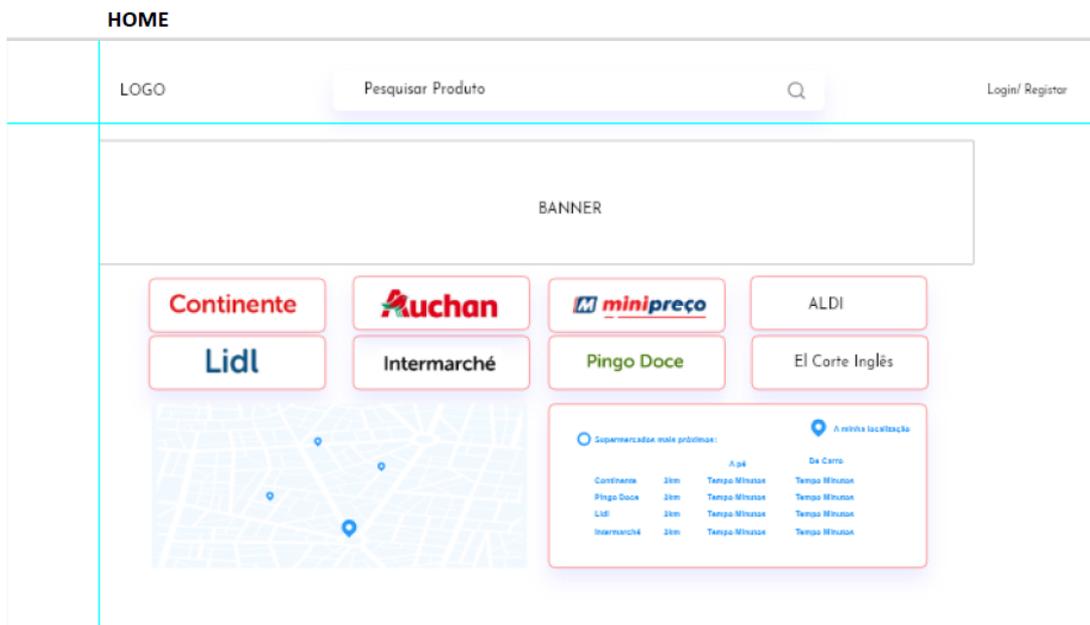




Figura 25. Protótipo Não Funcional - Mobile.

**Web:**



HOME COM REGISTO

PRODUTOS PESQUISA

LOGO

Pesquisar Produto

Sara Diago

PRODUTOS FRESCOS

SUB-CATEGORIAS

SUB-CATEGORIAS

SUB-CATEGORIAS

SUB-CATEGORIAS

SUPERMERCADOS

**Bolacha Maria**  
Bolacha Maria Integral, Saturno, 220g

10.0€ / un [Reportar Erro](#) [Editar Preço](#)

Quantidade: 1    Peso / Volume: Unidade (Lbs)    [Adicionar à Lista](#)

<b>Continente</b>	<a href="#">Adicionar a lista</a>	0,00€
<b>Lidl</b>	<a href="#">Adicionar a lista</a>	0,00€
<b>Aldi</b>	<a href="#">Adicionar a lista</a>	0,00€
<b>Auchan</b>	<a href="#">Adicionar a lista</a>	0,00€

Distância (Km):  
Tempo (Min):

LOGO

Pesquisar Produto

Sara Diago

**Lista de Compras**

PRODUCT	QUANTITY	PRICE
Bolacha Maria Continente	2.0	20€ 10.0€ / un

Total: 20€

LOGO

Pesquisar Produto

Sara Diago

**Registrar Produto**

Title  
Bolacha Maria

Não encontra o produto? [Clique aqui](#)

Price

Supermercado  
Continente

[Registrar](#)

LOGO

Pesquisar Produto



Sara Diogo

### Criar Novo Produto

Categoria  
Tartes e Bolos

Title  
Title

Quantidade: 1      Peso / Volume: Un

Description

Image:  
Escolher ficheiro Nenhum ficheiro selecionado

**Criar novo Produto**

LOGO

Pesquisar Produto



Sara Diogo

### Estadísticas de Consumo

PRODUCT	QUANTITY	PRICE
Bolacha Maria Cacelamento	2.0	20€ 10.00€/un

**Total: 20€**



**Figura 26. Protótipo Não Funcional - Website**

## Apêndice D – Questionário de Satisfação

### Questionário de Satisfação

Este questionário serve para relatar como foi a sua experiência ao utilizar a aplicação Price is Just ([www.priceisjust.com](http://www.priceisjust.com))

1-Género?

Feminino

Masculino

Prefiro não dizer

Outra: \_\_\_\_\_

2-Idade?

Menos de 18 anos

18 -30 anos

30 - 64 anos

Mais de 65 anos

3-Facilidade de navegação?

Muito má

Má

Normal

Boa

Excelente

4 -Facilidade na procura de informação?

Muito má

Má

Normal

Boa

Excelente

5 -Design dos ecrãs?

Muito mau

Mau

Normal

Bom

Excelente

6-Qualidade do Vocabulário?

Muito Mau

Mau

Normal

Bom

Excelente

6-Qual o grau de satisfação sobre esta aplicação?

1 - Muito Insatisfeito

2- Insatisfeito

3 - Normal

4 - Muito Satisfeito

7-Utilizaria esta aplicação?

Sim

Não

Se na pergunta anterior respondeu "Sim", onde utilizaria esta aplicação?

Website

Aplicação Móvel

Sugestões?

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

**Submeter**

Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) - [Termos de Utilização](#) - [Política de privacidade](#)

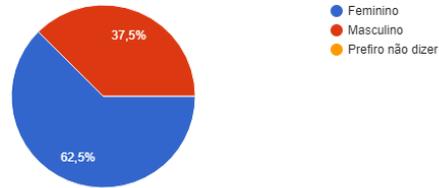
Google Formulários

Figura 27. Questionário de Satisfação

## Apêndice E – Resultados do Questionário de Satisfação

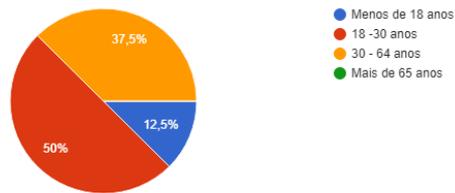
1-Gênero?

8 respostas



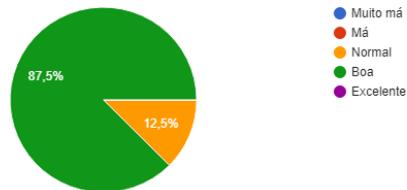
2-Idade?

8 respostas



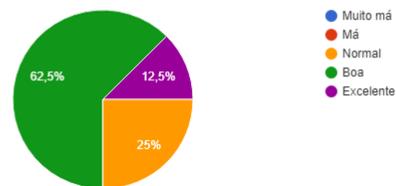
3-Facilidade de navegação?

8 respostas



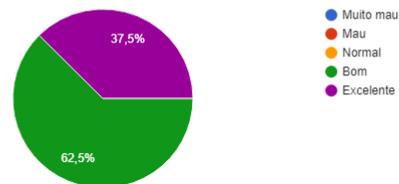
4 -Facilidade na procura de informação?

8 respostas



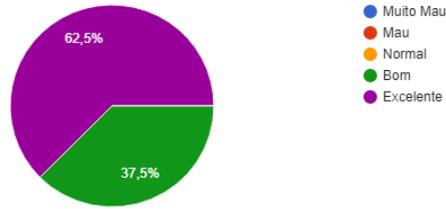
5 -Design dos ecrãs?

8 respostas



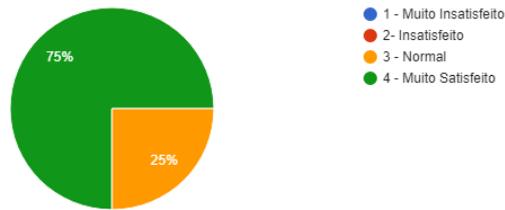
6-Qualidade do Vocabulário?

8 respostas



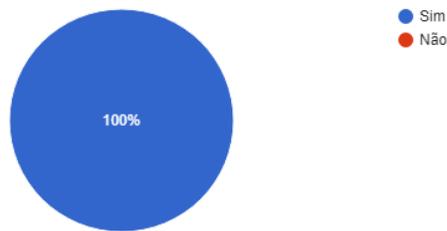
6-Qual o grau de satisfação sobre esta aplicação?

8 respostas



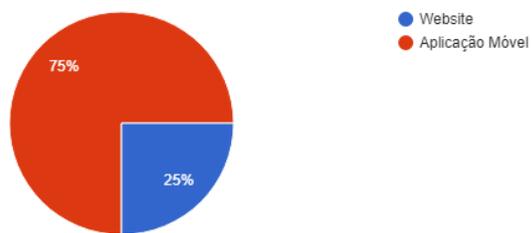
7-Utilizaria esta aplicação?

8 respostas



Se na pergunta anterior respondeu "Sim", onde utilizaria esta aplicação?

8 respostas



## Apêndice F – Resultado Final

### ➤ Website

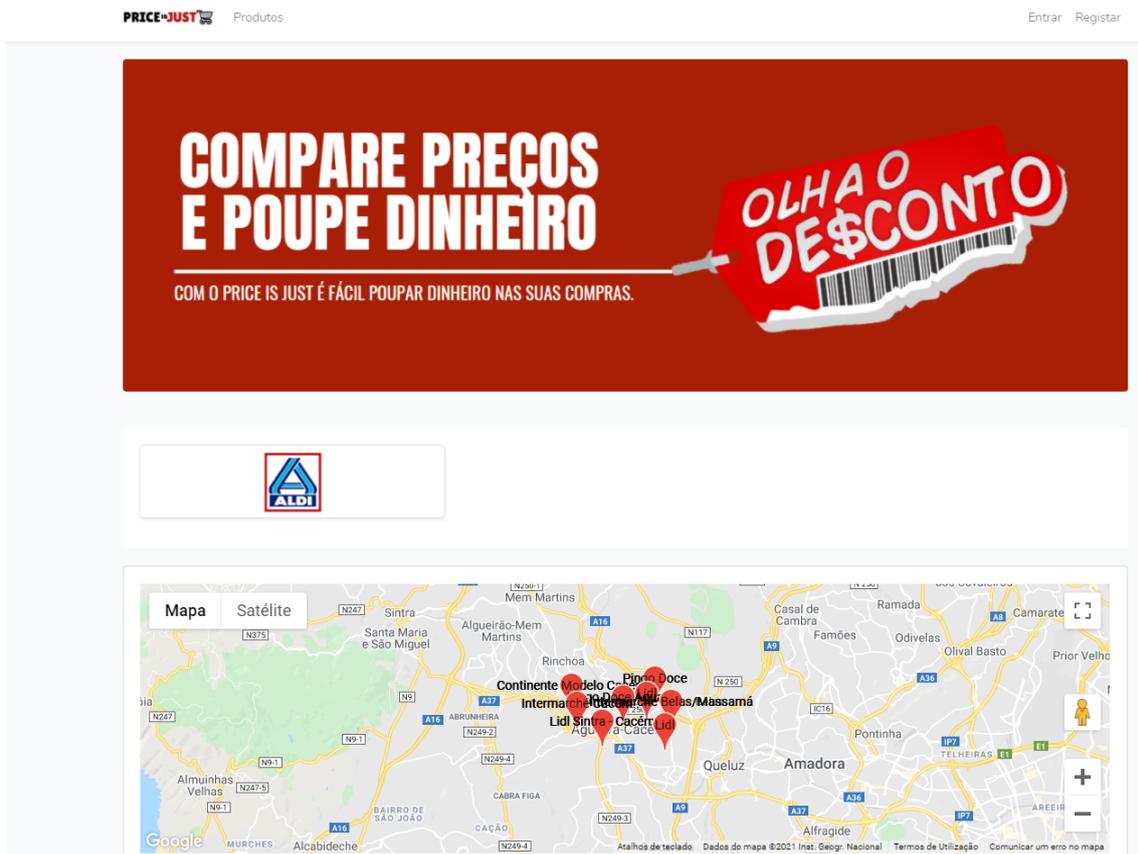


Figura 28. Home Page

The screenshot shows a registration form titled 'Registrar'. It contains five input fields: 'Nome:', 'Apelido:', 'Endereço de Email:', 'Palavra-Passe:', and 'Palavra-Passe:'. Below the fields is a red button labeled 'Registrar'.

Figura 29. Registo de Utilizador

PRICE-JUST  Produtos Sara ▾

Filtrar Produtos ▾

Selecionar Categoria ▾

Filtrar Supermercados ▾

- Continente
- Pingo Doce
- Intermarché
- Lidl
- Auchan
- Mini Preço
- Aldi

Pesquisar Produtos

9 Items Encontrados

 <p><b>Bolacha Maria Cuétara</b> Bolachas / Continente 1.79€ / un <a href="#">Adicionar à Lista</a></p>	 <p><b>Bolacha Recheada Oreo</b> Bolachas / Continente 1.49€ / un <a href="#">Adicionar à Lista</a></p>	 <p><b>Belgas Original</b> Bolachas / Continente 1.49€ / un <a href="#">Adicionar à Lista</a></p>
 <p><b>Bolacha Cracker Tuc</b> Bolachas / Continente 1.05€ / un <a href="#">Adicionar à Lista</a></p>	 <p><b>Donuts</b> Tartes e Bolos / Pingo Doce 1.59€ / un <a href="#">Adicionar à Lista</a></p>	 <p><b>Bollycao</b> Tartes e Bolos / Pingo Doce 1.44€ / un <a href="#">Adicionar à Lista</a></p>

Figura 30. Lista de Produtos



**Bolacha Maria Cuétara**  

Bolacha Maria Cuétara Pack+ 33% Grátis 4X200 G.  
Bolachas / Continente

**1.79€ / un**

Quantidade     Peso/volume  [Adicionar à Lista](#)

Produtos Semelhantes

 <p><b>Bolacha Recheada Oreo</b> Bolachas / Continente 1.49€ / un <a href="#">Adicionar à Lista</a></p>	 <p><b>Belgas Original</b> Bolachas / Continente 1.49€ / un <a href="#">Adicionar à Lista</a></p>	 <p><b>Bolacha Cracker Tuc</b> Bolachas / Continente 1.05€ / un <a href="#">Adicionar à Lista</a></p>
--	--	---

Figura 31. Descrição do Produto

### Registar Produto

Produto

Bolacha Maria Cuétara

Não encontra o produto? [Clique Aqui.](#)

Preço

1,00

Supermercado

Continente

**Criar produto**

Figura 32. Registar Novo Produto

PRICE-JUST Produtos Sara

PRODUTO	QUANTIDADE	PREÇO	
 Bolacha Maria Cuétara Continente	1.0	1.79€ 1.79€/un	
 Bolacha Recheada Oreo Continente	1.0	1.49€ 1.49€/un	

Supermercado: Continente

Total: 3.28€

**Concluir lista**

Figura 33. Lista de Compras

PRICE-JUST Produtos Sara

### Estatísticas

Visualizar os seus gastos.

Totais gasto	Média por Lista
3.28€ Total gasto	3.28€ % Média total por lista
Total Mensal	
3.28€ Este mês	0€ Mês anterior

Figura 34. Estatísticas de Consumo

➤ Mobile

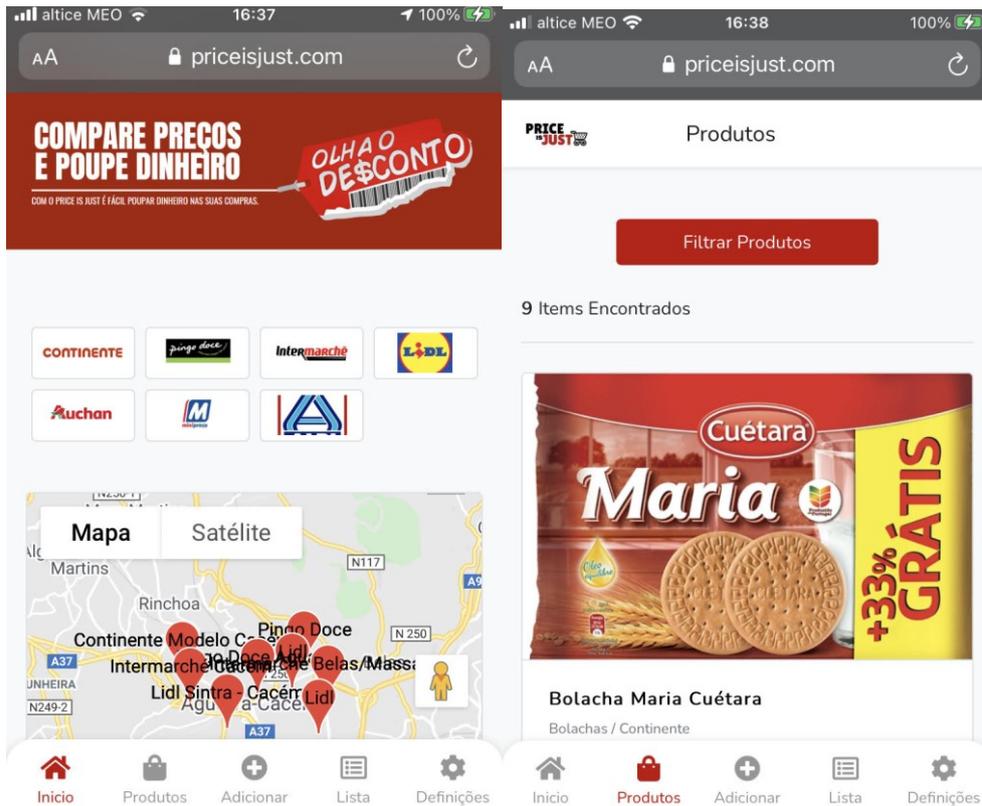


Figura 35. Home Page e Lista de Produtos

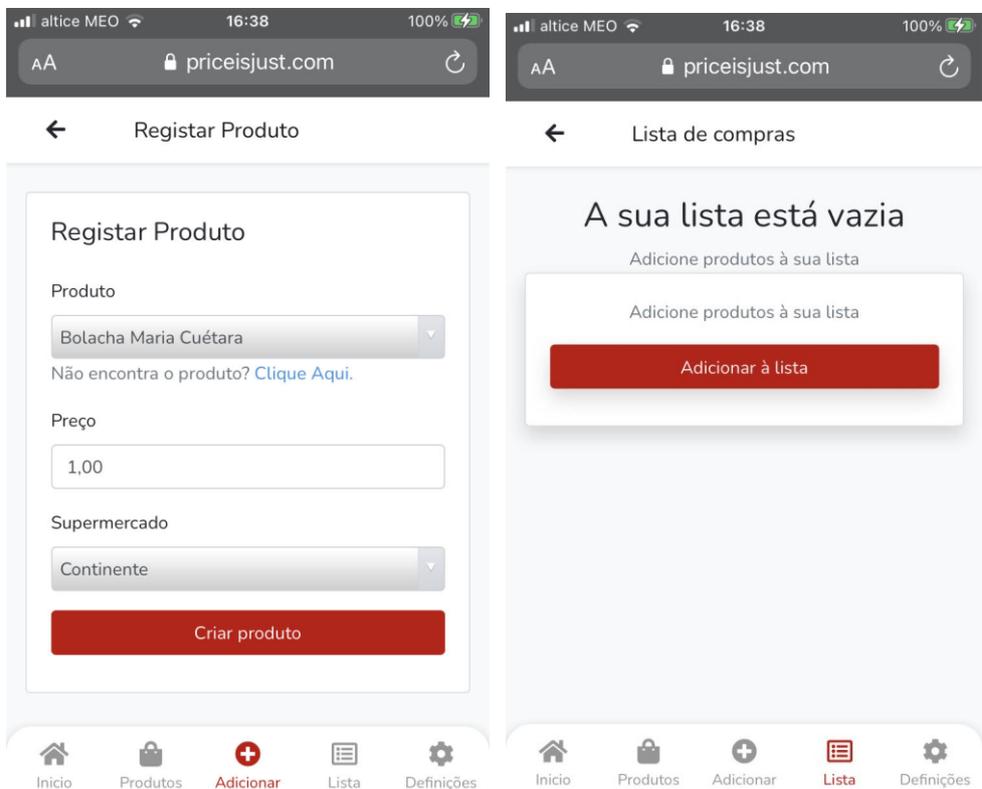
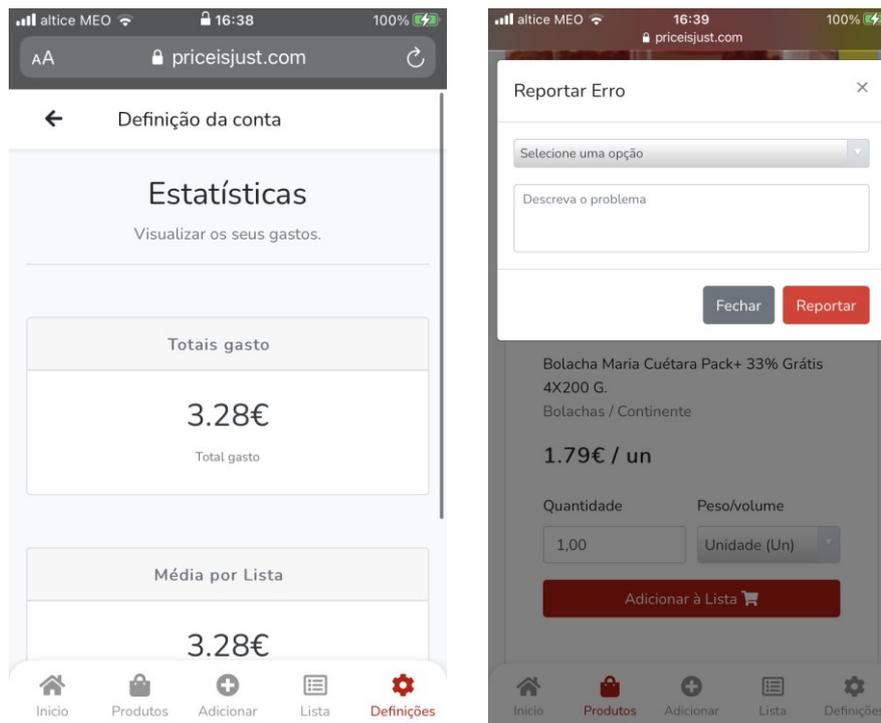


Figura 36. Registo de Novo Produto e Lista de Compras



**Figura 37. Estatísticas de Consumo e Reportar Erro**

## **Glossário**

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
MVC	Model View Controller
TLS	Transport Layer Security
SSL	Secure Socket Layers
HTTPS	Hyper Text Transfer Protocol Secure
HTTP	Hyper Text Transfer Protocol
URL	Uniform Resource Locator