



## **Acessibilidade na Web e no mundo e-commerce**

**CATARINA RODRIGUES COUTO**

Outubro de 2018

# **Acessibilidade na *Web* e no mundo *e-commerce***

**Catarina Rodrigues Couto**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Engenharia de Software**

**Orientador: Nuno Silva**

**Supervisor: Arménio Miranda**

Porto, outubro 2018



# Dedicatória

«Dedico este trabalho ao meu irmão, Samuel Couto, por acreditar sempre que posso atingir os meus objetivos e por ser uma enorme inspiração em todas as etapas da minha vida pessoal, académica e profissional. »



# Resumo

Nos últimos anos as compras *online* cresceram significativamente, devido à rápida evolução da tecnologia. Com isto as empresas da indústria de comércio *online* precisaram de pensar em formas de gerar mais lucro e de conseguirem atrair mais consumidores, melhorando o modo como os mesmos navegam nos *websites* e efetuam compras. Neste tipo de indústria, pensar em acessibilidade é garantir que um maior número de pessoas/consumidores consegue ver e navegar no conteúdo do *website*, gerando assim mais taxa de conversão e mais receitas para a própria marca. Consequentemente, não se pode menosprezar nenhum tipo de consumidores e é assim que a acessibilidade se torna num fator muito importante neste tipo de negócio. A acessibilidade permite que todo o conteúdo de um *website* seja acedido por um público o mais amplo possível, incluindo todas as pessoas, quer possuam incapacidades ou não. Desta forma, a acessibilidade na *web* visa permitir que os utilizadores possam consultar e interagir com a informação digital, usando qualquer dispositivo que tenham.

Com vista a atingir o objetivo de tornar os *websites e-commerce* acessíveis, irá ser desenvolvido um protótipo, composto por vários componentes (partes) já preparados nesta vertente, que servirão de base comum a vários *websites e-commerce* da organização. Para desenvolver este projeto é necessária a pesquisa e implementação de acessibilidade e respetiva aplicação de conceitos no protótipo. Quer o protótipo base quer os componentes do mesmo, devem ser incluídos em diversos *websites e-commerce*, com a finalidade de verificar se o objetivo foi atingido ou não. É também necessário verificar a satisfação quer dos utilizadores finais, quer das partes interessadas no projeto através de inquéritos finais e testes de usabilidade. Por fim, este protótipo deve servir como sensibilização à necessidade de implementação de acessibilidade e servir como referência de boas práticas do tema para qualquer projeto da organização.

**Palavras-chave:** Acessibilidade, Acessibilidade na *Web*, Incapacidade, *E-commerce*, Componentes



# Abstract

Over the last years, online shopping grew significantly due to the rapid evolution of technology. With this, e-commerce industry companies needed to think of new ways of generating profit and reach more consumers, improving the way they navigate and shop in their websites. In this type of industry, thinking in accessibility is to ensure that a greater number of persons/consumers can navigate the content of the website, increasing its conversion rate, thus leading to more income. Consequently, one cannot belittle any type of consumer, and that's how accessibility becomes such a relevant factor in this type of business. Accessibility allows that all the website's content is available to a broader audience, including everyone, whether they have some kind of disability or not. With this, accessibility on the web aims to enable users to consult and interact with digital information, using whatever device they have.

In order to make e-commerce sites accessible, a prototype will be developed, composed of several components (parts) already prepared in this strand, which will serve as a common base for various e-commerce websites of the organization. For this project's development, is required the research and implementation of accessibility and their respective application of concepts in the prototype. Both the base prototype and its components should be included in multiple e-commerce websites, with the goal to verify if the goal was actually achieved. It's also necessary to verify both the end-users satisfaction as well as the project's stakeholders, through final surveys and usability tests. Lastly, this prototype must act as sensibilization to the need for implementing accessibility and be a reference of the subject's good practices for every organization project.

**Keywords:** Accessibility, Web Accessibility, Disability, E-commerce, Components



# Agradecimentos

Em primeiro lugar e de forma especial, quero agradecer a todas as pessoas que fazem parte da minha vida pessoal e que contribuem, direta ou indiretamente, para o meu bem-estar e que me apoiam em todas as decisões que tomo. Obrigada por estarem presentes em todos os obstáculos que enfrentei até hoje e por partilharem sorrisos comigo em todos os momentos de vitória e superação pessoal.

A todas as pessoas constituintes da instituição de ensino (ISEP) a que pertenço, em especial ao Departamento de Engenharia Informática (DEI) pelos conhecimentos de excelência prestados e por todo o apoio e incentivo por parte de todos os docentes para que todos os meus objetivos académicos fossem atingidos.

Ao professor Doutor Nuno Silva por me ter orientado e por me mostrar sempre as melhores formas de atingir bons resultados, nunca esquecendo todas as críticas construtivas nem o tempo que disponibilizou para me acompanhar e encorajar.

À Farfetch por me proporcionar a oportunidade de realizar o presente projeto em tão prestigiada organização. Quero também agradecer à minha equipa, *Black & White*, pelo encorajamento e por todos os momentos de boa disposição e aprendizagem que me prestaram proporcionando assim o meu crescimento e evolução profissional e académica.

Deixo um especial agradecimento a toda a minha família que sempre me apoiou e nunca me deixou desanimar em nenhuma circunstância da vida.

Aos que estão presentes e aos que partiram deixando saudade.

Aos meus verdadeiros amigos e confidentes que me ensinaram a ser quem sou e por me alegrarem todos os dias.

A todos estes e mais alguns, o meu sincero agradecimento: obrigada!



# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto	1
1.1.1	Farfetch	1
1.1.2	Negócio	2
1.1.3	Acessibilidade	3
1.2	Problema	4
1.3	Objetivos	4
1.4	Abordagem	5
1.5	Planeamento do projeto	6
1.6	Estrutura do documento	7
<b>2</b>	<b>Estado de arte</b>	<b>9</b>
2.1	Enquadramento teórico	9
2.1.1	Interação humano-computador	9
2.1.2	Tecnologias assistivas	10
2.1.3	Acessibilidade e Usabilidade	11
2.1.4	E-commerce	13
2.1.5	Princípios da Inclusão	13
2.1.6	Acessibilidade é para todos	14
2.1.7	Leis e normas de acessibilidade	15
2.1.8	<i>Americans with Disabilities Act (ADA)</i>	16
2.2	Fatores de acessibilidade	17
2.2.1	Fatores económicos	17
2.2.2	Fatores geográficos	18
2.2.3	Diversidade de utilizadores	18
2.3	Enquadramento tecnológico	21
2.3.1	Acessibilidade em <i>e-commerce</i>	21
2.3.2	Diretrizes da acessibilidade	22
2.3.3	API de acessibilidade	24
2.3.4	Árvore de acessibilidade	25
2.4	Análise de soluções existentes	26
2.4.1	Screen-reader	26
2.4.2	Ferramentas de auditoria/avaliação	29
2.4.3	<i>Websites</i> acessíveis	30
2.5	Sumário	32
<b>3</b>	<b>Análise de valor</b>	<b>33</b>
3.1	Modelo New Concept Development	33
3.2	Identificação da oportunidade	34
3.2.1	Responsabilidade de garantir acessibilidade	35

3.3	Análise da oportunidade .....	37
3.3.1	Causas dos problemas de acessibilidade.....	37
3.3.2	Tipos de erros de acessibilidade .....	38
3.3.3	Vantagens do projeto .....	40
3.3.4	Vantagens da acessibilidade .....	40
3.3.5	Desvantagens da acessibilidade .....	41
3.4	Geração e enriquecimento de ideias .....	41
3.5	Seleção de ideias.....	42
3.5.1	Análise hierárquica (AHP) .....	42
3.6	Definição do conceito .....	45
3.7	Modelo de Canvas .....	46
<b>4</b>	<b>Análise e <i>Design</i> .....</b>	<b>49</b>
4.1	Análise .....	49
4.1.1	Requisitos funcionais.....	49
4.1.2	Requisitos não funcionais.....	51
4.2	Arquitetura de software .....	52
4.2.1	Princípios arquiteturais .....	52
4.2.2	Vista lógica.....	55
4.2.3	Vista de implementação.....	58
4.2.4	Vista de processo .....	60
<b>5</b>	<b>Implementação.....</b>	<b>63</b>
5.1	Cuidados gerais.....	63
5.1.1	Semântica HTML .....	63
5.1.2	Título das páginas.....	65
5.1.3	Landmarks.....	66
5.1.4	WAI-ARIA .....	69
5.1.5	<i>Tab order &amp; tabindex</i> .....	74
5.1.6	Estrutura da página <i>web</i> .....	76
5.1.7	Page regions .....	77
5.1.8	Labeling regions.....	78
5.1.9	Headings.....	79
5.1.10	Content structure.....	81
5.1.11	Focus e outline .....	82
5.1.12	Alternative text .....	84
5.1.13	Nomes acessíveis.....	86
5.1.14	Contraste de cores.....	87
5.1.15	Tratamento de erros .....	87
5.2	Componentes .....	88
5.2.1	Skip link .....	88
5.2.2	Visually Hidden.....	90
5.2.3	Botão.....	91
5.2.4	Product card.....	91
5.2.5	Select .....	93
5.2.6	SvgIcon.....	94

5.2.7	Pagination.....	95
5.2.8	Notificações de rotas .....	95
5.2.9	Header.....	96
5.2.10	Footer.....	97
5.3	Páginas.....	98
5.3.1	Homepage.....	98
5.3.2	Lista de produtos .....	99
5.3.3	Detalhe de um produto .....	100
5.3.4	Lista de desejos.....	101
5.3.5	Carrinho de compras .....	102
5.3.6	Checkout .....	103
5.3.7	Página de erro.....	103
5.4	Atalhos e opções das tecnologias assistivas.....	104
5.5	Accessibility statements .....	104
5.5.1	Compromisso aberto .....	105
5.5.2	Diretrizes e padrões de acessibilidade.....	106
5.5.3	Exceções.....	106
5.5.4	Informações de contacto .....	107
5.6	Reutilização e manutenção .....	107
5.6.1	Criação de componentes e sua utilização.....	107
5.6.2	Uso de componentes .....	108
5.6.3	Manutenção de componentes.....	108
5.7	Processo de contribuição .....	108
5.7.1	Templates.....	108
5.7.2	Issues.....	109
5.7.3	Merge requests .....	109
5.7.4	Commits .....	110
5.8	Testes unitários .....	111
5.8.1	Tecnologia e método.....	111
5.8.2	Coverage associada.....	112
<b>6</b>	<b>Avaliação.....</b>	<b>113</b>
6.1	Abordagem.....	113
6.1.1	Desenvolvedores .....	113
6.1.2	Clientes finais .....	114
6.1.3	Cliente/loja.....	114
6.2	Preparação de experiências.....	115
6.2.1	Experiências realizadas.....	115
6.2.2	Guião.....	115
6.2.3	Inquéritos .....	116
6.3	Avaliação estatística/objetiva.....	119
6.3.1	Feedback com a Léonie Watson .....	119
6.3.2	Resultados de inquéritos .....	123
6.3.3	Satisfação do uso das boas práticas de acessibilidade .....	130
6.3.4	Satisfação do fluxo da aplicação com o uso de tecnologias assistivas .....	131
6.3.5	Satisfação e resultados através do uso de ferramentas de auditoria.....	131

6.3.6	Vontade do utilizador e satisfação geral .....	131
<b>7</b>	<b>Conclusões e trabalho futuro .....</b>	<b>133</b>
7.1	Objetivos alcançados.....	133
7.2	Limitações.....	134
7.3	Trabalho futuro.....	134
7.4	Apreciação final e pessoal.....	135

# Lista de Figuras

Figura 1 - Diagrama de Componentes de negócio .....	2
Figura 2 - Comunicação entre os <i>browsers</i> , tecnologias assistivas e a API de acessibilidade (Accessible) .....	25
Figura 3 – Árvore de acessibilidade do projeto.....	26
Figura 4 - <i>Website</i> acessível: Centro de Controlo de doenças dos Estados Unidos .....	30
Figura 5 - <i>Website</i> acessível: Administração da Segurança Social dos Estados Unidos .....	31
Figura 6 - <i>Website</i> acessível: <i>Website</i> oficial do governo dos Estados Unidos da América.....	32
Figura 7 - Modelo NCD ( <i>New Concept Development</i> ) .....	34
Figura 8 - Árvore hierárquica de decisão .....	43
Figura 9 - Diagrama de Casos de Uso da aplicação .....	50
Figura 10 - Níveis de <i>atomic design</i> (Frost, 2013) .....	54
Figura 11 - Conceito atomic design associado ao projeto .....	54
Figura 12 - Diagrama de alto nível representativo do sistema .....	56
Figura 13 - Diagrama de componentes com detalhe da solução implementada .....	57
Figura 14 - Diagrama representativo de árvore de componentes .....	57
Figura 15 - Diagrama explicativo de composição de componentes.....	57
Figura 16 – Contribuição para o projeto ao nível do átomo em conjunto com o <i>atomic design</i> .....	58
Figura 17 – Diagrama explicativo do projeto desenvolvido.....	58
Figura 18 – Diagrama explicativo do uso de <i>atomic design</i> ao nível da aplicação.....	59
Figura 19 - Diagrama de sequência de interações entre o utilizador e o sistema .....	60
Figura 20 - Exemplo de HTML semântico no <i>header</i> da aplicação.....	64
Figura 21 - Título da página inicial no separador do <i>browser</i> .....	65
Figura 22 - Utilização do <i>landmark navigation</i> com um <i>aria-label</i> apropriado .....	68
Figura 23 - Relação de <i>headings</i> e de respetivas subsecções (Nomensa) .....	80
Figura 24 - Estilos incluídos através do <i>Javascript</i> para retirar o <i>outline</i> a navegar com o rato	84
Figura 25 - <i>Link</i> para <i>homepage</i> quando é utilizada a navegação através do teclado .....	84
Figura 26 - Tratamento de erro ao adicionar um produto ao carrinho de compras .....	88
Figura 27 – Resultado visual do uso do componente <i>Skip Link</i> .....	90
Figura 28 – Resultado visual do uso do componente <i>VisuallyHidden</i> num elemento <i>focusable</i> .....	91
Figura 29 - Representação do <i>ProductCard</i> com o <i>link</i> para a página de detalhe do produto e com o <i>link</i> para avançar para o próximo produto.....	93
Figura 30 - Aspeto visual do componente <i>Pagination</i> .....	95
Figura 31 - <i>Header</i> da aplicação .....	97
Figura 32 - <i>Footer</i> da aplicação .....	98
Figura 33 - <i>Homepage</i> da aplicação .....	99
Figura 34 – Lista de produtos da aplicação.....	100
Figura 35 – Detalhe de um produto no projeto .....	101
Figura 36 – <i>Wishlist</i> do projeto .....	102

Figura 37 – <i>Bag</i> do projeto .....	102
Figura 38 - Página de <i>checkout</i> do projeto .....	103
Figura 39 - Página de erro do projeto.....	104
Figura 40 – Guião distribuído pelos <i>stakeholders</i> .....	116
Figura 41 – Inquérito Pergunta 1: Grau de satisfação no <i>header</i> .....	123
Figura 42 – Inquérito Pergunta 2: Grau de satisfação no <i>footer</i> .....	123
Figura 43 – Inquérito Pergunta 3: Grau de satisfação na <i>homepage</i> .....	124
Figura 44 – Inquérito Pergunta 4: Grau de satisfação na lista de produtos .....	124
Figura 45 – Inquérito Pergunta 5: Grau de satisfação no detalhe do produto .....	124
Figura 46 – Inquérito Pergunta 6: Grau de satisfação na <i>wishlist</i> .....	125
Figura 47 – Inquérito Pergunta 7: Grau de satisfação no <i>bag</i> .....	125
Figura 48 – Inquérito Pergunta 8: Grau de satisfação no <i>checkout</i> .....	125
Figura 49 – Inquérito Pergunta 9: Grau de satisfação com o fluxo do <i>screen-reader</i> do projeto .....	126
Figura 50 – Inquérito Pergunta 10: Grau de satisfação com o fluxo do uso do teclado no projeto .....	126
Figura 51 – Inquérito Pergunta 11: Ferramentas de auditoria utilizadas para testar o projeto .....	126

# Lista de Tabelas

Tabela 1 - Comparação de <i>screen-readers</i> existentes.....	28
Tabela 2 - Nível de complexidade de alguns erros de acessibilidade na <i>web</i> .....	39
Tabela 3 - Tabela de avaliação AHP.....	44
Tabela 4 - Matriz normalizada do método de avaliação HP .....	44
Tabela 5 - Pesos associados aos critérios de avaliação hierárquica .....	45
Tabela 6 - Modelo de CANVAS representativo do projeto .....	47
Tabela 7 - Lista de elementos nativos HTML5 e respetivos <i>landmarks</i> .....	69
Tabela 8 - Exemplo de ARIA <i>roles</i> utilizados ao longo do projeto.....	71
Tabela 9 - Exemplo de ARIA <i>properties</i> utilizados ao longo do projeto .....	73
Tabela 10 - Exemplo de ARIA <i>states</i> utilizados ao longo do projeto .....	74
Tabela 11 – Avaliações realizadas por <i>stakeholder</i> .....	115
Tabela 12 – Boas práticas sugeridas por Léonie Watson.....	120



# Lista de Acrónimos

ADA	Americans with Disabilities Act
API	Application programming interface
B2C	Business to Consumer
CLI	Command-line interface
CSS3	<i>Cascading Style Sheets</i> , versão 3 é um mecanismo para adicionar estilo, cores, fontes, espaçamento, entre outros) a um documento na web.
DEI	Departamento de Engenharia Informática
DX	Developer Experience
GIT	Sistema de controlo de versões usado no desenvolvimento de software
HTML5	<i>Hyper Text Markup Language</i> , versão 5 é uma linguagem para estruturação e apresentação de conteúdo para a internet.
ISEP	Instituto Superior de Engenharia do Porto
MEI	Mestrado em Engenharia Informática
NCD	New Concept Development
NPD	Novo Processo de Desenvolvimento
SCRUM	Metodologia ágil para gestão e planeamento de projetos
SEO	Search Engine Optimization
SPA	Single Page Application
URL	Uniform Resource Locator
UX	User Experience
WCAG	Web Content Accessibility Guidelines
WWW	World Wide Web
W3C	<i>World Wide Web Consortium</i> – principal organização responsável pela padronização da World Wide Web
WAT	Web Accessibility Toolbar



# 1 Introdução

Este capítulo apresenta o contexto, problema e objetivos do trabalho, bem como o seu respetivo planeamento e tarefas realizadas.

Seguidamente, é feita uma breve descrição da abordagem metodológica e tecnologias/ferramentas utilizadas no projeto.

Por fim, é descrita a organização do relatório para que o leitor compreenda toda a sua estrutura e conteúdos.

## 1.1 Contexto

Este subcapítulo apresenta os conceitos importantes inerentes ao projeto, como o termo acessibilidade e sua importância para o negócio. Além disso, é explicado o modelo de negócio da *Farfetch* e o modo como operam no desenvolvimento de *software*, e em particular o foco da equipa *Black & White*, na qual o projeto se insere.

### 1.1.1 Farfetch

“A *Farfetch* é uma plataforma *online* que dá acesso a uma seleção de peças de moda de luxo, por todo o mundo e possui como visão ser a plataforma tecnológica global para a indústria de moda de luxo, com o cliente sempre no coração e como foco principal” (Farfetch, 2008).

“Esta plataforma de *e-commerce* associa-se às melhores boutiques e marcas de luxo do mundo, localizadas desde Tóquio a Toronto e de Milão a Miami” (Farfetch, 2008), conseguindo assim um modelo de negócio exclusivo que garante uma variedade de peças exclusivas e com estilo único para comprar, de marcas de luxo estabelecidas com os novos e mais conhecidos *designers* da moda (Farfetch, 2008).

Fundada em 2008 pelo empresário português José Neves, a *Farfetch* é um *website* traduzido em 9 idiomas que torna mais fácil do que nunca comprar moda de luxo, pois ainda nada parecido existia como esta plataforma e é o que faz da mesma um destino único (Farfetch,

2008). Tem 11 escritórios espalhados pelo mundo e envia encomendas para mais de 190 países (Farfetch, 2008).

### 1.1.2 Negócio

*Black and White* é uma unidade de negócio independente pertencente à *Farfetch*. Segundo o fundador da *Farfetch*, José Neves, esta unidade de negócio desenvolve *websites e-commerce* de diversas marcas de luxo que, por estarem em parceria com a *Farfetch*, usam a plataforma e serviços da mesma para construir o seu modelo *e-commerce*, de forma modular, contendo diversas áreas: *design*, construção e operação sobre o *website*, serviço ao cliente e *marketing* digital (Business of Fashion, 2015).

Ao usar este serviço, as marcas podem ter acesso a tudo o que o *website* da plataforma *online* *Farfetch* já possui: diversas línguas, mercados globais como a China, Japão, Brasil e Rússia, modos de pagamento já verificados, serviço ao cliente, produção e edição de catálogo de produtos e fotografias (Business of Fashion, 2015).

Assim, quando uma marca realiza uma parceria com a unidade de negócio *Black and White*, está a ter acesso a uma facilidade na criação de todo o processo de *e-commerce* para a sua marca, ao mesmo nível do que a *Farfetch* já possui no momento e os mesmos serviços e estruturas de *software*.

Visto que a *Farfetch* já existe na indústria *e-commerce*, é natural que já disponha de serviços, plataformas auxiliares e *software* próprio, devidamente implementado, testado e, consequentemente, funcional. Também a unidade de negócio *Black & White* utiliza e contribui para os mesmos recursos e por isso também os disponibiliza aos seus clientes e marcas de parceiros.

A figura seguinte ilustra um *website e-commerce* realizado em parceria com *Black & White*:



Figura 1 - Diagrama de Componentes de negócio

O utilizador final, ao interagir com o componente *website* (através da interface com o utilizador) consegue utilizar e navegar no mesmo, com a finalidade de usar as suas funcionalidades. Estas funcionalidades assentam na lógica de negócio de *Black & White* e, consequentemente, na lógica de negócio da *Farfetch* – que assentam no componente denominado “*Business Logic*” da figura 1. A lógica de negócio da *Farfetch* (componente “*Business Logic*”) é constituída por serviços que se conectam que constituem funcionalidades importantes para manipular e interpretar os dados.

### 1.1.3 Acessibilidade

Acessibilidade significa “permitir o acesso a um lugar ou a um conjunto de informações, permitindo que não só pessoas com incapacidades ou com mobilidade reduzida, participem em atividades que incluem o uso de informação, mas também permite a inclusão dessas facilidades e acessos a todas as pessoas presentes numa determinada população, visando assim a adaptação e melhoria de sistemas, eliminando os obstáculos que são colocados no que diz respeito à acessibilidade” (MDN Web Docs, 2017).

Na área tecnológica, a acessibilidade traduz-se em “ter acesso a todo o material áudio, vídeo e digital, adaptando todos os meios que a tecnologia permite” (Debian Accessibility, 2016). A informação digital (*websites*, jornais, entre outros) que é considerada acessível permite às pessoas incapacitadas utilizarem os recursos e informação existentes. A informação digital pode ser acessível através de ferramentas que ajudam a entender e a manipular a informação digital, tais como:

- Leitores de ecrã (*screen-reader*) para minimizar a incapacidade visual (cf. secção 2.4.1);
- Teclados virtuais<sup>1</sup>, para minimizar a incapacidade motora ou falta de coordenação motora;
- Aparelhos de reconhecimento de fala, para minimizar incapacidade motora ou visual.

A acessibilidade é extremamente importante para os negócios baseados em *websites*, pois um dos seus objetivos é a utilização por um público o mais amplo possível. Este público inclui não só as pessoas comuns, mas também as pessoas com incapacidades. Assim, a acessibilidade na *web* visa permitir que “todas as pessoas possam consultar e interagir com a informação digital presente na *web*, usando qualquer dispositivo que tenham” (Keijonen, 2017).

O facto de a acessibilidade estar cada vez mais presente é uma grande ajuda para os utilizadores na compreensão e melhoria de informação, podendo assim garantir que o conteúdo está acessível a mais pessoas, dando a possibilidade de estes desfrutarem corretamente de um *website* conseguindo colmatar as deficiências ou incapacidades temporárias.

De acordo com o *The World Bank* (Schenker, 2016), 15% das pessoas sofrem de algum tipo de incapacidade ou necessidade especial, como por exemplo: pessoa invisual, surda, com paralisia cerebral, utilizadores com deficiências<sup>2</sup> motoras, com uma deficiência temporária (braço fraturado, cansaço à noite, entre outros) ou com deficiências tecnológicas (computador antigo, rato avariado, entre outros) (cf. secção 2.2.3).

Assim sendo, é necessário que as aplicações/sites *web* sejam cada vez mais acessíveis para que as pessoas com incapacidades consigam realizar as suas tarefas/atividades diárias de

---

<sup>1</sup> *Software* que permite a entrada e escrita de texto em dispositivos como forma alternativa ao teclado convencional.

<sup>2</sup> Incapacidades

forma mais prática e confortável. Para que isto seja possível, é necessário que surjam, cada vez mais, softwares/*websites* mais acessíveis.

Acessibilidade na *Web* é a prática de construir *websites* que possam ser utilizados por todas as pessoas, quer tenham algum tipo de incapacidade quer não, permitindo o acesso à informação de forma universal a todos os utilizadores, independentemente das suas necessidades especiais. É também caracterizada por facilitar o acesso e o uso de ambientes e serviços a qualquer pessoa, em diferentes contextos.

Posto isto, quando os conteúdos ou serviços de uma organização são disponibilizados de forma acessível, eles estão ao alcance de uma ampla gama de utilizadores, não só para os que estão limitados, mas também para os que possuem qualquer tipo de limitação temporária.

## 1.2 Problema

O processo de elicitação de requisitos de “*customer experience*” dos *websites e-commerce* restringiu a sua visão a quem são os seus clientes/utilizadores, negligenciando o princípio da inclusão (cf. secção 2.1.5), e excluindo, mesmo que sem intenção, aqueles que poderiam vir a ser os seus potenciais clientes – as pessoas com incapacidades físicas/cognitivas, que se sentem mais confortáveis a comprar os seus produtos no conforto da sua casa.

O software da *Black & White* tem assim diversas limitações ao nível da acessibilidade (cf. secção 3.3.1 e secção 3.3.2) o que leva ao não cumprimento de algumas regras e padrões impostos por lei (cf. secção 2.1.7) em alguns países e à perda de clientes potenciais.

## 1.3 Objetivos

O objetivo deste projeto é o desenvolvimento duma solução que permita aos *websites* desenvolvidos e fornecidos pela equipa *Black & White* o cumprimento das leis e normas de acessibilidade definidas em diversos países (cf. secção 2.1.7) e dessa forma aumentar o universo dos clientes e utilizadores dos *websites*, contribuindo assim para uma sociedade mais inclusiva.

Assim sendo, os sub-objetivos deste projeto são os seguintes:

- Compreender as principais causas relacionadas com acessibilidade (cf. secção 3.3.1) que fazem com que os utilizadores abandonem os *websites* de *e-commerce*;
- Desenvolver um protótipo/piloto *website e-commerce* que solucione as principais causas previamente identificadas, permitindo desta forma o acesso a um tipo de pessoas mais diversificado, com diferentes tipos de incapacidades, de acordo com as leis e normas de acessibilidade (cf. secção 2.1.7);
- Desenvolver software que potencia a adoção das soluções desenvolvidas em protótipo, através da sua reutilização e, conseqüentemente, dos seus componentes gráficos/lógicos;

- Providenciar um protótipo piloto que tenha como objetivo principal o suporte do máximo das funcionalidades do mesmo, em vários *browsers*, em versão *desktop*.

## 1.4 Abordagem

O projeto proposto pode ser observado através da perspectiva de três partes interessadas (*stakeholders*) relevantes:

- Utilizador, que pretende ter acesso a um *website* que cumpre certos requisitos de acessibilidade, conseguindo usufruir de uma navegação acessível e fácil. Com acessibilidade nos *websites*, o utilizador usufrui também de um *website* mais cuidado e pormenorizado a nível de estruturação e semântica de código (semântica HTML) o que traz muitas vantagens para os *screen-readers* e tecnologias assistivas (TA);
- Empresa/Marca, que pretende perceber em que medida é vantajoso a relação custo-benefício do desenvolvimento de mecanismos de acessibilidade. O projeto em causa permite à organização um melhor conhecimento dos seus clientes e utilizadores e, o mais importante, permite que os serviços e conteúdos sejam acessíveis a mais pessoas, com restrições e limitações, mas às custas do desenvolvimento.
- Equipa de desenvolvimento, que pretende organizar e reutilizar código e algoritmos, de forma a que se consiga ter especial atenção à acessibilidade, conseguindo atingir os requisitos de acessibilidade de forma determinística, sistemática e comum entre todos.

Para tal, serão realizadas diversas tarefas que permitem a concretização das expectativas das partes interessadas:

- Interpretar o problema de acessibilidade de acordo com as necessidades da empresa;
- Efetuar uma vasta pesquisa de técnicas de acessibilidade e tecnologias assistivas, levando à escolha de elementos e boas práticas a ser implementadas;
- Explorar tecnologias assistivas que possibilitam a interpretação/leitura dos *websites* em casos de limitações específicas;
- Criar e refazer componentes existentes, com boas práticas de acessibilidade;
- Entrada numa lista de produtos através da navegação apenas com teclado (*mouseless*)<sup>3</sup>;
- Entrada no detalhe de um produto apenas com o teclado;
- Adicionar ao carrinho de compras apenas com teclado;

---

<sup>3</sup> Navegar em sistemas computacionais sem auxílio do rato, um periférico de entrada.

- Exploração de opções de acessibilidade (*screen-reader, mouseless*);
- Desenvolvimento de projeto com possibilidade de melhoramentos e reutilização de componentes;
- Estudo da possibilidade de configurar definições de visualização do *website* (tamanho e tipo de letra, entre outros).

Assim, deve ser realizado um estudo do estado da acessibilidade nas aplicações existentes na equipa do qual deve resultar um levantamento dos principais problemas a nível de acessibilidade, que serão categorizados por níveis e princípios, como ditam as respetivas diretrizes (cf. secção 2.3.2).

Depois da categorização realizada, é expectável que seja possível resolver algumas lacunas, quer a nível de componentes, quer a nível de projetos, com o objetivo de aumentar e melhorar os níveis de acessibilidade cobertos e explorados pela equipa. A exploração de acessibilidade promove a reutilização de código na equipa através da criação e uso de estrutura de projeto e componentes partilhados, tornando-se numa mais-valia, no sentido em que todos os projetos que herdaram um determinado componente acessível ficam com esta dimensão/requisito atingido, em determinada área do mesmo.

## 1.5 Planeamento do projeto

O projeto foi planeado e delineado de acordo com reuniões de acompanhamento realizadas com os *stakeholders* e o orientador da empresa e conforme as dúvidas que foram surgindo, quer ao longo do desenvolvimento como no decorrer do cumprimento das tarefas.

Assim, estabeleceu-se um processo de desenvolvimento incremental e iterativo, em que para cada tarefa/fase alcançada com sucesso era realizado o controlo e revisão da mesma através de testes de sistema e de análise do fluxo da mesma. Com a revisão final de cada tarefa, foi possível minimizar erros que possam decorrer de fases anteriores e conseguir ter um fluxo de aplicação correto, com a ajuda de processos *GIT* e com recurso a quadros de trabalho *Trello* e *JIRA*, com metodologia *SCRUM*.

Neste processo de controlo das funcionalidades e tarefas, o orientador e os respetivos *stakeholders* davam o seu feedback e opinião e, caso necessário, procedia-se às correções/melhorias sugeridas com o mesmo processo e logo de seguida eram definidas novas tarefas, prioridades e requisitos das mesmas.

Para além deste processo se ter demonstrado vantajoso, foi recursivo e constante até ao final do projeto, assentando em quatro pilares fundamentais que a área de gestão e controlo possui: Planear, Executar, Controlar/Verificar e Agir/Ajustar.

Ao longo do projeto o planeamento sofreu algumas alterações e, seguindo as boas práticas de *SCRUM*, o mesmo foi ajustado, pois foram sentidas algumas dificuldades na compreensão do negócio/projeto, conceitos e arquitetura a utilizar.

A realização do projeto foi assim dividida em três grandes fases:

- 1ª Fase: Planeamento e Investigação. Esta primeira fase incidiu sobre a familiarização e conhecimento do que já é existente no mercado acerca do tema acessibilidade. Foi realizada a análise de todos os pontos fortes e fracos do projeto, bem como todo o meio ambiente contextual inerente ao negócio. Em conjunto, foram estudadas as melhores ferramentas e metodologias a utilizar no desenvolvimento do projeto e foi realizada uma análise às aplicações *web* existentes, com especial atenção no que diz respeito à acessibilidade, resultando assim as necessidades de maior prioridade a implementar. Depois de saber quais as necessidades a implementar e quais os níveis de acessibilidade a atingir foi necessário começar a pesquisar e a perceber como era possível atingir os mesmos parâmetros.
- 2ª Fase: Análise, Implementação, Testes e Controlo. Nesta segunda fase, foi realizada uma análise dos requisitos e prioridades definidos anteriormente e, com base nisso, foram definidos alguns diagramas de arquitetura e componentes, que serão descritos na secção de análise e *design* (cf. secção 4), com o intuito de melhor compreender e documentar o que era pretendido com o projeto. Foram ainda especificados, com mais rigor, todos os requisitos e todo o caminho a realizar ao longo de cada tarefa para se chegar ao objetivo final. Cada requisito foi implementado consoante o plano definido para o mesmo, resultando do mesmo algumas sugestões e melhorias sugeridas pela autora que possam beneficiar o projeto. Ainda nesta fase, foi realizada uma demonstração de resultados ao orientador e supervisor (testes e controlo), procedendo-se à correção de erros encontrados, à implementação de melhorias sugeridas e à demonstração de resultados finais aos *stakeholders* para uma recolha final de nova opinião e *feedback*.
- 3ª Fase: *Outputs* e resultados. Por último, e não menos importante, esta fase incidiu na elaboração de documentos de suporte ao projeto: melhorar documentação no código, documentação de suporte, melhoria de diagramas de componentes e arquitetura, elaboração de inquéritos sobre a satisfação do utilizador, análise das respostas aos inquéritos, elaboração de estatísticas e recolha da opinião do utilizador face aos projetos que foram mudados com o *input* da autora, bem como a realização do presente documento. Esta fase decorreu em simultâneo com a anterior, uma vez que à medida que o projeto avançava, eram registados todos os passos dados e objetivos alcançados com sucesso.

## 1.6 Estrutura do documento

No primeiro capítulo é descrito todo o âmbito em que o projeto está envolvido e é feita uma breve apresentação da organização *Farfetch* e da equipa *Black & White*. É no primeiro capítulo que é descrito o problema, objetivos e abordagem do projeto.

No segundo capítulo é realizado um enquadramento teórico e tecnológico, que permite um melhor entendimento de conceitos-chave e ajuda a aferir o conhecimento necessário para a leitura dos capítulos seguintes. É ainda no segundo capítulo que é realizada uma análise do mercado no que diz respeito a aplicações existentes de tecnologias assistivas, como o *screen-reader* e é feita uma análise de *websites* acessíveis e de ferramentas de auditoria de acessibilidade existentes.

No terceiro capítulo é descrita a proposta de valor do projeto, descrevendo a identificação da oportunidade, bem como análise da mesma e descrição das ideias geradas. É neste capítulo que é usado o método de avaliação hierárquico para conseguir determinar qual a ideia mais vantajosa a implementar e é discriminado o modelo de Canvas (cf. secção 3.7) para melhor entender o negócio.

O quarto capítulo é direcionado para o design arquitetural do software, sendo também apresentadas algumas situações com detalhe, recorrendo ao modelo de representação 4+1 e a diagramas em notação UML. É neste capítulo que se dão a conhecer os requisitos funcionais e não funcionais da aplicação.

No quinto capítulo sintetiza-se o trabalho de implementação do sistema descrito no capítulo 4, descrevendo boas práticas adotadas e detalhes de implementação. Este capítulo descreve o desenvolvimento de situações gerais e termina com a forma como foram implementadas situações particulares explicando o processo de contribuição da aplicação. No seguimento do capítulo quinto, surge o capítulo sexto que descreve a maneira como foram abordados e executados os testes à aplicação e apresenta os resultados dos inquéritos de satisfação dos utilizadores face à aplicação implementada.

No último capítulo é feita uma síntese de todo o trabalho onde são descritos todos os objetivos atingidos, conclusões obtidas com a implementação, limitações encontradas e trabalho futuro.

## 2 Estado de arte

Este capítulo contextualiza o âmbito de desenvolvimento deste projeto, enquadrando-o num contexto mais global. Esta contextualização é feita através de um estudo de análise de mercado e do que é a acessibilidade em vários pontos-chave, conseguindo assim situar o projeto nas áreas de negócio envolventes e existentes.

Este capítulo está dividido em três grandes partes: o enquadramento teórico, o enquadramento tecnológico e os fatores que influenciam a acessibilidade. Estas partes estão relacionadas com conceitos teóricos sobre o tópico da acessibilidade.

### 2.1 Enquadramento teórico

Neste capítulo serão abordados conceitos teóricos importantes e adjacentes ao tema do projeto e que são necessários para a elaboração e entendimento do mesmo.

#### 2.1.1 Interação humano-computador

Com o avanço e inovação da tecnologia, cada vez mais o ser humano se vê obrigado a acompanhar esta inovação e, conseqüentemente a interagir e evoluir com todos os conteúdos que estão disponíveis em formato digital (Shneiderman, 2016) .

As informações a que o utilizador quer aceder, estão cada vez mais disponíveis e armazenadas em formato digital (Shneiderman, 2016), o que traz vantagens no que diz respeito à flexibilidade e facilidade na recuperação de conteúdos, mas também apresenta desvantagens no sentido de minimizar as relações interpessoais e de disponibilizar mais distrações, uma vez que toda a informação está concentrada num único dispositivo que possui várias funções. Este armazenamento e consumo/manipulação de informação só é possível com o uso intermediário de máquinas, sendo a mais conhecida, o computador e as suas derivações, como o *tablet* e o telemóvel (Shneiderman, 2016).

Esta interação entre o humano e o computador traduz-se no facto de a comunicação ocorrer entre as pessoas e os sistemas interativos, através de uma interface. As interfaces, por sua vez

devem ser úteis (eficazes) e de fácil utilização (eficientes), a fim de permitirem a fácil aprendizagem e utilização, tornando o processo intuitivo (Wodehouse, 2017).

A cada dia que passa e com o avanço tecnológico, surgiu uma necessidade maior e crescente de existir interfaces de fácil utilização, pois estão cada vez mais dependentes da tecnologia e, consequência disto e da concorrência, mais exigentes com a qualidade da mesma.

Algumas características que tornam a interação humano-computador mais rica, são a usabilidade e acessibilidade do sistema, que são duas propriedades que vão ser descritas com mais pormenor nas secções seguintes. A par disto, é importante considerar também a satisfação subjetiva do utilizador final com o objetivo de conseguir satisfazer todas as suas necessidades e conseguir que o mesmo aprecie a interface, quer a nível de facilidade de utilização, quer a nível *layout*.

Assim, quando se desenvolve um *software* que vai ser utilizado por várias pessoas não é boa prática ficar concentrado apenas na funcionalidade (sob o ponto de vista do sistema), mas também deve existir uma concentração de esforços na sua usabilidade, acessibilidade e *layout* (para garantir a satisfação do utilizador), pois é o meio pelo qual o sistema é mostrado a todos os utilizadores e é a forma pela qual a informação é entendida e analisada como produto final.

Em suma, é necessário ter em conta todos o tipo de utilizadores e desenvolver *software*/aplicações com interfaces que possam ser adaptadas para todo o tipo de clientes, ou seja, de boa usabilidade, aprendizagem e acessibilidade, tendo como finalidade atingir a satisfação do utilizador.

### **2.1.2 Tecnologias assistivas**

“Tecnologia assistiva é um termo recente que é utilizado para identificar todos os recursos e serviços que contribuem para melhorar as habilidades e capacidades funcionais de pessoas com deficiência e, conseqüentemente, promover a vida independente e inclusão. Pode também ser definida como uma ampla gama de equipamentos, estratégias, serviços e práticas estabelecidas e aplicadas para minimizar os problemas encontrados pelos indivíduos com deficiências” (Cook, A. M.; Hussey, 2002).

O termo tecnologia assistiva (*Assistive Technology*), traduzido, foi criado em 1988 e é um elemento jurídico importante dentro da legislação norte-americana conhecida como Lei pública 100-407 e foi renovado em 1998 como *Assistive Technology Act*. Em conjunto com outras leis, como é o caso da lei ADA – *Americans with Disabilities Act* (cf. secção 2.1.8), ajuda a “regular os direitos dos cidadãos com deficiência nos Estados Unidos da América (EUA), além de prover a base legal dos fundos públicos para compra dos recursos que estes necessitam” (Sartoretto, Mara; Bersch, 2017).

Segundo Tonolli e Bershe (Tonolli, José Carlos; Bersche, 2006), a tecnologia assistiva possui várias categorias para diversas áreas que podem ser alvo de adaptações:

- Auxílios para a vida quotidiana;
- CAA ou CSA – Comunicação Aumentativa (Suplementar) e alternativa;

- Recursos de acessibilidade no computador;
- Sistemas de controlo do ambiente;
- Projetos de arquitetura para acessibilidade;
- Órteses e próteses;
- Adaptação de postura;
- Auxílios de mobilidade;
- Auxílios para cegos ou com défice de visão;
- Auxílios para surdos ou com défice auditivo;
- Adaptações em veículos.

Em suma, a tecnologia assistiva está presente em várias áreas da sociedade com o intuito de proporcionar às pessoas com deficiência ou alguma dependência, qualidade de vida e inclusão social, através da ampliação da sua comunicação, mobilidade e integração com a sociedade.

Neste projeto, a categoria de tecnologia assistiva que está mais presente é a de **Recursos de acessibilidade no computador**, que trata dos equipamentos de entrada e saída (síntese de voz, rato e teclado), alternativas de acesso, teclados modificados, softwares especiais (de reconhecimento de fala, *screen-reader*, entre outros), que permitem que as pessoas com deficiência ou algum tipo de incapacidade consigam usar o computador e outros dispositivos semelhantes (Tonolli, José Carlos; Bersche, 2006). Neste projeto, este conceito vai ser estendido para o conceito de acessibilidade na área de *e-commerce*, mais precisamente, na acessibilidade em *websites e-commerce*.

### 2.1.3 Acessibilidade e Usabilidade

Os termos acessibilidade e usabilidade são normalmente citados em conjunto, pois o seu fim e objetivo é partilhado pelos dois: o *website/aplicação/serviço* tem de ser fácil de usar, intuitivo e acessível a todas as pessoas, que possuam incapacidades ou não, através do cumprimento das diretrizes associadas aos dois temas (Keijonen, 2017). Na maioria das situações, como é o exemplo de desenvolver *websites* e aplicações, é mais fácil e produtivo abordar estes dois temas em conjunto, para benefício do produto final.

Todos os critérios de usabilidade estão diretamente relacionados e alinhados com os ideais de acessibilidade e tecnologias assistivas, como sugerido por Sami Keijonen (Keijonen, 2017):

- *Design* intuitivo. “Uma aplicação possui um *design* intuitivo quando os utilizadores entendem a estrutura do site e a sua navegação, de forma intuitiva, ou seja, a estrutura é percebida por intuição” (Keijonen, 2017).

- Facilidade de aprendizagem. “Um *website* é considerado de fácil aprendizagem quando os utilizadores compreendem rapidamente qual é o seu foco principal e completam/fazem as principais tarefas do *website*” (Keijonen, 2017).
- Eficiência de uso. “O uso de uma aplicação é considerado eficiente quando o utilizador consegue realizar as tarefas que deseja, até serem concluídas, de forma rápida” (Keijonen, 2017).
- Memória de uso. “É considerado que uma aplicação tem boa memória quando o utilizador consegue facilmente voltar às mesmas tarefas de forma simples e com toda a informação anterior disponível” (Keijonen, 2017).
- Frequência de erros e sua gravidade. “Nas aplicações deve ser analisado se os utilizadores cometem erros na sua navegação e se os mesmos são graves ou não. Com esta classificação dos erros é sempre possível definir prioridades e saber quais são os erros que a aplicação de deve focar primeiro” (Keijonen, 2017).
- Satisfação subjetiva do utilizador. “As características de usabilidade referidas acima estão perfeitamente alinhadas com os princípios e ideais de acessibilidade” (Keijonen, 2017).

Mesmo que acessibilidade se concentre em pessoas com incapacidades, muitos requisitos deste tema melhoram a usabilidade para os utilizadores. A acessibilidade não só beneficia pessoas com incapacidades como pessoas que possuem algum tipo de limitação, nem que seja temporária, como estar a utilizar o computador sob luz solar intensa, estar em ambientes escuros ou barulhentos, entre outros.

Existem vários requisitos em acessibilidade:

- Requisitos mais específicos para pessoas com incapacidades - garantir que os *websites* funcionam bem com tecnologias assistivas, como leitores de ecrã, aplicações para ampliar o conteúdo das páginas da *web* e *software* de reconhecimento de fala. A maioria destes requisitos é técnico e está relacionada ao código subjacente e não à aparência visual;
- Requisitos que também são princípios gerais de usabilidade - que estão incluídos nos requisitos de acessibilidade porque podem ser barreiras significativas para pessoas com incapacidades. Por exemplo, um *website* desenvolvido para que possa ser usado sem o periférico rato é um caso de boa usabilidade; e usar um *website* sem rato é um requisito de acessibilidade porque pessoas com algumas incapacidades físicas ou visuais não o conseguem usar.

Normalmente, quando se define os requisitos de acessibilidade de um projeto, o foco está em aspetos que coloquem as pessoas com incapacidades no mesmo ponto de vantagem que pessoas sem incapacidades. Por outro lado, quando se define os requisitos de usabilidade e experiência do utilizador, estes aspetos sobrepõem-se de forma significativa aos de acessibilidade porque existe uma preocupação acrescida com utilizadores específicos, que possuem incapacidades, incluindo várias considerações sobre acessibilidade, como tecnologias assistivas.

Assim, ao definir os requisitos de um projeto podem ser incorporadas técnicas de usabilidade para melhorar a acessibilidade e, podem ser incorporadas técnicas de acessibilidade para que os projetos funcionem melhor e para mais pessoas em diversas situações, promovendo a usabilidade.

Com isto, a acessibilidade na *web* é um recurso importante porque fornece uma exploração mais aprofundada da importância e dos benefícios da mesma como uma disciplina distinta que continua a concentrar esforços em pessoas com incapacidades, incentivando o aumento da comunicação e coordenação entre a acessibilidade e usabilidade, conseguindo promover a prática na concepção e desenvolvimento de diretrizes, *websites*, *browsers*, tecnologias assistivas e outras ferramentas da *web*.

Portanto, quando existe uma preocupação com a implementação de acessibilidade e com o cumprimento das suas diretrizes, naturalmente estão a ser tidos em conta alguns dos princípios da usabilidade, e vice-versa. Assim, abordar a acessibilidade e a usabilidade em conjunto pode levar, de uma forma mais eficaz, a uma *web* mais acessível.

#### **2.1.4 E-commerce**

Antigamente, o termo *e-commerce* costumava ser um conceito simples: “ser uma extensão de lojas físicas que proporcionam aos internautas e a oportunidade de comprarem bens e serviços a partir do conforto das suas casas” (Moore, Lindsay; Mon, Gonzalo; Skelton, 2017).

À medida que a popularidade do *e-commerce* aumentou, começaram a surgir discussões sobre o *design* dos *websites e-commerce* como “*customer experiences*” (Moore, Lindsay; Mon, Gonzalo; Skelton, 2017). Ou seja, começou a existir uma preocupação crescente sobre a forma como o *website* aparece ao utilizador e sobre como este reagiria ao interagir com o mesmo. Ainda assim, não foi só a preocupação com o *User Experience (UX)* que aumentou a necessidade de um melhor *e-commerce*, mas também se começou a ter mais atenção a uma melhor *Developer Experience (DX)* - melhorar a implementação dos *websites* para melhor corresponderem ao seu propósito. Quanto mais rica e imersiva for a UX e a DX, melhor (Moore, Lindsay; Mon, Gonzalo; Skelton, 2017), pois permite a criação de novas experiências digitais que captam a atenção do consumidor/utilizador, tornando-se assim numa estratégia de o levar a entrar de forma mais eficiente no processo de compra (*checkout*) e consequente transação.

#### **2.1.5 Princípios da Inclusão**

Em 2009, foi formado um grupo para desenvolver uma estrutura base que conseguisse dar resposta às diversas necessidades dos utilizadores de um campus universitário, nomeadamente da Universidade de Waterloo (University of Waterloo).

Com este grupo e o seu estudo sobre as necessidades, “os Princípios da inclusão foram criados para reconhecer, comunicar e promover a compreensão da complexidade e singularidade da

comunidade do campus, apresentando os ideais pelos quais os membros do campus se devem reger: respeito, sensibilidade e equidade” (University of Waterloo).

Estes princípios falam no local de trabalho de Waterloo e apoiam a estratégia de internacionalização, promovendo um melhor sistema para todos os alunos, funcionários e professores (University of Waterloo).

Os princípios da inclusão, que se deve ter em mente quando se fala de acessibilidade na *web*, são os seguintes (University of Waterloo):

- Reconhecer que existem pessoas que têm necessidades únicas e particulares tanto no ambiente de aprendizagem, como no trabalho;
- Respeitar o direito de cada pessoa se expressar em relação às suas crenças, religião, cultura, origem étnica, orientação sexual, habilidade física e mental, entre outros;
- Promover a inclusão de todos os indivíduos, ajustando os procedimentos, atividades e ambientes;
- Acreditar na capacidade de uma pessoa, não ligando a pressupostos;
- Ser inclusivo em todas as formas de comunicação, promovendo a integridade da população;
- Atuar no quotidiano com sensibilidade, respeito e justiça.

Todos estes princípios podem ser adaptados na acessibilidade na *web*. Também a acessibilidade na *web* se deve reger com respeito, sensibilidade e equidade no que diz respeito a todos os utilizadores, quer possuam incapacidades ou não, promovendo assim uma melhor experiência nos momentos em que usam aplicações *web*.

### **2.1.6 Acessibilidade é para todos**

Segundo o *World Bank*, “mil milhões de pessoas, ou 15% da população mundial, tem algum tipo de deficiência/incapacidade e a prevalência da incapacidade é maior nos países em desenvolvimento. Um quinto do total global estimado da população, ou seja, entre 110 milhões e 190 milhões de pessoas, possui deficiências significativas” (The World Bank, 2017).

“As pessoas com incapacidade estão mais propensas a resultados socioeconómicos adversos do que as pessoas sem deficiência/incapacidade, com menos educação, menos resultados na saúde, menos oportunidades de emprego e maiores taxas de pobreza. A própria economia dos países, a legislação e o ambiente social pode criar ou manter barreiras para com a participação de pessoas com incapacidades na vida em comunidade, cívica e económica. Estas barreiras podem incluir edifícios inacessíveis, falta de transporte acessível, menos acesso às tecnologias de informação e menor nível de serviços acessíveis e financiamento desses serviços” (The World Bank, 2017).

“A pobreza pode aumentar o risco de incapacidade através da desnutrição, acesso inadequado à educação e aos cuidados de saúde, condições de trabalho inseguras, ambiente

poluído e falta de acesso à água potável e saneamento. Por outro lado, a incapacidade pode aumentar o risco de pobreza, por falta de emprego e oportunidades de educação, salários mais baixos e aumento do custo de vida com a deficiência/incapacidade” (The World Bank, 2017).

A acessibilidade deve beneficiar pessoas com e sem incapacidades - apesar do foco principal estar nas pessoas com deficiências, a acessibilidade também beneficia outras pessoas, incluindo (WC3, 2012):

- Idosos;
- Pessoas com pouca literacia ou não fluentes em determinado idioma;
- Pessoas com tecnologias antigas (computadores antigos) ou com uma conexão fraca de *Internet*;
- Utilizadores novos ou inexperientes;
- Utilizadores com dispositivos *mobile*;
- Utilizadores com dislexia;
- Pessoas daltónicas;
- Pessoas com pouca visão.

Em suma, todas as pessoas, com algum tipo de deficiência ou incapacidade temporária podem usufruir da acessibilidade.

### **2.1.7 Leis e normas de acessibilidade**

A acessibilidade é exigida por lei em muitos países. Na União Europeia (UE) existem diretrizes<sup>4</sup> para tornar mais acessíveis os *websites* e aplicações móveis de órgãos no setor público. Estas diretrizes foram aprovadas no dia 22 de dezembro de 2016 e abordam especificamente os *websites* do setor público (Keijonen, 2017). No entanto, todas as diretrizes e boas práticas de acessibilidade podem e devem ser incluídas em diversas áreas e em todo o tipo de *websites* (Keijonen, 2017), como é o caso de *websites e-commerce*.

Na Europa, as leis da acessibilidade abordam quatro pontos principais que explicam como construir serviços digitais e *websites* mais acessíveis (Keijonen, 2017):

- Perceível: As informações e componentes da interface com o utilizador devem ser apresentadas aos utilizadores de forma a que estes percebam.
- Operável: Os componentes da interface com o utilizador e a navegação na mesma devem ser operacionais, ou seja, de fácil manuseamento.

---

<sup>4</sup> *Guidelines* no termo original.

- **Compreensível:** A informação e as operações na interface com o utilizador devem ser de fácil compreensão.
- **Robusto:** O conteúdo do *website* deve ser suficientemente robusto para que possa ser interpretado de forma confiável por uma ampla variedade de utilizadores, incluindo o uso de tecnologias assistivas.

As regras mencionadas acima estão de acordo com as Diretrizes de Acessibilidade para o conteúdo na *web* (WCAG – *Web Content Accessibility Guidelines*) (cf. secção 2.3.2).

A convenção das Nações Unidas sobre os direitos das pessoas com deficiências/incapacidades é retificada de forma iterativa e contínua ao longo dos anos e promove o acesso das pessoas com deficiência/incapacidade às novas comunicações e informações, especialmente na *Internet* (Stevens, 2016).

Nos Estados Unidos, as leis de acessibilidade são referidas na secção 508, da Lei da Reabilitação de 1973. A secção 508 rege a acessibilidade das tecnologias de informação, no governo. As pessoas com deficiência receberam a oportunidade de aceder a novas informações desde 1998, quando a Lei de Reabilitação de 1973 foi alterada, exigindo que as agências federais disponibilizassem a sua informação tecnológica para pessoas com deficiência (Stevens, 2016). A Secção 508 desenvolveu padrões de acessibilidade que podem ser incorporados em qualquer tipo de produto que envolva as tecnologias da informação, bem como dar padrões de acessibilidade que podem ser usados por todo tipo de produtos e empresas.

A Secção 508 dispõe de uma lista de verificação (*Section 508 Checklist*) que é uma lista de padrões a seguir em HTML. Com esta lista foi possível fornecer as diretrizes do que era ideal e definido por lei (Stevens, 2016).

### **2.1.8 Americans with Disabilities Act (ADA)**

Segundo o *U.S. Census Bureau*, praticamente uma em cada cinco pessoas dos Estados Unidos da América (EUA) tem algum tipo de deficiência/incapacidade, o que representa uma fatia significativa da população (Census Bureau, 2012).

As autoridades reguladoras, com o intuito de fornecer proteção e adaptações aos americanos incapacitados, criaram uma lei, a ADA (Americans with Disabilities Act), que foi assinada e decretada a 26 de Julho de 1990, pelo presidente George H. W. Bush (U.S. Department of Justice, 2017).

“A ADA traduz-se numa das leis de direitos civis mais abrangentes dos Estados Unidos da América (EUA), que proíbe a discriminação e garante que as pessoas com deficiência/incapacidade tenham as mesmas oportunidades que todos os outros seres humanos e assim conseguirem participar ativamente na vida corrente americana. Esta lei defende que todas as pessoas incapacitadas devem conseguir ter oportunidades de emprego, comprar bens e serviços e participar nos programas do estado ou locais. Em suma, a ADA é uma lei de igualdade de oportunidades para pessoas com deficiência/incapacidade” (U.S. Department of Justice, 2017).

Para ser abrangido por esta lei, “é necessário ter uma deficiência/incapacidade, que é definida pela ADA como física ou mental, que limita substancialmente uma ou mais atividades importantes da vida quotidiana, ser uma pessoa que tenha histórico ou registo de tais danos/prejuízos; ou ser uma pessoa que é percebida por outros como tendo essa deficiência/défice. No entanto, a ADA não menciona especificamente todas as deficiências que estão cobertas” (U.S. Department of Justice, 2017).

## 2.2 Fatores de acessibilidade

O termo acessibilidade pode ser definido de uma forma simples, que descreve os requisitos contextuais nos quais deve existir principal foco: "Um *website* é acessível quando alguém, independentemente das circunstâncias económicas, geográficas ou físicas, consegue aceder e navegar no mesmo."

Isto significa que podem existir fatores económicos, geográficos e físicos que podem influenciar a acessibilidade, e os mesmos vão ser descritos nesta secção. Embora sejam apresentados todos os fatores que afetam a acessibilidade, para este projeto em particular, os fatores a serem considerados são os fatores físicos/diversidade de utilizadores (cf. secção 2.2.3).

### 2.2.1 Fatores económicos

Os fatores económicos da acessibilidade existem de duas formas:

- Conexão da internet: Depende do contexto económico e da frequência de utilização por parte do cliente/utilizador;
- Tempo gasto na pesquisa de informação: Depende maioritariamente do tempo disponível do cliente/utilizador.

Ambos os fatores podem estar relacionados e podem ter impacto direto na acessibilidade, pois o cenário económico pode ser medido pelo tipo de conexões que as pessoas têm. Indiscutivelmente, as pessoas com elevados padrões económicos escolhem conexões de banda larga (fibra ótica por exemplo), enquanto que as pessoas com baixo rendimento optam por ligações mais limitadas, no contexto de um país em desenvolvimento.

O tipo de conexão à internet determina a velocidade de navegação, o que influencia diretamente o tempo que um utilizador passa a navegar. Quanto mais lenta a conexão, mais o utilizador se vai sentir desencorajado, e vice-versa.

No entanto, o *SEO*<sup>5</sup> lida com este problema até certo ponto - o grau de filtragem e a sequência das listagens feitas pelos motores de busca irão influenciar a acessibilidade (Sambhanthan & Good, 2012).

### 2.2.2 Fatores geográficos

O aspeto geográfico é outro grande fator que determina a acessibilidade - aceder a um *website* a partir de um meio rural com fraca conexão é um desafio. Em várias zonas do interior do país a tecnologia 4G, por exemplo, não está disponível ou não tem alcance suficiente, o que vai limitar o uso da internet por parte do consumidor (Sambhanthan & Good, 2012). Embora este fator também esteja relacionado com os fatores económicos, também pode existir devido à localização geográfica dos utilizadores.

Outro aspeto onde a geografia tem um tremendo impacto é em casos como a China ou a Coreia do Norte, que bloqueiam deliberadamente inúmeros conteúdos e serviços.

### 2.2.3 Diversidade de utilizadores

Existem inúmeras razões pelas quais as pessoas podem ter diferentes graus de deficiências, sejam elas visuais, auditivas, cognitivas, físicas ou até de fala. Por exemplo, podem advir de deficiências de nascença, doenças, acidente ou podem desenvolver deficiências com a idade.

Embora existam diferentes definições e interpretações literárias para o termo “deficiência” (W3C, 2008), Nielsen interpreta este termo como "qualquer dificuldade em interagir com um sistema informático" (Nielsen, 2000). Esta interpretação vai ao encontro do objetivo deste projeto e ao foco da acessibilidade nos *websites* de *e-commerce*.

#### 2.2.3.1 Visão

Segundo (Nielsen, 2000), as deficiências visuais são as que causam problemas de acessibilidade mais sérios.

Estas variam desde perda de visão leve ou moderada num ou em ambos os olhos (baixa visão) até perda de visão substancial e não corrigível em ambos os olhos (cegueira). Além disto, algumas pessoas têm sensibilidade a certas cores reduzida ou mesmo inexistente (daltonismo), ou maior sensibilidade a cores brilhantes. Estas variações na perceção de cores e brilho podem ser independentes da acuidade visual (WAI-W3C, 2012).

Tipicamente, as pessoas com deficiências visuais tendem a mudar a apresentação de conteúdo na *web* para melhor servir as suas necessidades específicas, tais como (WAI-W3C, 2012):

---

<sup>5</sup> *Search Engine Optimization* - Maneira de aumentar os acessos de um *website* através de um conjunto de técnicas e estratégias que permitem que o mesmo melhore o seu posicionamento nos resultados orgânicos em mecanismos de pesquisa.

- Aumentar/reduzir o tamanho do texto e imagens;
- Personalizar o tipo de letra, cores e espaçamento;
- Ouvir descrições áudio (para vídeos).

Para estes e outros métodos funcionarem, o *software* (e em particular neste projeto, o do *website*) tem de garantir que a apresentação de conteúdo seja independente da estrutura subjacente, e que esta estrutura esteja devidamente codificada para ser apresentada de formas diferentes pelos vários *browsers* e tecnologias assistivas. Um *design* acessível suporta diferentes apresentações do conteúdo da *web* e diferentes formas de interação (WAI-W3C, 2012).

### 2.2.3.2 Audição

“As deficiências auditivas variam desde perda auditiva leve ou moderada num ou ambos os ouvidos (dificuldade de audição) até perda de audição substancial e não corrigível em ambos os ouvidos (surdez). Algumas pessoas com deficiência auditiva podem ouvir sons, mas às vezes não são suficientes para entender toda a fala, especialmente quando há ruído de fundo” (WAI-W3C, 2012).

“Embora os conteúdos multimédia ofereçam muitas oportunidades para pessoas com este tipo de deficiência, também colocam desafios quando o conteúdo não é preparado para ser acessível. O caso do vídeo, por exemplo, pode ser utilizado para comunicar informações visualmente, mas o conteúdo áudio precisa de alternativas, como transcrições e/ou legendas” (WAI-W3C, 2012).

Tipicamente, as pessoas com deficiências auditivas dependem de (WAI-W3C, 2012):

- Transcrições e legendas de conteúdo de áudio;
- Reprodutores de vídeo que exibem legendas e oferecem opções para ajustar o tamanho do texto e as cores das legendas;
- Opções para parar, pausar e ajustar o volume do conteúdo áudio (independentemente do volume do sistema);
- Áudio em primeiro plano de alta qualidade que é claramente distinguível de qualquer ruído de fundo.

Para algumas pessoas com deficiência auditiva, a linguagem gestual é o idioma principal, e podem não conseguir entender, de forma integral, a linguagem escrita corrente (WAI-W3C, 2012). Fornecer informações importantes em linguagem gestual e usar texto mais simples que é complementado por imagens, gráficos e outras ilustrações são formas que ajudam a transformar o conteúdo da *web* mais compreensível para muitas pessoas (WAI-W3C, 2012). No entanto, é importante lembrar que nem todas as pessoas com deficiência auditiva conhecem a linguagem gestual (WAI-W3C, 2012).

### 2.2.3.3 Cognição

As deficiências cognitivas (dificuldades de aprendizagem e neurológicas), envolvem neuro diversidade e distúrbios neurológicos, bem como distúrbios comportamentais e de saúde mental que não são necessariamente neurológicos. Podem afetar qualquer parte do sistema nervoso e afetar o quão bem as pessoas ouvem, movem, veem, falam e entendem informações. De notar que este tipo de deficiências não afeta necessariamente a inteligência de uma pessoa” (WAI-W3C, 2012).

Dependendo das necessidades individuais, tipicamente as pessoas com deficiências cognitivas, de aprendizagem e neurológicas dependem de (WAI-W3C, 2012):

- Conteúdo claramente estruturado que facilita visão geral e orientação;
- Etiquetagem consistente de formulários, botões e outras partes de conteúdo;
- Hiperligações com destinos previsíveis;
- Diferentes formas de navegar em *websites*, tais como menus hierárquicos e possibilidade de pesquisa;
- Opções para suprimir conteúdos intermitentes ou de cariz distrativo;
- Texto complementado por imagens, gráficos e outras ilustrações.

### 2.2.3.4 Físico

As deficiências físicas (ou incapacidades motoras) incluem fraqueza e limitações do controlo muscular (como movimentos involuntários, tremores, falta de coordenação ou paralisia), limitações da sensação, distúrbios das articulações (como artrite), dor que impede o movimento, e membros em falta.

Tipicamente, pessoas com deficiência física usam *hardware* e *software* especializados, tais como:

- Teclado/rato ergonómico ou com *design* especial;
- Apontadores de cabeça, bocais ou outras ajudas para a digitação;
- Teclado no ecrã, *joysticks* ou outros dispositivos apontadores;
- Interruptores operados pelo pé, ombro ou outros movimentos;
- Reconhecimento de fala, rastreamento ocular ou outras abordagens para interação de mãos livres.

De notar que pessoas com alguma deficiência podem precisar de mais tempo para digitar, clicar<sup>6</sup> ou realizar outra interação, da mesma forma que podem digitar sequências de teclas em vez de pressionar teclas em simultâneo para ativar determinados comandos.

## 2.3 Enquadramento tecnológico

Nesta secção serão abordados conceitos tecnológicos importantes e adjacentes ao tema do projeto e que são necessários para a elaboração e entendimento do mesmo.

### 2.3.1 Acessibilidade em *e-commerce*

O significado e a importância dos padrões de acessibilidade na *web* e no comércio *online* é conhecido. Muitos investigadores do conceito, investigam a acessibilidade no comércio *online* e recomendam que os desenvolvedores de software de comércio *online* devem considerar a acessibilidade também no *design* (Sohaib & Kang, 2016). "Como as organizações públicas e as empresas privadas contam com mais tecnologias baseadas na *web* para as compras *online*, estas empresas devem implementar estratégias para garantir que todos os utilizadores possam integralmente aceder ao conteúdo na *web*" e propõe um modelo de acessibilidade na *web* para beneficiar todas as organizações públicas e empresas privadas (Sohaib & Kang, 2016).

Segundo Sohaib e Kang (Sohaib & Kang, 2016), os *websites* de comércio *online* perdem até 50% de potenciais vendas porque os utilizadores não conseguem encontrar a informação, ou aceder a ela com facilidade. Um *website* acessível é aquele que fornece uma experiência satisfatória para os utilizadores finais, aumentando as vendas e as receitas para as marcas, no caso de serem *websites* de *e-commerce* acessíveis (Sohaib & Kang, 2016).

De acordo com Dolson (Dolson, 2009), "as deficiências físicas dos visitantes de um *website* são um fator que se deve considerar". Dolson acredita que "é possível obter mais clientes fiéis e evitar os desafios legais com o bom uso da acessibilidade. Se o consumidor consegue ter acesso a informações relevantes, então a sua confiabilidade no *website* aumenta e, conseqüentemente, aumenta a sua intenção de compra" (Sohaib & Kang, 2016).

Faulkner, em colaboração com a *Vision Australia*, o *Paciello Group* e o consórcio de ferramentas de acessibilidade na *web* (Sohaib & Kang, 2016), desenvolveu uma barra de ferramentas de acessibilidade da *Web* (WAT – *Web Accessibility Toolbar*), para auxiliar na avaliação de um *website* e se o mesmo está em conformidade com as diretrizes WCAG (cf. secção 2.3.2) (Sohaib & Kang, 2016). Faulkner "também reconhece que a informação nos *websites* de comércio *online* varia em qualidade, na medida em que se o consumidor conseguir entender que o *website* apresenta informações de qualidade e fácil tangibilidade, este fator irá gerar confiança no mesmo" (Sohaib & Kang, 2016).

---

<sup>6</sup> Escolher uma opção ou desencadear uma ação através de um botão do rato ou teclado.

Assim sendo, cada vez mais, existe uma necessidade de implementação de acessibilidade na *web* e do uso de critérios/ferramentas de medição de qualidade da mesma para o conteúdo do comércio *online*.

### 2.3.2 Diretrizes da acessibilidade

As diretrizes de acessibilidade do conteúdo *web* (conhecidas como WCAG – *Web Content Accessibility Guidelines*) são um conjunto, internacionalmente reconhecido, de recomendações para melhorar a acessibilidade da *web*. Estas diretrizes explicam como tornar os serviços digitais acessíveis a todos (W3C, 2018), incluindo utilizadores com algumas deficiências ou incapacidades, tal como explicadas anteriormente na secção de fatores de acessibilidade (cf. secção 2.2). Ao seguir estas diretrizes é possível tornar o conteúdo da *web* mais acessível para os utilizadores, no geral (W3C, 2018).

Qualquer aplicação deve seguir estas diretrizes de acessibilidade, pois ao fazê-lo há uma certificação que as diretrizes para cumprir acessibilidade estão a ser satisfeitas e o trabalho irá coincidir com as mesmas e irá contribuir para que a *web* seja um local mais acessível para o utilizador.

As diretrizes WCAG, são possuem versões e atualmente encontram-se na versão 2.1, sendo assim conhecidas como WCAG 2.1. Estas diretrizes estão organizadas em quatro (4) princípios: **Perceptíveis, operáveis, compreensíveis e robustos**. Para cada diretriz, existem critérios de teste que são classificados em três níveis, para classificar o sucesso da mesma:

- **Nível A** – Principiante;
- **Nível AA** – Intermédio;
- **Nível AAA** – Avançado.

Os critérios de sucesso são escritos como passos de testes que não são específicos, nem contêm linguagem técnica, nem tecnológica. A satisfação dos vários critérios de sucesso leva a que uma determinada diretriz (*guideline*) possa ser atingida no nível A, AA ou AAA.

#### 2.3.2.1 Princípio 1 – Perceptível

Para garantir que o *website* possui o Princípio 1 é necessário que os utilizadores reconheçam e usem o serviço de forma intuitiva. Para isso, existem algumas diretrizes (W3C, 2018):

- D1.1: Textos alternativos: Usar textos alternativos (*alt text*) para conteúdos não textuais de modo a que este possa ser mudado e compreendido de outras formas que os utilizadores precisem (*zoom*, *braille*, *screen-reader*, linguagem mais simples, entre outros);
- D1.2: Alternativas para conteúdo áudio e vídeo: Usar transcrições, legendas e alternativas que ajudem a entender o conteúdo;

- D1.3: Adaptável: Criar conteúdo que possa ser apresentado de diferentes formas, como ter dois tipos de *layout* - o mais estruturado e o mais simples, sem perder informações ou estrutura;
- D1.4: Distinguível: Facilitar aos utilizadores a visão e audição do conteúdo de forma simples, incluindo a separação distinta entre o primeiro plano e o plano de fundo através do uso de cor, contraste, controlo de áudio, redimensionar o texto, não permitir áudio de fundo, entre outros;

#### 2.3.2.2 Princípio 2 – Operável

Para garantir que o *website* possui o Princípio 2 é necessário que o utilizador consiga encontrar o conteúdo que deseja, independentemente da forma como o procura (teclado ou comandos de voz). Para isto existem algumas diretrizes (W3C, 2018):

- D2.1: Acesso com o teclado: Certificar que todas as funcionalidades estão disponíveis com o uso do teclado;
- D2.2: Tempo suficiente: Fornecer aos utilizadores tempo suficiente para ler e usar e manipular o conteúdo;
- D2.3: Convulsões: Não usar conteúdo a piscar/tremer ou intermitente, que possa causar convulsões ou ataques;
- D2.4: Navegável: Fornecer formas de ajudar os utilizadores a navegar na aplicação, encontrar conteúdo e determinar onde eles estão na aplicação;
- D2.5: Acessibilidade para apontadores: Permitir que os utilizadores operem os apontadores de forma fácil, para conteúdo que consiga interpretar os mesmos;
- D2.6: Periféricos de entrada adicionais: Garantir que estes periféricos são suportados e não são um obstáculo para os utilizadores.

#### 2.3.2.3 Princípio 3 – Compreensível

Para garantir que o *website* possui o Princípio 3 – Compreensível é necessário que os utilizadores consigam entender o conteúdo do *website* e como funciona o serviço. Para isto existem algumas diretrizes (W3C, 2018):

- D3.1: Legível: Criar conteúdo de texto legível e compreensível, usando linguagem simples e mantendo as frases curtas, ou explicando palavras e/ou expressões que os utilizadores possam não conhecer;
- D3.2: Previsível: Criar conteúdo que opera e funciona de forma previsível e consistente, dando a entender ao utilizador o estado de algo quando o mesmo está a ser modificado (estado *focus*), providenciado etiquetas que identificam o contexto;
- D3.3: Assistência de entrada: Ajudar os utilizadores a evitar e a corrigir os erros, através da identificação dos erros e da sua possível resolução, bem com fornecer instruções de preenchimento de formulários.

#### 2.3.2.4 Princípio 4 – Robusto

Para garantir que o *website* possui o Princípio 4 – Robusto é necessário que o conteúdo possa ser interpretado da mesma forma por uma grande variedade de utilizadores. Para isto existem algumas diretrizes (W3C, 2018):

- D4.1: Compatível: Garantir a compatibilidade com tecnologias assistivas, através do uso de HTML válido e possível de interpretar, garantindo que as mesmas conseguem identificar qual é o propósito de cada funcionalidade, em que estado está e que valores possui.

### 2.3.3 API de acessibilidade

Antes de abordar qualquer assunto sobre implementação de acessibilidade de acordo com o uso de tecnologias assistivas é necessário entender que estas tecnologias, tal como os leitores de ecrã, são ferramentas separadas que funcionam em conjunto com os *browsers*. Assim, as ferramentas apresentadas na secção 2.4.1 não fazem parte do *browser* nem comunicam diretamente com este. Pelo facto de não estarem integrados com os *browsers*, os leitores de tela precisam de um mecanismo para obter e decifrar o conteúdo do *browser*, para este ser corretamente identificado pelos utilizadores de tecnologias assistivas.

Os leitores de tela usam a API de acessibilidade (também conhecida como *accessibility API*) para identificar e decifrar o conteúdo de um *browser*. Uma API é uma interface que estabelece uma conexão e possibilita obter dados necessários para uma melhor compreensão de certa aplicação, e é isso que acontece com a API de acessibilidade e com os leitores de ecrã. O sistema operativo/*browser* expõe as informações e eventos para as tecnologias assistivas conseguirem interpretar, por meio desta API. A API de acessibilidade é o ponto de conexão entre o *browser* e os leitores de ecrã. Na realidade, tanto os utilizadores de tecnologias assistivas, como os desenvolvedores não precisam de interagir diretamente com esta API, apenas precisam de entender os seus mecanismos e funcionalidades para tirar o maior proveito da mesma.

A API de acessibilidade não só interpreta as informações presentes nos *browsers* para fornecer os dados da melhor forma aos utilizadores de tecnologias assistivas, mas também interpreta as interações dos utilizadores com os leitores de ecrã com o intuito de perceber qual a informação que os mesmos querem obter e conseguir assim interpretá-la.

Na figura seguinte é possível ver uma ilustração da comunicação entre os *browsers* e as tecnologias assistivas, tendo como conector principal a API de acessibilidade.

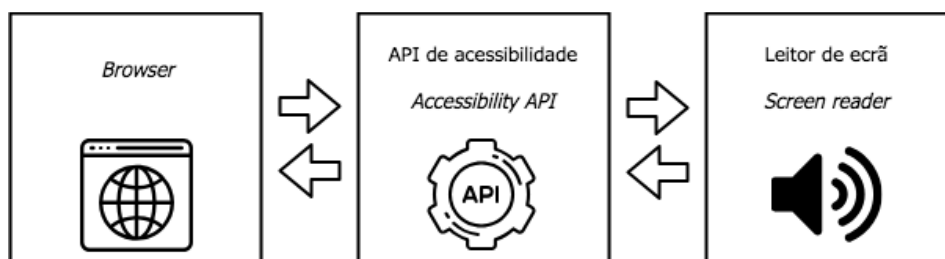


Figura 2 - Comunicação entre os *browsers*, tecnologias assistivas e a API de acessibilidade (Accessible)

### 2.3.4 Árvore de acessibilidade

O *browser* converte os elementos da DOM para aquilo a que chamamos de árvore de acessibilidade (*accessibility tree*), que é uma árvore de elementos HTML que representa a estrutura da interface com o utilizador. Esta árvore, por sua vez, é exposta para as tecnologias assistivas por meio da API de acessibilidade abordada na secção anterior, e possui um fluxo específico:

- A aplicação (*web* ou outra aplicação) expõe uma versão semântica do seu conteúdo e interface com o utilizador para as tecnologias assistivas, por intermédio da API de acessibilidade;
- A tecnologia assistiva usa as informações obtidas através da API para criar uma alternativa da página para o utilizador conseguir interpretar a informação. Por exemplo, um leitor de ecrã cria uma interface na qual o utilizador ouve uma representação falada da aplicação;
- A tecnologia assistiva também pode permitir que o utilizador interaja com a aplicação de uma maneira diferente. Por exemplo, a maioria dos leitores de ecrã fornecem atalhos e dicas para permitir que um utilizador simule facilmente um clique com o rato;
- A tecnologia assistiva transmite de volta a interação do utilizador (como "clique") de volta à aplicação por meio da API de acessibilidade. A aplicação, por sua vez, fica com a responsabilidade de interpretar a ação adequadamente no contexto da interface com o utilizador original e reproduzir o conteúdo e ações desejadas.

Para os *browsers*, existe um passo extra em cada direção, porque o *browser* é, na verdade, uma plataforma para aplicações *web* que são executadas através dele. Portanto, o *browser* precisa de converter a aplicação *web* numa árvore de acessibilidade e deve certificar-se de que os eventos apropriados são disparados, via *JavaScript*, com base nas ações do utilizador provenientes da tecnologia assistiva.

Todo este fluxo é responsabilidade do *browser*, no entanto quem desenvolve páginas *web* deve estar ciente de qual é o fluxo que acontece entre as tecnologias assistivas e as aplicações para conseguir tirar o máximo proveito deste processo e conseguir criar experiências acessíveis para os utilizadores. Basta garantir que a semântica das páginas *web* é a mais correta, conseguindo com que os elementos das páginas possuam o comportamento, estados

e propriedades corretas para serem acessíveis e que tenham nomes e descrições acessíveis. Com estas mudanças e preocupações, os *browsers* permitem que as tecnologias assistivas consigam aceder a estas informações para criar experiências personalizadas.

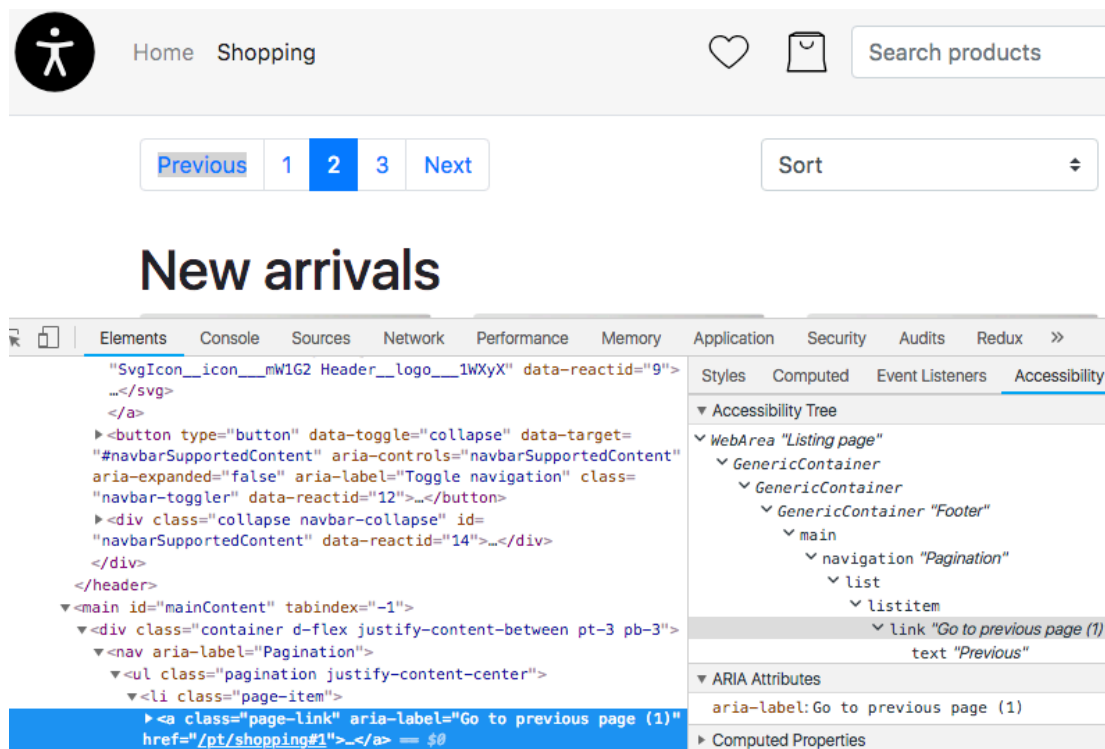


Figura 3 – Árvore de acessibilidade do projeto

A imagem acima apresenta a árvore de acessibilidade do presente projeto. A árvore de acessibilidade é possível ser vista com recurso às ferramentas providenciadas pelo *Chrome DevTools*, na *tab* de acessibilidade.

## 2.4 Análise de soluções existentes

Esta secção apresenta uma análise das soluções existentes no mercado no que diz respeito a ferramentas e tecnologias assistivas, nomeadamente *screen-readers*, ferramentas de auditoria e *websites* com as boas práticas de acessibilidade implementadas.

Esta contextualização serve em muito para situar o projeto nas áreas de negócio envolventes e existentes, bem como onde e como devem ser adotadas as tecnologias assistivas para delas retirar vantagens. No que diz respeito aos *websites* analisados, irá ser revisto quais as boas práticas que cumprem e vantagens que demonstram para o utilizador.

### 2.4.1 Screen-reader

Um *screen-reader* é uma tecnologia assistiva, que é essencial para as pessoas cegas, para pessoas com incapacidades visuais, analfabetos ou pessoas com dificuldade na aprendizagem

de novos conceitos. Este meio de tecnologia assistiva tenta transmitir o que as pessoas com visão normal veem no ecrã para os utilizadores invisuais, através de narração do conteúdo da página ou dispositivos *Braille* (L. Watson, 2005).

De forma geral, um *screen-reader* permite que pessoas com incapacidades visuais graves usem um computador. Sem ser no contexto de uma página *web*, um *screen-reader* também colabora com o sistema operativo (SO) do computador para fornecer informações sobre ícones, menus, caixas de diálogo, ficheiros e pastas. O dispositivo/computador fornece acesso total ao sistema, incluindo às aplicações (L. Watson, 2005).

Esta tecnologia assistiva transmite informações/*feedback* aos utilizadores de duas formas:

- Narração;
- Braille.

O mecanismo *Text-To-Speech (TTS)* que os *screen-readers* usam, para traduzir as informações, pode ser uma aplicação de *software*, que vem incorporada no próprio *screen-reader*, ou pode ser um dispositivo externo (*hardware*). No caso de ser *hardware*, pode ser um dispositivo externo para fornecer informações em *Braille*. À medida que a informação na interface do computador muda, a informação *Braille* do dispositivo também muda, conseguindo fornecer assim informações atualizadas diretamente do computador.

Como a maioria dos utilizadores desta tecnologia assistiva não usa rato, todos os *screen-readers* usam uma grande variedade de atalhos de teclado para realizar algumas tarefas. Estas tarefas incluem a leitura parcial ou total de um ficheiro, navegar em páginas *web*, abrir e fechar ficheiros ou ouvir música.

Um utilizador com incapacidade visual usa uma combinação de atalhos do *screen-reader* e do sistema operativo para conseguir realizar muitas das tarefas que um computador consegue executar. Hoje em dia, todos os sistemas operativos atuais possuem os seus próprios atalhos de teclado, que estão disponíveis para todos e não apenas para utilizadores de tecnologias assistivas.

Estes tipos de tecnologias estão disponíveis para os sistemas operativos mais comuns, como *Linux*, *Mac OS* e *Windows* e suportam uma vasta gama de aplicações das plataformas nativas.

Cada *screen-reader*, possui um idioma predefinido, que corresponde ao idioma do sistema operativo. Além disso, os *screen-readers* são capazes de lidar com diferentes idiomas dentro dos documentos, ou seja, se um texto em uma página *web* estiver em francês, o mesmo vai ser interpretado com a alteração de acentos, tom e entoação para imitar o estilo da língua francesa falada. A maioria deste tipo de leitores suporta idiomas comuns, como o inglês, francês, espanhol, italiano e alemão.

Quando as páginas *web* estão construídas com um código bem estruturado, os *screen-readers* conseguem interagir com elas com alguma facilidade. As páginas *web* bem estruturadas devem incluir cabeçalhos, listas, parágrafos, bem como tabelas que incluam informações relevantes sobre o seu conteúdo, imagens que tenham uma descrição de texto alternativa e *links* que possuam um texto informativo preciso e claro. Todas estas informações adicionais devem ser realizadas com recurso ao idioma do computador em que a página *web* está a

aparecer. A razão pela qual estes elementos devem estar presentes no código é porque um *screen-reader* irá executar o código da página e garantir que determinados atalhos para os elementos apresentados possam estar disponíveis.

De uma forma geral, todos os *screen-readers* usam uma variedade de atalhos para navegar entre o conteúdo disponível, como por exemplo: atalhos para navegar entre listas, títulos, *links*, imagens e quase todo o tipo de elementos que seja possível encontrar em páginas *web* (L. Watson, 2005).

#### 2.4.1.1 Job Access With Speech - JAWS

JAWS é o *screen-reader* para o sistema operativo Windows mais popular do mundo - 46.6% dos participantes de um inquérito usam-no como *screen-reader* principal (WebAIM, 2017), desenvolvido para utilizadores invisuais ou cuja perda de visão os impossibilita de ver o ecrã e/ou navegar com o rato. Este software disponibiliza saída de voz e de *braille* para um infinito número de aplicações de computador (Freedom Scientific, 2018).

#### 2.4.1.2 NonVisual Desktop Access - NVDA

NVDA é um *screen-reader* gratuito, para o sistema operativo Windows, que permite a utilizadores cegos ou com deficiências visuais usarem o computador e é utilizado por cerca de 31,9% da população respondeu a um inquérito de acessibilidade (WebAIM, 2017). Este *software* interpreta o conteúdo no ecrã com uma voz sintetizada, e é possível controlar o que é lido através da colocação do cursor (com o rato ou com o teclado) em determinada área de texto. Também consegue converter texto em *braille*, se o utilizador tiver um dispositivo/periférico de entrada que assim o permita (NV Access, 2018).

#### 2.4.1.3 Apple VoiceOver

É um *screen-reader* incorporado no sistema operativo macOS (bem como nos outros sistemas operativos da Apple - iOS, tvOS, watchOS e iPod) que enuncia o texto e descreve em voz alta o que aparece no ecrã. Este leitor de ecrã é utilizado por cerca de 11,7% de participantes de um inquérito de acessibilidade realizado pela WebAIM (WebAIM, 2017) e possui também suporte para vários dispositivos *braille*. Com o *VoiceOver* é possível controlar o computador através do seu cursor, para percorrer o ecrã, seleccionar botões e outros controlos para ler e editar texto (Apple, 2016).

#### 2.4.1.4 Comparação dos *screen-readers*

Em seguida, na tabela é possível visualizar algumas informações comparativas entre os vários *screen-readers* que foram explicitados nas secções prévias.

Tabela 1 - Comparação de *screen-readers* existentes

Nome	Plataforma	Licença	Criador	Percentagem de utilização
JAWS	Windows	Comercial para	Freedom	46,6%

Nome	Plataforma	Licença	Criador	Percentagem de utilização
		Windows, gratuito para DOS	Scientific	
NVDA	Windows	Livre e open source (GPL2)	NonVisual Desktop Access project	31,9%
Voiceover	MacOS, iOS, tvOS, watchOS, iPod	Livre para produtos Apple	Apple Inc.	11,7%

#### 2.4.2 Ferramentas de auditoria/avaliação

As ferramentas de avaliação de acessibilidade na *web* são programas de *software* ou serviços *online* que ajudam a determinar se o conteúdo das páginas *web* ou aplicações estão de acordo com as diretrizes de acessibilidade com recurso a testes automáticos que são realizados sobre o código HTML da página. Em seguida é apresentada uma lista de ferramentas de auditoria de acessibilidade que podem ser usadas para garantir que a maioria das diretrizes sejam cumpridas:

- *aXe* – analisa o código HTML e procura encontrar potenciais problemas de acessibilidade. Esta ferramenta pode ser utilizada através de extensões de *browsers* (*Firefox / Chrome*) e permitem gerar um relatório de erros de acessibilidade na página em que o utilizador se encontra. É um *script* que pode ser utilizado de diversas formas e possui uma *CLI* que pode ser integrada em *pipelines*, ambientes de teste ou durante a implementação;
- *Google Chrome Audits Panel* – painel com ferramentas de auditoria com base no *Lighthouse*<sup>7</sup>, que inclui vários testes automáticos, destacando-se os testes de acessibilidade e boas práticas. Esta ferramenta retrata os erros encontrados e sugere soluções para os mesmos;
- *Tenon.io* – ferramenta *online* que possui um validador onde é possível inserir o *URL* da aplicação a verificar ou o seu código fonte. Esta ferramenta identifica problemas das diretrizes da secção 508 e da WCAG 2.1 (cf. secção 2.3.2), tendo como única desvantagem o facto de ser paga.
- *WAVE* – esta extensão verifica várias funcionalidades, tais como: contraste de cores, uso de texto alternativo, uso dos atributos ARIA, formulários acessíveis, boa estrutura de HTML, uso de HTML semântico, títulos em falta, conteúdo duplicado, entre outros.

<sup>7</sup> *Lighthouse* é uma ferramenta de auditoria e testes automáticos *open-source* que tem como objetivo melhorar a qualidade das páginas *web* em vários níveis: performance, acessibilidade, entre outros.

Esta ferramenta também fornece dicas sobre qual a melhor forma de corrigir os erros encontrados e referencia as diretrizes de acessibilidade.

Por fim, na secção de avaliação é possível ver alguns dos relatórios resultantes do uso de ferramentas de auditoria de acessibilidade no projeto e alguma explicação de melhorias e possíveis soluções de alguns erros encontrados.

### 2.4.3 Websites acessíveis

Um *website* acessível, no contexto deste projeto, é aquele que fornece uma experiência satisfatória para os utilizadores finais, aumentando as vendas e as receitas para as marcas, no caso de serem *websites* de *e-commerce* acessíveis (Sohaib & Kang, 2016) e que permite o uso de todo o tipo de tecnologias assistivas auxiliares.

Tal como falado anteriormente (cf. secção 2.1.8), os EUA dispõem de uma lei, a lei ADA que define algumas regras e padrões mínimos que devem ser cumpridos para com a acessibilidade dos *websites*. Posto isto, não é de surpreendente ver os *websites* do governo dos EUA, como referência de *websites* acessíveis e que cumprem acessibilidade e que seguem as diretrizes WCAG (cf. secção 2.3.2).

Os melhores *websites* do governo são os *websites* do Centro de Controlo de Doenças (CDC - Centers for Disease Control and Prevention; 2009) e da Administração da Segurança Social (United States Social Security Administration, 2016) e estão ilustrados nas figuras 4 e 5, respetivamente.



Figura 4 - Website acessível: Centro de Controlo de doenças dos Estados Unidos



Figura 5 - Website acessível: Administração da Segurança Social dos Estados Unidos

A acessibilidade implementada nos dois *websites* é muito semelhante. Ambos incluem muitas características das diretrizes de acessibilidades. Algumas características se podem verificar nos *websites* em questão:

- Referências ARIA (ARIA *roles*), em diferentes partes do *website*, incluindo no *header*<sup>8</sup>, navegação, pesquisa e na informação de texto e complementar;
- Boa estrutura e semântica HTML de *headings*<sup>9</sup>;
- Inclusão no topo da página, de *links* escondidos para secções da própria página (*skip links*) que permitem uma navegação fácil para conteúdo específico. Estes *links* ficam visíveis quando o utilizador usa a navegação por teclado, recorrendo ao *tab*;
- Suporte de vários idiomas.

O *website* oficial do governo dos Estados Unidos da América - *usa.gov* (United States of America government, 2008) representado na figura 6, é também um bom exemplo no que diz respeito a ser acessível e possui algumas características interessantes:

- Totalmente controlado com a técnica *mouseless*, ou seja, é totalmente funcional apenas com o uso do teclado. Os utilizadores conseguem navegar no *website* e perceber, de forma clara, onde estão no momento;
- Possui botões de pausa e de controlo para conteúdo que seja apresentado, como *slides* e vídeos.

<sup>8</sup> Região/cabeçalho no topo de cada página *web* que contém informações de todo o *website*.

<sup>9</sup> *Tags HTML* h1 a h6 que definem uma hierarquia de títulos e subtítulos numa página de um *website*.

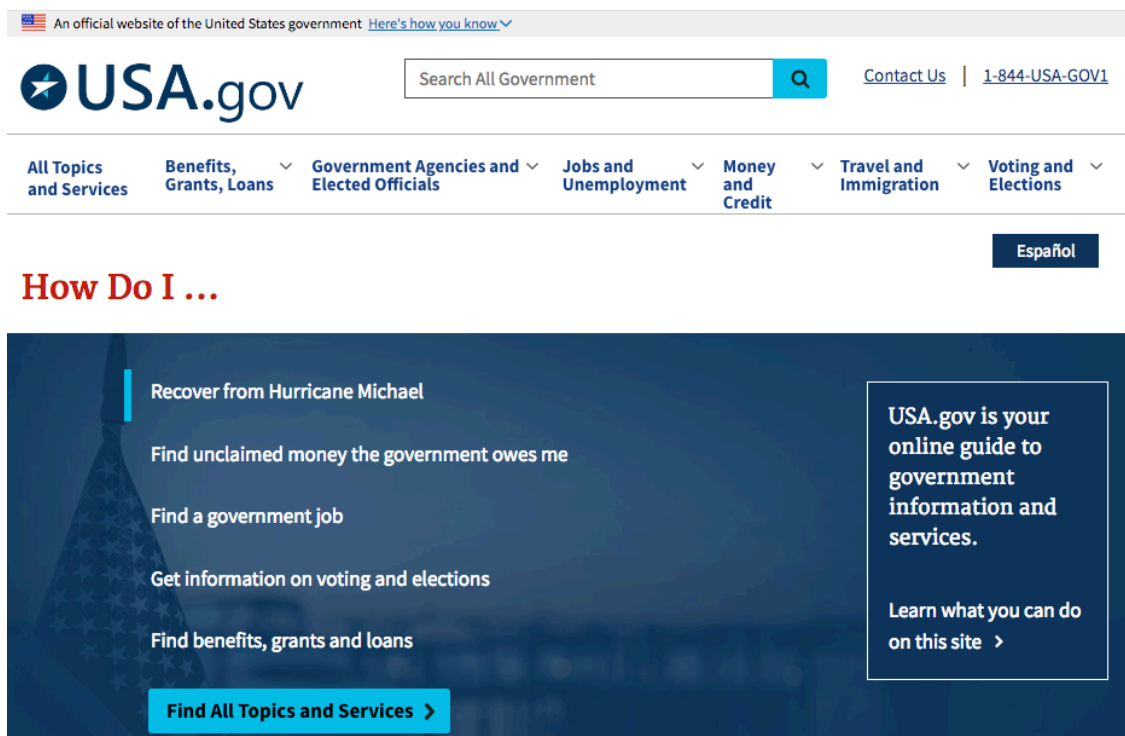


Figura 6 - *Website* acessível: *Website* oficial do governo dos Estados Unidos da América

Estes *websites*, ao longo do projeto vão ser usados como referência e elemento de pesquisa para verificar qual a melhor maneira de implementar determinada funcionalidade no protótipo e respetivos componentes.

## 2.5 Sumário

Todos os conhecimentos adquiridos através do estudo de cada subsecção deste capítulo vão ser tidos em conta durante o desenvolvimento do projeto.

Com a leitura deste capítulo foi possível perceber a importância da acessibilidade e perceber que esta é para todos e não apenas para as pessoas com incapacidades, isto é, todos beneficiam quando um *website* é acessível.

Em suma, a acessibilidade na web requer muitos cuidados na implementação, no entanto traz muitos benefícios para todas as partes intervenientes.

## 3 Análise de valor

Este capítulo apresenta uma análise de valor do projeto, com o intuito de provar o seu valor de mercado, para a organização e para os clientes.

Esta análise fará uso do modelo NCD (*New Concept Development*) para perceber quais são as necessidades dos clientes, bem como as ideias geradas e selecionadas para chegar ao conceito de negócio final. É ainda utilizado o Modelo CANVAS para facilitar a visão geral do negócio e é utilizado o método AHP que permite o uso de critérios qualitativos e quantitativos no processo da avaliação das ideias e desenvolvimento, facilitando a compreensão e avaliação do projeto.

A organização e respetiva equipa onde o projeto se insere realiza *websites* para vários clientes (cf. secção 1.1). Alguns destes clientes são americanos, onde é obrigatório por lei garantir que algumas diretrizes e pontos-chave de acessibilidade são cumpridos nas aplicações. Juntando a necessidade legal à evolução e inovação contextuais, e visto que ainda não existe processo de desenvolvimento guia para acessibilidade, tornou-se prioridade conseguir esclarecer algumas diretrizes de acessibilidade e perceber como se implementam, bem como desenvolver componentes de software reutilizáveis que cumpram as diretrizes de acessibilidade, e, conseqüentemente, a lei americana (ADA). Ainda assim, ao juntar a inovação é também necessário que surjam ideias novas e novos processos de teste e desenvolvimento de aplicações com acessibilidade.

### 3.1 Modelo New Concept Development

O modelo NCD (*New Concept Development*) é um modelo iterativo que fornece uma linguagem e visão comuns e permite esclarecer a visão e cultura de uma organização/projeto. Com este modelo é possível saber quais são os fatores de influência do ambiente externo, que consiste nas capacidades organizacionais, estratégia de negócio e mundo exterior (canais de distribuição, clientes e concorrentes).

Este modelo define cinco elementos de atividade:

- Identificação da oportunidade;

- Análise da oportunidade;
- Geração de ideias;
- Seleção de ideias;
- Desenvolvimento do conceito.

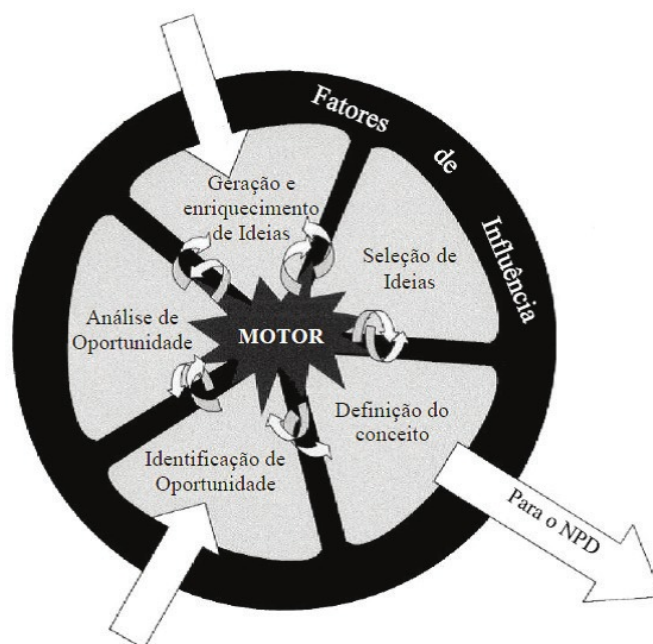


Figura 7 - Modelo NCD (*New Concept Development*)

A figura acima ilustra o modelo NCD e, como se pode observar, possui um formato circular com o intuito de sugerir que as ideias devem fluir e interagir entre os cinco elementos. As setas que apontam para o interior do modelo representam pontos de partida e indicam que os projetos e as suas soluções devem começar na identificação de oportunidades ou geração de ideias e enriquecimento das mesmas. A seta de saída, para o exterior do diagrama, representa que todos os conceitos que resultam do modelo entram no novo processo de desenvolvimento de produto (NPD) (Koen, 2001).

### 3.2 Identificação da oportunidade

Nos últimos anos, as compras *online* cresceram significativamente, devido ao rápido crescimento da tecnologia. A acompanhar esta evolução as empresas continuam a estender as suas funcionalidades, tanto no *design* como na tecnologia, para os seus *websites* direcionados a *Business-to-Consumer (B2C)*. No entanto, face a esta evolução e crescimento é importante também ter em conta e implementar a acessibilidade na *web*, com recurso às diretrizes WCAG (cf. secção 2.3.2).

Um *website de e-commerce* é a forma central de comunicação entre uma marca e os seus consumidores (Sohaib & Kang, 2016). As marcas, no geral, procuram fornecer a melhor

experiência de compra *online* ao seu consumidor, independentemente da sua incapacidade. As características dos *websites* das marcas encorajam ou desencorajam o consumidor a comprar *online*, dependendo da facilidade de chegar à informação que este pretende e da facilidade de navegar na aplicação.

Os *websites* precisam de ser acessíveis a todas as pessoas, inclusive àquelas que possuem algum tipo de deficiência. Estas pessoas têm limitações para ir às compras às lojas físicas, o que se torna inconveniente por causa da desvantagem física. No entanto, as pessoas com deficiência podem adquirir uma sensação de estabilidade emocional se no conforto de sua casa, conseguirem realizar compras *online* de forma fácil e acessível, tal como é garantido por lei (Sohaib & Kang, 2016).

Posto isso, surgiu a ideia de melhorar os processos existentes de acessibilidade tendo em conta o que a organização precisa – *websites* B2C com acessibilidade e conhecimento existente sobre o assunto. É pretendido que esta oportunidade seja aproveitada para conseguir melhorar a acessibilidade de alguns componentes de *software* que pertencem a diversos *websites* (botões, imagens, navegação, entre outros) e das aplicações de *software* da organização em geral.

### **3.2.1 Responsabilidade de garantir acessibilidade**

A acessibilidade na *web* é uma responsabilidade de todos os membros de uma equipa pertencente a um projeto e deve incluir todos os *stakeholders*. A acessibilidade na *web* precisa ser sustentada além da implementação do projeto, isto é, à medida que novos conteúdos e requisitos são desenvolvidos. Com isto, é importante que todas as partes interessadas compreendam os quatro princípios básicos da acessibilidade na *web*: Percetível, Operável, Compreensível e Robusto (cf. secção 2.3.2).

É tentador pensar que a responsabilidade de possuir um *website* acessível pertence aos desenvolvedores e especialistas em acessibilidade. Na realidade esta questão é assim vista em muitas empresas, no entanto, nas empresas de grande dimensão é conveniente que toda a equipa e todos os departamentos estejam envolvidos pois há imensas decisões sobre projetos que podem beneficiar ou prejudicar a acessibilidade. De seguida surgem alguns dos exemplos de diferentes papéis que contribuem para o sucesso dos projetos e que podem interferir com questões de acessibilidade:

- Membros da equipa com cargos séniores
  - Sensibilizar e cultivar a cultura de que acessibilidade faz parte do ciclo de vida dos projetos.
- Project manager / product owner
  - Aquando dos levantamentos dos requisitos, os critérios de sucesso, planeamento do trabalho e datas das entregas devem ser definidos com o pressuposto de que acessibilidade está incluída;

- Deve estar consciente da importância da acessibilidade para conseguir escrever o conteúdo das tarefas e dos critérios de sucesso das mesmas da melhor forma.
- Produção de conteúdos
  - Trabalhar bem as descrições de todos os produtos e imagens dinâmicas;
  - Produzir conteúdo claro e objetivo;
  - Produzir textos para *links* e botões que sejam objetivos e explicitem o resultado final da ação associada;
  - Otimizar o conteúdo com conhecimento de como reagem as tecnologias assistivas ao mesmo.
- Designers
  - Considerar o contraste das cores;
  - Pensar num *layout* simples e objetivo;
  - Considerar o tamanho dos elementos e o espaço que os mesmos ocupam;
  - Devem ser os evangelistas de acessibilidade na equipa, no sentido em que devem defender o seu uso e sensibilizar os restantes elementos para seguirem as sugestões de *design*, garantindo o uso de boas práticas.
- Desenvolvedores
  - Usar *JavaScript* para complementar da melhor forma as funcionalidades de acessibilidade;
  - Usar ferramentas de auditoria e tecnologias assistivas aquando do desenvolvimento;
  - Pensar nas interações e eventos com acessibilidade em mente;
  - Criar componentes que possam ser reutilizáveis.
- Quality assurance tester
  - Testar com as diversas tecnologias assistivas e ferramentas de auditoria;
  - Estar familiarizado com problemas comuns de acessibilidade;
  - Verificar se o projeto está acessível com navegação por teclado e leitor de ecrã.

Em suma, a implementação de acessibilidade num projeto é responsabilidade de todos os membros da equipa associada. Se um dos membros não cumprir com sucesso a sua parte,

torna-se significativamente mais difícil criar e nutrir uma cultura de acessibilidade duradoura dentro de uma organização e de uma equipa.

### 3.3 Análise da oportunidade

Tendo em conta alguns *websites* que já incluem acessibilidade e onde já foi referido que a acessibilidade ajudou a atrair mais clientes e a gerar mais receitas, também a implementação de acessibilidade no contexto deste projeto é uma grande mais-valia à organização, uma vez que fará com que a mesma possua uma base de implementação e trabalho com diretrizes (cf. secção 2.3.2) já implementadas que podem ser utilizadas por qualquer *website*.

Visto que a organização precisa de *websites* com acessibilidade, foi verificado que com uma pesquisa sobre o que é possível melhorar internamente neste campo, tanto a nível de *websites* como a nível de componentes dos mesmos, era possível aumentar a acessibilidade da web e conseguir ter um maior número de utilizadores no *website*, como pessoas com incapacidades diversificadas (cf. secção 2.2.3), tais como incapacidades sensoriais (audição e visão), incapacidades motoras (uso limitado das mãos) e incapacidades cognitivas (linguagem e aprendizagem).

Um *website* acessível pode utilizar todo o tipo de tecnologias assistivas auxiliares, como é o caso de *screen-reader* (cf. secção 2.4.1), teclados adaptados ou aparelhos de reconhecimento de fala. Com a ajuda de estas tecnologias assistivas, a acessibilidade na *web* torna-se um elemento importante no *design* de *websites* de comércio *online*. A combinação da tecnologia e da interação humano-computador é uma parte vital da sociedade moderna que permite alcançar uma gama ampla de consumidores e de benefícios económicos.

As subsecções seguintes, explicam alguns pontos-chave que irão beneficiar com a implementação do projeto.

#### 3.3.1 Causas dos problemas de acessibilidade

Existe uma série de fatores que contribuem para os problemas de acessibilidade. Inicialmente os problemas de acessibilidade eram causados por falta de *design* e de implementação que tivesse acessibilidade como um ponto a cumprir (Sambhathan & Good, 2012). Mais recentemente, já foram identificados outros fatores. A falta de consideração, no *design*, de todos os tipos de utilizadores, com ou sem deficiências, é grave nos dias que correm e as mais recentes pesquisas sugerem que os *designers* não têm em consideração/não conhecem as necessidades de uma vasta variedade de utilizadores.

Embora a acessibilidade nos *websites* esteja a melhorar cada vez mais, os problemas de acessibilidade ainda persistem para utilizadores com deficiência (Leitner & Strauss, 2008). Várias pesquisas concluem o mesmo ponto: a falta de envolvimento e *feedback* por parte do utilizador durante o processo de *design* é dos principais fatores que contribui para os problemas de acessibilidade.

De acordo com Lynch e Horton (Lynch & Horton, 2004), apenas uma parte do problema pode ser atribuída à falta de consideração para com as necessidades dos utilizadores. O problema

real é que os *designers* não conseguem ter em consideração o equipamento utilizado pela maioria das pessoas com deficiências. Muitas vezes, os *designers* trabalham com equipamentos de alta qualidade. No entanto, o utilizador final nem sempre possui os mesmos padrões. Uma diferença entre o tamanho de visor ou o navegador *web* pode ser suficiente para já existir uma diferença significativa na acessibilidade de um *website*. Um *website* até pode funcionar em visores de alta definição ao usar um navegador específico, por outro lado, pode não ter o mesmo comportamento num computador de 15 polegadas, usado por uma pessoa comum (Lynch & Horton, 2004).

Por outro lado, existe também o caso inverso: muitos *designers* gráficos/web não conseguem ter, durante o processo de desenvolvimento, envolvimento com o utilizador final, o que, frequentemente, resulta em problemas para o projeto. É possível criar projetos de *design* gráfico/web sem o envolvimento dos utilizadores, mas não é necessariamente a melhor opção. Não ter o utilizador final envolvido no processo até pode ser rentável e menos demorado a curto prazo, no entanto, pode deixar o trabalho dos *designers* gráficos/web vulnerável a acusações de discriminação.

Para colmatar estes problemas entre *design* e utilizador final, a Comissão de Direitos Humanos e Igualdade de Oportunidade delineou os principais elementos que têm impacto na acessibilidade (Sambhanthan & Good, 2012) e que vão de encontro com as diretrizes de acessibilidade (cf. secção 2.3.2), que incluem:

- Falta de texto alternativo nas imagens;
- Falta de informação complementar para elementos não textuais (imagens, vídeo e áudio), como transcrições e legendas;
- Falta de informação complementar para tabelas e falta de instruções de preenchimento nos formulários;
- Falta de navegação e conteúdos acessíveis;
- Falta de cor e contraste nos primeiros planos e planos de fundo;
- Falta de formatação de texto e legibilidade do mesmo.

Embora esta lista identifique alguns dos mesmos problemas que as diretrizes de acessibilidade da W3C (cf. secção 2.3.2), não é tão completa e não tem em consideração as necessidades de grupos de utilizadores específicos e não fornece o mesmo nível de orientação. Por isso, neste projeto ambas devem ter sido em conta, uma vez que a acima referida segue os mesmos princípios que as diretrizes da W3C.

### **3.3.2 Tipos de erros de acessibilidade**

Hoje em dia, existe uma crescente necessidade em abordar o problema da acessibilidade quer no comércio *online*, quer noutras áreas. Embora os *websites* não sejam projetados com todo o tipo de utilizadores em mente, a seguinte tabela apresenta os níveis de complexidade de alguns tipos de erros de acessibilidade.

É ainda desejável que, no presente projeto, se faça esforços e se consiga resolver a maior parte dos problemas enumerados na seguinte tabela, para garantir que os consumidores com deficiências consigam usufruir de um acesso aos *websites* de forma igual aos outros utilizadores.

Tabela 2 - Nível de complexidade de alguns erros de acessibilidade na *web*

Tipos de erros de acessibilidade	Nível	Facilidade de implementação
Falta de texto alternativo e complementar ( <i>alt text</i> ) em todos os elementos não textuais e estáticos (imagens, vídeos, <i>iframes</i> , conteúdo áudio, entre outros)	A	Fácil
Falta de informação complementar no uso exclusivo de cores	A	Moderado
Falta de etiquetas ( <i>labels</i> ) para campos de formulários	A	Moderado
Falta de estrutura lógica de cabeçalhos (semântica de <i>headings</i> )	A	Moderado
Inclusão de funcionalidades/animações que causem intermitência do ecrã	A	Difícil
Falta de contraste entre cores entre o plano de fundo e o primeiro plano	A	Fácil
Falta de identificação do elemento atual ao navegar por teclado ( <i>focus</i> )	A	Fácil
Falta de títulos descritivos para <i>links</i>	AA	Fácil
Uso de medidas absolutas (pixéis) em vez de medidas relativas (valores em percentagem)	AA	Moderado
Falta de identificação do idioma de texto	AAA	Moderado
Falta de especificação lógica da navegação do conteúdo do <i>website</i> , por teclado ( <i>tab - mouseless</i> )	AAA	Difícil
Falta de atalhos para <i>links</i> frequentes	AAA	Moderado
Falta de agrupamento de <i>links</i> relacionados	AAA	Moderado
Falta de especificação de atributos ARIA apropriados e falta de conteúdo extra de leitura para os <i>screen-readers</i>	AAA	Difícil

### 3.3.3 Vantagens do projeto

O presente projeto traz algumas vantagens, para além das nomeadas acima - perceber as causas dos problemas de acessibilidade e resolvê-los. Este projeto traz ainda mais algumas vantagens que se revelam importantes quer a nível de aumento de conhecimento da equipa, quer a nível de elaborar e implementar código e criar *websites* mais acessíveis. Essas vantagens são as seguintes:

- Aumentar o segmento e clientes, conseguindo fazer com que mais pessoas tenham acesso aos produtos;
- Possuir *websites* que estejam de acordo com as leis americanas, ou seja, *websites ADA Compliance* (cf. secção 2.1.8);
- Aumentar o número de receitas (*revenue*);
- Melhorar a experiência do desenvolvedor (*developer experience*);
- Melhorar a experiência do utilizador (*user experience*);
- Aumentar a reutilização de código;
- Tornar o código mais fácil de manter, devido à sua reutilização;
- Melhorar os resultados de pesquisa nos *browsers* (*SEO*);
- Melhorar a semântica HTML do código;
- Aumentar a lealdade e reputação das marcas associadas aos clientes (*brand loyalty*).

### 3.3.4 Vantagens da acessibilidade

Tornar um *website* mais acessível a todos significa também trazer mais tráfego e visitantes regulares ao site, o que conseqüentemente pode trazer mais compras e o aumento da taxa de conversões<sup>10</sup>, no caso de um *website* de *e-commerce* (Schenker, 2016).

Ao não considerar as pessoas com deficiências relacionadas com os fatores de acessibilidade (cf. secção 2.2) – ou quaisquer doenças incapacitantes que podem surgir nos idosos – a soma de todos pode ser uma quantidade substancial de visitantes e utilizadores de *websites* que não estão preparados com acessibilidade.

Hoje em dia, um *website* com recurso a acessibilidade e a pensar nos utilizadores é uma mais-valia, pois consegue atrair mais potenciais compradores/utilizadores, quer seja com deficiência ou uma incapacidade temporária, conseguindo assim totalizar uma maior taxa de conversão e receitas.

---

<sup>10</sup> Percentagem de utilizadores que efetua uma ação desejada no *website*. No contexto *e-commerce*, a taxa de conversão tipicamente representa de percentagem de visitantes que efetivamente compra.

Uma vez que grande parte dos clientes da equipa onde este projeto está a ser desenvolvido, *Black & White*, são americanos ou atua no Estados Unidos da América (EUA), é vital e importante garantir que os seus *websites* vão estar em conformidade com a lei ADA (cf. secção 2.1.8).

Em suma, com a implementação de acessibilidade surgem as seguintes vantagens:

- a nível de produto/*website*, melhora a quantidade de potenciais compradores e evita qualquer problema legal e de má imprensa;
- ao nível do utilizador, melhora a qualidade de vida das pessoas com incapacidades, conseguindo oferecer a oportunidade de as pessoas fazerem o que mais gostam/precisam (fazer compras, votar, comunicar com outras pessoas, tratar de assuntos bancários, etc) de uma forma cómoda, através do conforto do seu lar;
- ao nível do utilizador, melhora a qualidade de vida e adaptação a novas necessidades/realidades, quer temporárias ou permanentes, que podem surgir ao longo da vida.

### 3.3.5 Desvantagens da acessibilidade

Como foi visto anteriormente, a acessibilidade na *web* traz muitas vantagens quando corretamente pensada e implementada.

No entanto, parece ser tentador e mais fácil concentrar os esforços dos projetos no *design* gráfico/*web*, isto é, em elaborar os mesmos sem acessibilidade, pois esta acrescenta aos projetos tempo de desenvolvimento, complexidade, necessidade de competência necessária e custos associados relacionados com o desenvolvimento e competências. Implementar *websites* acessíveis pode implicar um maior conhecimento (*know how*<sup>11</sup>) quer para desenvolver de forma acessível, como para testar se a acessibilidade está corretamente implementada. Testar e verificar acessibilidade pode ser um processo moroso e complexo, pois implica a aprendizagem do uso de atalhos de teclado (testar *mouseless*) e do uso de *screen-readers* em diversos sistemas operativos.

## 3.4 Geração e enriquecimento de ideias

Quando a necessidade de implementar e melhorar a acessibilidade surgiu, surgiram também algumas ideias que pudessem ser o foco deste projeto.

Assim, surgiram algumas ideias:

- Ideia 1: Ferramenta para gerar o texto alternativo das imagens, interpretando-as;

---

<sup>11</sup> Termo em inglês que significa “saber como”. É o conjunto de conhecimentos práticos (fórmulas, informações tecnológicas, técnicas, procedimentos, entre outros) adquiridos.

- Ideia 2: Ferramentas de definições de utilizador personalizáveis (cor de fundo, tamanho de letra, luminosidade e cores) para ser incorporada nos *websites* da organização, na área de cliente;
- Ideia 3: Componentes/protótipo base com acessibilidade, de forma a que sejam reutilizáveis e partilhados por vários *websites*.

A ideia 3 sugere a criação de um protótipo e componentes base com acessibilidade, de forma a que sejam componentes reutilizáveis e partilhados por vários *websites* e é, das três ideias geradas, a que mais atinge as vantagens descritas (cf. secções 3.3.3 e 3.3.4) e a que transparece uma proposta de valor maior no que diz respeito à relação complexidade algorítmica/benefício.

## 3.5 Seleção de ideias

Para este projeto, e depois de avaliar todas as ideias geradas de acordo com a análise hierárquica (cf. secção 3.5.1), e ver qual espelhava mais vantagens a curto prazo, a ideia selecionada foi a de implementar componentes e um protótipo base com acessibilidade, de forma a que todos os componentes e mesmo o protótipo sejam reutilizáveis e partilhados por várias aplicações.

Neste protótipo devem ser analisadas as principais lacunas existentes em diversas áreas dos *websites e-commerce* da organização (página inicial, navegação, lista de produtos, página de produto, página do carro de compras, página de processo de compra, entre outras) e melhorar as mesmas áreas num protótipo.

Este protótipo deve incluir componentes, que também vão ser alvos de revisão e melhorias a nível de acessibilidade, e deve ser testado sempre tendo em conta a navegação por teclado e o uso de tecnologias assistivas, como o *screen-reader* e perceber se a informação que é lida e interpretada pelo mesmo corresponde ao que o *website* pretende demonstrar.

### 3.5.1 Análise hierárquica (AHP)

Um método de decisão multicritério recorre a técnicas numéricas e matemáticas que auxiliam os decisores a escolher uma opção de um conjunto discreto de alternativas. Este processo é efetuado com base no cruzamento das alternativas com os critérios existentes. Estes métodos de apoio à decisão multicritério permitem a priorização de alternativas em situações de critérios conflituosos, procurando satisfazer as restrições, com objetivos. Assim sendo, estes métodos podem fornecer métodos para o apoio à negociação e/ou decisão em grupo (Buchanan & Gardiner, 2003).

O método de decisão multicritério, que será usado para determinar qual a solução gerada que deve ser selecionada é o método de análise hierárquica (*AHP - Analytic Hierarchy Process*). Este método permite o uso de critérios qualitativos bem como quantitativos no processo de avaliação e tem como ideia principal dividir o problema de decisão em níveis hierárquicos, facilitando, assim, a sua compreensão e avaliação.

Conforme supracitado (cf. secção 3.2), a proposta de valor para este projeto é melhorar os processos existentes de acessibilidade da organização, melhorando os *websites* e seus componentes de software. A implementação e pesquisa de acessibilidade permitirá que mais pessoas consigam navegar e utilizar os *websites*, sem problemas de acessibilidade.

De acordo com as ideias geradas (cf. secção 3.4), foi construída a árvore hierárquica de decisão.

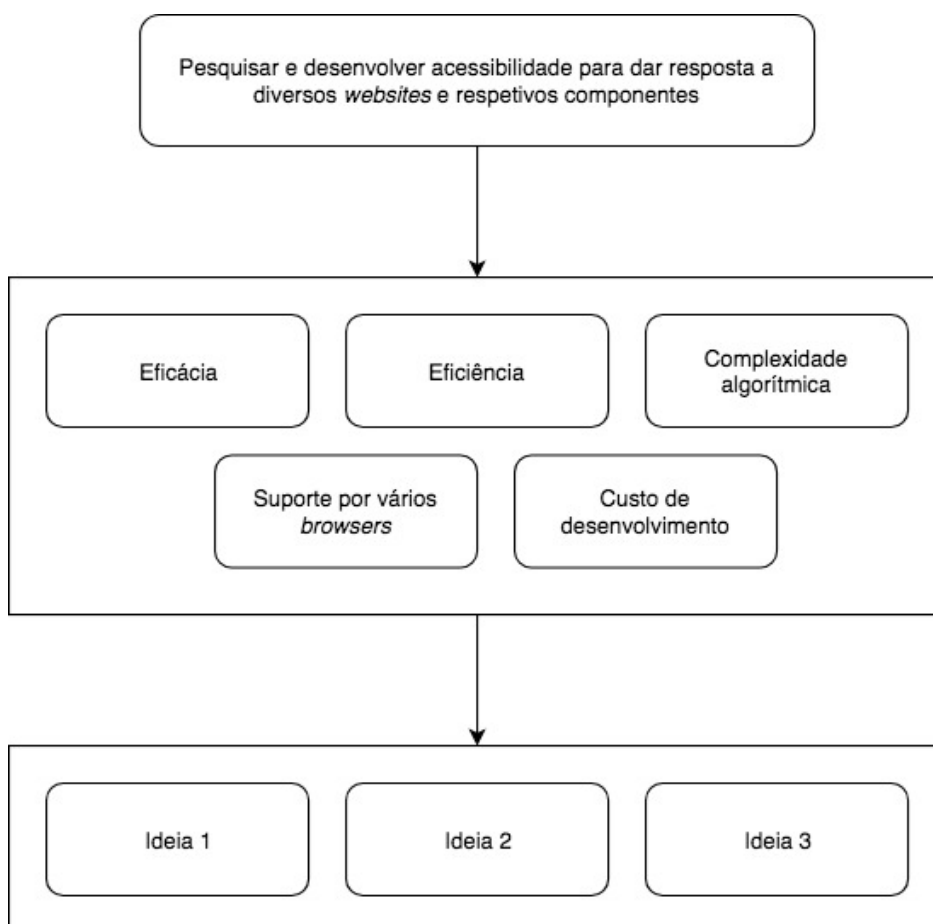


Figura 8 - Árvore hierárquica de decisão

Como observado na figura 8, a primeira camada representa o objetivo principal (pesquisar e desenvolver acessibilidade para dar resposta a diversos *websites* e respetivos componentes). Os critérios envolvidos para avaliar qual a melhor ideia para chegar à solução são:

- Eficácia, se a solução cumpre os princípios de acessibilidade;
- Eficiência, de que forma é que a solução cumpre os princípios de acessibilidade;
- Complexidade (algorítmica), se a solução é fácil ou complexa de desenvolver para atingir o objetivo supracitado;
- Suporte por vários *browsers*, em que medida a solução é suportada por *browsers*;

- Custo de desenvolvimento, relação tempo/recursos necessários para a implementação de determinada solução.

Com recurso a estes critérios é possível avaliar qual é a melhor ideia/solução para atingir o objetivo principal. A próxima tabela inclui a atribuição de pesos para cada critério, segundo a escala de AHP e descreve a matriz normalizada com os pesos finais, para cada critério.

Tabela 3 - Tabela de avaliação AHP

Critérios de avaliação	Eficácia	Complexidade algorítmica	Eficiência	Suporte por vários <i>browsers</i>	Custo de desenvolvimento
Eficácia	1	2	7	8	2
Complexidade algorítmica	1/2	1	7	8	7
Eficiência	1/7	1/7	1	6	3
Suporte por vários <i>browsers</i>	1/8	1/8	1/6	1	5
Custo de desenvolvimento	1/2	1/7	1/3	1/5	1
Total	2 1/4	3 2/5	15 1/2	23 1/5	18

A eficácia e a complexidade algorítmica são os dois critérios que mais importância têm na escolha/seleção da ideia a implementar, uma vez que impactam diretamente os resultados e a implementação que a solução pode ter. A eficiência, suporte por vários *browsers* e custo de desenvolvimento são também critérios que ajudam na seleção, no entanto, não se revelaram tão importantes como a eficácia a complexidade algorítmica.

Tabela 4 - Matriz normalizada do método de avaliação HP

Critérios de avaliação	Eficácia	Complexidade algorítmica	Eficiência	Suporte por vários <i>browsers</i>	Custo de desenvolvimento
Eficácia	0,4409	0,5864	0,4516	0,3448	0,1111
Complexidade algorítmica	0,2205	0,2932	0,4516	0,3448	0,3889
Eficiência	0,0630	0,0419	0,0645	0,2586	0,1667

Critérios de avaliação	Eficácia	Complexidade algorítmica	Eficiência	Suporte por vários <i>browsers</i>	Custo de desenvolvimento
Suporte por vários <i>browsers</i>	0,0551	0,0366	0,0105	0,0431	0,2778
Custo de desenvolvimento	0,2205	0,0419	0,0215	0,0086	0,0556
Soma	1,0000	1,0000	1,0000	1,0000	1,0000

Tabela 5 - Pesos associados aos critérios de avaliação hierárquica

Critérios de avaliação	Pesos
Eficácia	0,483721
Complexidade algorítmica	0,424749
Eficiência	0,14867
Suporte por vários <i>browsers</i>	0,10585
Custo de desenvolvimento	0,08701

De todas as ideias geradas e analisando as tabelas 3, 4 e 5, a que mais eficácia tem e menos complexidade algorítmica possui, numa relação pesquisa/aplicação de conhecimentos, é a ideia 3. Assim sendo, para este projeto, e de acordo com a avaliação hierárquica, a ideia 3 será selecionada para responder às necessidades do negócio.

### 3.6 Definição do conceito

O presente projeto e conceito inerentes devem compreender como é que as pessoas com incapacidade exploram a *web* e os *websites* e conseguir melhorar os mesmos e respetivos padrões de desenvolvimento com base nisso.

Este projeto terá um especial foco na acessibilidade geral, acessibilidade com o uso do teclado (*mouseless*) e a acessibilidade com o auxílio de *screen-readers*. Por exemplo, pessoas com determinada deficiência física só podem usar o teclado ao navegar no site, as pessoas com pouca visão que usam *softwares* de narração precisam de uma melhor legibilidade no *website*.

O objetivo deste projeto é fornecer bases e componentes reutilizáveis, permitindo que sirvam de exemplo para futuros desenvolvimentos. Assim, irá ser desenvolvido um protótipo que possui a navegação no *website* ao usar o teclado e navegar no *website* ao usar o teclado e um software de narração padrão do sistema operativo.

Existe então um conjunto de passos que devem ser totalmente acessíveis:

1. Entrar na homepage do *website* e aceder à navegação com o intuito de selecionar uma categoria;
2. Ao selecionar a categoria, deve abrir uma página de produtos (*listing page*). Nessa página devem ser adicionados filtros de navegação, ordenar os resultados por preço e alterar a disposição da lista;
3. Selecionar um produto da lista de produtos e ser redirecionado para a página de produto;
4. Na página de um produto deve ser possível personalizar o mesmo (quantidade, cor e tamanho) e adicionar o mesmo ao carro de compras (*bag*);
5. Ainda na página de produto, deve ser possível navegar entre as imagens do mesmo para serem lidas mais informações do produto;
6. Adicionar vários produtos ao carro de compras repetindo os procedimentos acima;
7. Na página do carro de compras, deve ser possível remover um produto, mudar a quantidade de outro produto e começar o processo de compra (*checkout*);
8. Deve ser possível entrar na página de compra.

Os requisitos discriminados acima são os requisitos mínimos do projeto, pois cada fase deve ser dotada de acessibilidade, quer a nível de legibilidade para o *screen-reader* quer a nível de navegação através do teclado.

Em cada um dos requisitos deve ser alvo de revisão a lista de verificação da acessibilidade da W3C e serem implementados os padrões recomendados.

### 3.7 Modelo de Canvas

O modelo Canvas ou *Business Model Canvas*, é uma ferramenta de planeamento estratégico que permite desenvolver um modelo de negócio novo ou melhorar um existente, de forma a conhecer todas as partes intervenientes do processo.

Este modelo possui 9 partes que ajudam a conhecer melhor o negócio:

- Parcerias principais, que incluem as atividades-chave realizadas por terceiros e os recursos principais adquiridos fora da organização;
- Atividades-chave, que são as atividades essenciais para que seja possível entregar a Proposta de valor;
- Recursos principais, que são os recursos necessários para realizar as atividades-chave;

- Proposta de valor, que estabelece o que a organização vai oferecer ao mercado envolvente e qual o valor que será entregue aos clientes;
- Relacionamento com os clientes, que estabelece como é que a organização se relaciona com cada segmento de clientes;
- Canais, que estabelece como é que o cliente compra/recebe o seu produto/serviço;
- Segmento de clientes, que estabelece quais são os segmentos de clientes que são o foco da organização;
- Estrutura de custos, que espelha os custos relevantes e necessários para que a estrutura proposta possa funcionar;
- Fontes de receita, que estabelece as formas de obter receitas por meio de propostas de valor.

Tabela 6 - Modelo de CANVAS representativo do projeto

Acessibilidade na <i>web</i> e no mundo <i>e-commerce</i>				
Parcerias principais (Key partners)	Atividades-chave ( <i>Key activities</i> )	Proposta de valor (Key propositions)	Relacionamento com clientes (Customer relationships)	Segmento de clientes (Customer segments)
Equipa Black & White  Projetos internos da equipa <i>Black &amp; White</i>	Conhecer as diretrizes de acessibilidade e saber como aplicar;  Levantar as necessidades de acessibilidade e implementar.	Oferecer alternativas à navegação para aceder ao conteúdo, com o uso de tecnologias assistivas ( <i>screen-reader</i> e teclado), com acessibilidade implementada.	Departamento comercial	Cliente/Loja que pretende ter acessibilidade nos seus <i>websites</i> , para estar em cumprimento com a lei.

	Recursos principais (Key resources) Screen-reader Mouseless Desenvolvedores (para desenvolver e aprovar o desenvolvimento)		Canais (Channels)  Aplicação web Produto web	
Estrutura de custos (Cost structure)  Desenvolvedores/programadores; Screen-reader; Formação em acessibilidade.			Fontes de receita (Revenue streams)  Vendas de <i>websites</i> através de parcerias com cliente/loja e consequente aumento de receitas através da venda dos produtos que estão incluídos no <i>website</i> do cliente/loja.	

Como observado na modelo *Canvas* (Tabela 6), ao usar técnicas de *screen-reader*, técnicas *mouseless* e desenvolvedores para implementar acessibilidade e para rever e aprovar o desenvolvimento é possível que a proposta de valor seja atingida, ou seja, é possível que as alternativas à navegação para aceder ao conteúdo de um *website*, com o uso de tecnologias assistivas, sejam implementadas com recurso à acessibilidade e conhecimento adquirido pelos desenvolvedores. Este conhecimento adquirido em acessibilidade é uma atividade-chave do projeto, bem como o levantamento de necessidades de acessibilidade implementar.

As fontes de receita exetáveis do projeto advém do crescimento da carteira de clientes da equipa *Black & White*, ou seja, dos clientes/lojas interessados em realizar um *website* e ter acesso ao nosso produto/ serviço com todas as suas funcionalidades, incluindo acessibilidade. Ao estabelecer novos contratos com clientes/lojas, consequentemente, as formas de receita também aumentam, na medida em que a venda de produtos nos *websites e-commerce* dos clientes/loja também atuam como fontes de receita do projeto/equipa.

As principais parcerias estão relacionadas com a unidade de negócio, isto é, com a equipa *Black & White*, pois incluem os elementos da equipa e os projetos internos já existentes na equipa como fonte de conhecimento e como base de implementação.

## 4 Análise e *Design*

Este capítulo tem dois objetivos:

- Sistematizar os requisitos (funcionais e não funcionais) enunciados nos capítulos anteriores;
- Apresentar e debater alternativas de *design* arquitetural do software a ser implementado.

### 4.1 Análise

Nesta secção serão enumerados todos os requisitos funcionais e não funcionais inerentes ao projeto bem como a apresentação de tecnologias e *frameworks* a ser utilizadas.

#### 4.1.1 Requisitos funcionais

Requisitos funcionais descrevem as funcionalidades que se espera que a aplicação disponibilize, de uma forma completa e consistente. São as funcionalidades de que o utilizador se serve e que aguarda que o sistema ofereça, atendendo aos propósitos para o qual o sistema será desenvolvido. Em suma, abordam o que o sistema deve fazer e retratam as funcionalidades que deve ser possível realizar. Esses requisitos são os descritos na imagem seguinte em forma de casos de uso:

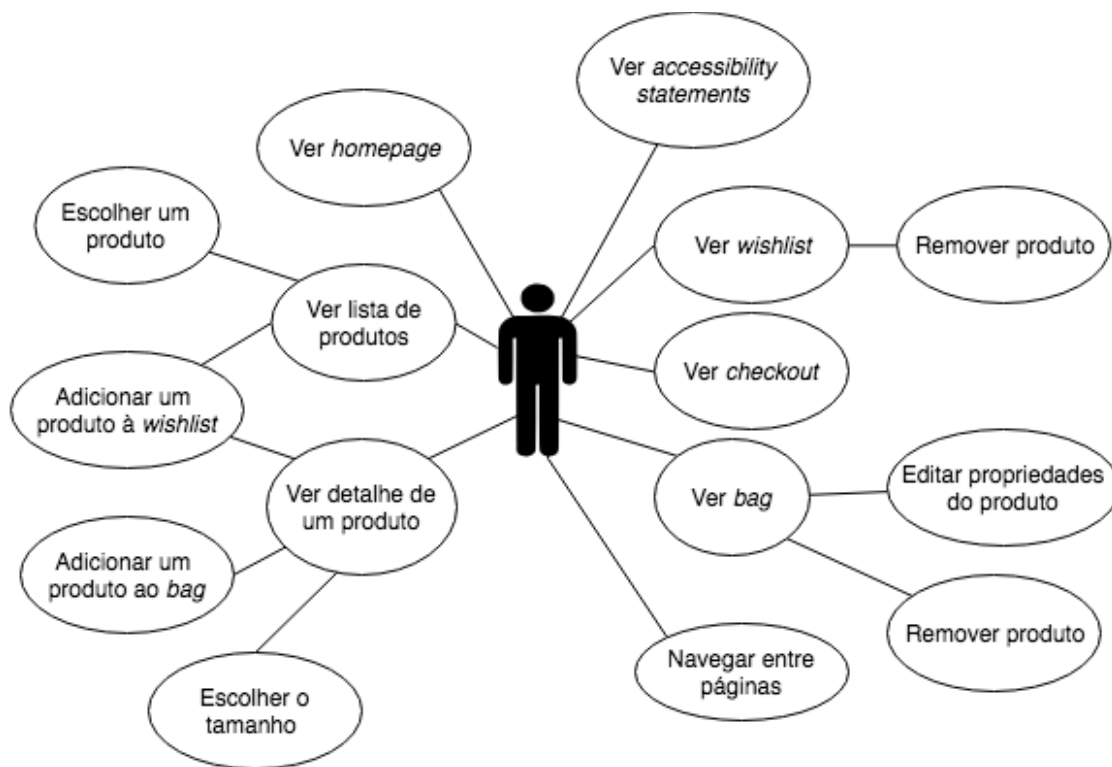


Figura 9 - Diagrama de Casos de Uso da aplicação

Mediante os casos de uso referidos na figura acima, é possível usufruir da aplicação com o seguinte fluxo:

1. Entrar na homepage do *website* e aceder à navegação com o intuito de seleccionar uma página e navegar para a mesma;
2. Entrar numa lista de produtos (*listing page*) e navegar entre produtos, conseguindo adicionar os mesmos à *wishlist*;
3. Seleccionar um produto da lista e ser redireccionado para a página de detalhe do produto;
4. Na página de um produto deve ser possível personalizar o mesmo (seleccionar o tamanho) e adicionar o mesmo ao carro de compras (*bag*);
5. Ainda na página de produto, deve ser possível navegar entre as imagens do mesmo para serem lidas mais informações do produto com as tecnologias assistivas;
6. Adicionar vários produtos ao *bag* repetindo os procedimentos acima;
7. Na página de *bag*, deve ser possível remover um produto, mudar a quantidade de outro produto e começar o processo de compra (*checkout*);
8. Deve ser possível entrar na página de compra.

9. No *footer* da aplicação devem ser possível encontrar o *link* para a página de *accessibility statements*.

#### 4.1.2 Requisitos não funcionais

Os requisitos não funcionais da aplicação, que já têm vindo a ser descritos no decorrer do documento, estão classificados segundo o modelo de FURPS+. Esses requisitos são os apresentados na seguinte tabela, com referência à sua classificação no modelo de FURPS+:

Classificação FURPS +	Requisito não funcional
+ → Legal	Cumprir as leis de acessibilidade (cf. secção 2.1.7) através do uso dos componentes em vários projetos.
F → Funcionalidade	Cumprir as diretrizes de acessibilidade (cf. secção 2.3.2).
F → Funcionalidade	Seguir a estrutura multi-package repository.
F → Funcionalidade	Seguir a metodologia de componentes <i>atomic design</i> .
S → Suporte	Modularidade: alta coesão e baixo acoplamento dos componentes, definida pela interface consumida e disponibilizada de cada um.
S → Suporte	Adequação dos componentes desenvolvidos à adoção noutros projetos.
S → Suporte	Suporte dos <i>browsers</i> mais comuns: Chrome, IE, Safari e Firefox.
U → Usabilidade	Disponibilizar componentes de forma acessível.
U → Usabilidade	Eficácia e eficiência, na medida em que é garantida acessibilidade. Este parâmetro pode ser verificado através dos resultados das ferramentas de auditoria e com reutilização dos componentes acessíveis.
+ → Integração	Baixa complexidade algorítmica, na medida em que é possível reutilizar o componente de forma fácil.
P → Performance	Boa performance, na medida em que os

Classificação FURPS +	Requisito não funcional
	dados aparecem e são anunciados de forma rápida.
<b>F</b> → Funcionalidade	Baixa frequência de falhas com o recurso a testes unitários (cf. secção 5.8).
<b>+</b> → Integração	Fácil manutenção, na medida em que é possível reutilizar o componente de forma fácil.
<b>U</b> → Usabilidade	Compatibilidade entre componentes e projetos.
<b>+</b> → Implementação	Tecnologia a adotar: Javascript (React), HTML 5 e CSS 3.

## 4.2 Arquitetura de software

Após a análise do protótipo/aplicação a desenvolver, foi realizado o estudo de alternativas para a arquitetura do software e o seu processo de desenvolvimento.

“O desenho arquitetural é das primeiras atividades que compõe a área do conhecimento em desenho de software. Devido à complexidade dos projetos, a parte comum a qualquer engenharia para se construir um artefacto complexo, um *website* acessível neste caso, é construir o projeto de acordo com um plano. Por outras palavras, desenho arquitetural é projetar um sistema antes deste ser construído para evitar falhas e se construir um plano/caminho a seguir” (Staveley, 2011).

A descrição do *design* arquitetural fará uso das vistas lógica e de processo do modelo 4+1 (Staveley, 2011). Antes, contudo, serão descritos dois princípios/padrões arquiteturais relevantes.

### 4.2.1 Princípios arquiteturais

Nesta secção serão apresentados dois pontos que fazem parte de boas práticas de soluções arquiteturais e gestão de projetos de software:

- Multi-packages repository;
- Atomic design.

#### 4.2.1.1 Multi-packages repository

*Multi-packages repository*, ou *monorepo*, é um padrão de controlo de origem que serve para gerir projetos de software, onde todo o código-fonte (mesmo de vários

projetos/componentes/aplicações) é mantido num único repositório - ao contrário de multi-repositórios, onde cada projeto tem o seu próprio repositório.

Ou seja, é um repositório que contém mais que um projeto, e tipicamente é grande (no número de *commits*, *branches* e/ou *tags* (cf. secção 5.7), número de ficheiros e tamanho.

#### 4.2.1.2 Atomic design

*Atomic design* (popularizado por Brad Frost no seu livro “*Atomic Design*”)(Frost Brad, 2015) é uma metodologia para a criação de sistemas de *design* (Frost, 2013) - uma abordagem ao *design* de componentes. Ou seja, podemos dizer que é uma linguagem para descrever componentes escritos em múltiplas camadas de abstração, como é o caso deste projeto.

Esta metodologia encoraja a pensar na interface como uma hierarquia de componentes e, consequentemente, a reutilização.

Existem 5 níveis distintos em *atomic design*:

- Átomos, que são os blocos/elementos básicos de construção (Frost, 2013). São os componentes mais simples, mais pequenos e mais abstratos (sem lógica), e providenciam os blocos básicos para a construção de componentes maiores, formando assim a base para uma biblioteca de componentes. Tal como os átomos verdadeiros, tipicamente não são vistos isoladamente e são particularmente úteis quando são agregados para criar componentes maiores, mais ricos e mais significativos;
- Moléculas, que são grupos de átomos conectados e são as unidades fundamentais mais pequenas de um composto. Estas moléculas têm as suas próprias necessidades e servem de espinha dorsal para os sistemas de *design*. Embora possam vir a ser complexas, as moléculas tipicamente são combinações relativamente simples de átomos construídos para a reutilização (Frost, 2013). Mesmo que uma molécula seja composta por múltiplos átomos, estes devem ser compostos de uma forma estratégica para lidar com uma responsabilidade clara e distinta. Isto ajuda a garantir que seja fácil de entender e que pode consumir e compor outras moléculas em organismos de nível superior;
- Organismos, que são grupos de moléculas unidas para formar uma secção relativamente complexa e distinta de uma interface. É neste ponto que as interfaces começam a ficar mais concretas - com os organismos já conseguimos ver a interface final a ganhar forma. Construir moléculas para organismos incentiva a criação de componentes autónomos, portáteis e reutilizáveis (Frost, 2013). Um organismo é mais complexo que uma molécula, pois é composto por moléculas e outros átomos – este módulo tipicamente resulta num componente de interface com o utilizador que controla/manipula uma secção do ecrã/aplicação;
- *Templates*, que consistem principalmente em grupos de organismos juntos para formar páginas. É nesta fase que se começa a ver o projeto/*design* a unir-se e a ver o *layout* final em ação. Os *templates* são muito concretos e fornecem contexto para todas as moléculas e organismos relativamente abstratos (Frost, 2013);

- Páginas, que são instâncias específicas de *templates*. Aqui o conteúdo de exemplo/teste é substituído por conteúdo representativo real para dar uma descrição precisa do que um utilizador irá ver em última análise. Este passo é essencial pois é onde se testa a eficácia do sistema de *design* (Frost, 2013).

A seguinte figura surge com o propósito de ilustrar os níveis de *atomic design* que foram descritos acima.

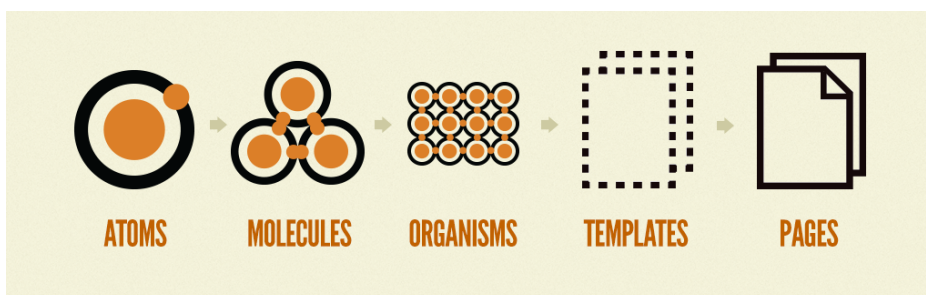


Figura 10 - Níveis de *atomic design* (Frost, 2013)

O conceito de componente ao nível da arquitetura de software do presente projeto é muito importante na medida em que os elementos de acessibilidade podem ser implementados nos mesmos, de forma a conseguirem ser a base para várias páginas, conseguindo manter as boas práticas de reutilização e fácil manutenção. Com o estudo do conceito do *atomic design*, esta rapidamente demonstrou ser a abordagem certa para a aplicação, na medida em que é totalmente escalável e oferece imensas opções.

Se se pensar que todas as partes integrantes do conceito de *atomic design* se completam por ordem de grandiosidade, desde a página (elemento maior) até ao átomo (elemento mais singular), é possível considerar que cada um deles é um componente sendo que quanto mais grandioso, tipicamente contém mais componentes na sua composição, como observado na imagem seguinte. No caso da página de *checkout* pode ser considerada um componente, na medida em que pode ser totalmente reutilizável, no entanto esta também é composta por outros componentes, como Templates, organismos, moléculas e átomos.

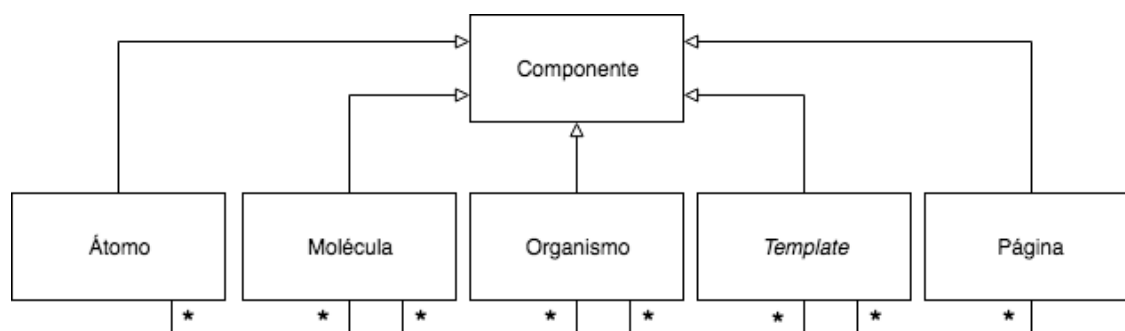


Figura 11 - Conceito atomic design associado ao projeto

## 4.2.2 Vista lógica

A aplicação a desenvolver irá fazer uso dos pontos supracitados: *multi-packages repository* e *atomic design* e está dividida em três principais camadas/áreas:

- Camada do cliente final através da interface gráfica (*user interface*);
- Camada de apresentação, que representa o *website* e todos os componentes *front-end*;
- Camada de lógica de negócio que inclui a manipulação de dados e serviços.

A camada do cliente final representa o ponto de partida de qualquer interação com a aplicação e respetivos componentes. Esta camada é constituída pela interface com o utilizador, pelo utilizador e pelas interações do mesmo com o sistema.

A camada de apresentação e o respetivo motor de renderização de componentes (Componente *render* na figura 12), são os elementos retratados ao longo do projeto, ou seja, é a camada que engloba toda as interações da interface com o utilizador com o *website*, a apresentação de dados recebidos através da comunicação com os serviços e a renderização de componentes através do motor de renderização, que é o foco deste projeto.

A única informação que se sabe sobre a camada de manipulação de dados (Base de dados, *Business Logic e Adapters* da figura 12) é que são realizadas várias comunicações e pedidos a esta camada no sentido de não só obter dados e informações necessárias para representar e manipular na camada de apresentação, mas também guardar informações e registos que o utilizador queira inserir ou modificar na base de dados.

Para melhor compreender a estrutura/arquitetura do sistema desenvolvido foram criados diagramas de componentes que retratam todos os componentes da aplicação. Um diagrama deste género retrata todos os componentes de software envolvidos, as suas interfaces e dependências e são normalmente utilizados para modelar sistemas de alto nível ou para mostrar componentes que pertencem a um nível mais baixo. Um componente representa um agrupamento físico de elementos relacionados logicamente, como é o caso dos componentes representados na figura 12:

- *User Interface*, responsável pela interação do cliente final com o sistema, ou seja, a interação do cliente final com a camada de apresentação;
- *Website*, responsável pela disponibilização de conteúdos e por responder a estímulos que o utilizador reproduza através da *user Interface*. Neste projeto, o *Website* é ainda responsável por comunicar, ou não, com o componente/motor de renderização com a finalidade de receber e conseguir renderizar componentes auxiliares à apresentação de conteúdo;
- *Render*, responsável por renderizar componentes ou pedaços de código estruturados e lógicos de forma a garantir a reutilização e manutenção dos algoritmos e componentes existentes;

- *Adapters*, responsáveis por manipular as respostas dos serviços de forma legível e de fácil interpretação para a camada de apresentação;
- *Business Logic*, responsável por manipular os dados e responder a estímulos e executar ações vinda da camada de apresentação;
- Base de dados, responsável por guardar informação do sistema.

O estilo arquitetural da aplicação define o conjunto de camadas que são constituintes de um sistema. A arquitetura proposta da aplicação a desenvolver pode ser consultada na figura 12 e representa um diagrama de componentes de alto nível do sistema.

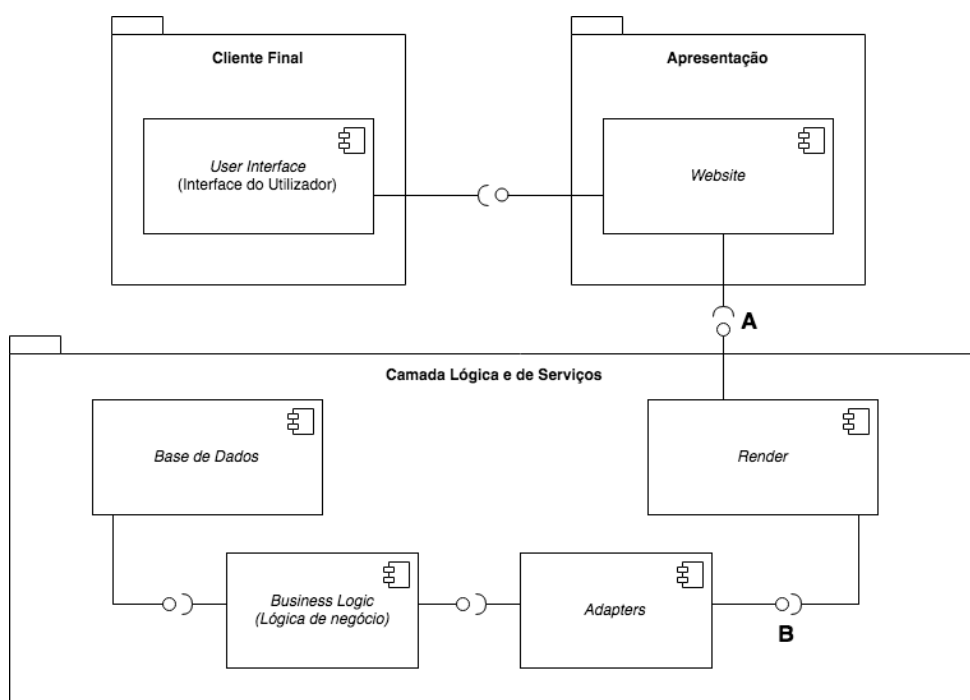


Figura 12 - Diagrama de alto nível representativo do sistema

De forma a direcionar a arquitetura para o sistema desenvolvido no âmbito deste projeto, o *Website* e o motor de renderização - *render*, foi realizado um segundo diagrama de componentes com mais detalhe (figura 13). Este tem como objetivo referenciar a estratégia de gestão de projetos de *software multi-packages repository* (cf. secção 4.2.1.1) no componente direcionado para o motor de renderização de componentes.

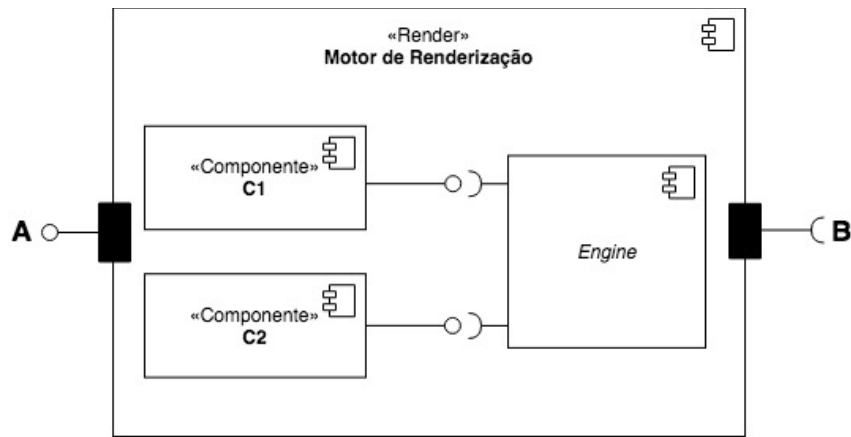


Figura 13 - Diagrama de componentes com detalhe da solução implementada

Com isto, ao entrar em detalhe no conceito de componentes, as figuras 14 e 15 ilustram a possível agregação/composição entre componentes. Um componente pode existir sozinho, mas também pode existir num contexto que necessite de vários componentes. Assim, um componente pode conter (agregar) vários componentes, e por sua vez esses componentes podem agregar mais componentes (figura 14). Em suma, cada componente podem agregar outros componentes (figura 15).

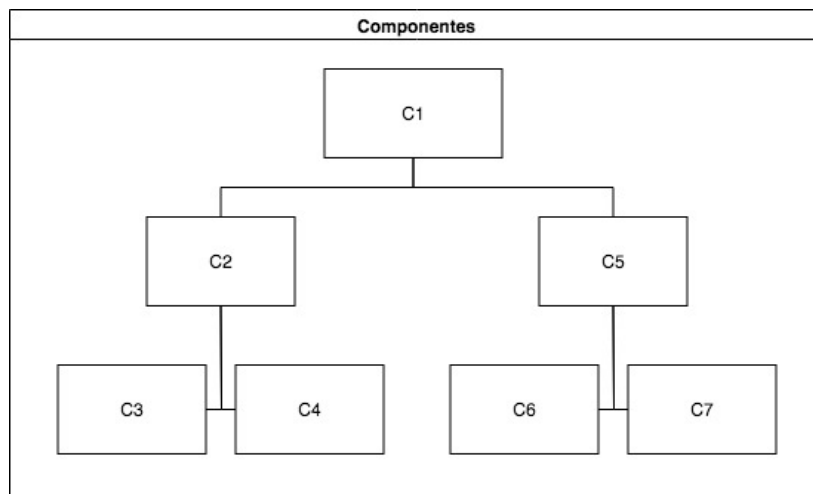


Figura 14 - Diagrama representativo de árvore de componentes

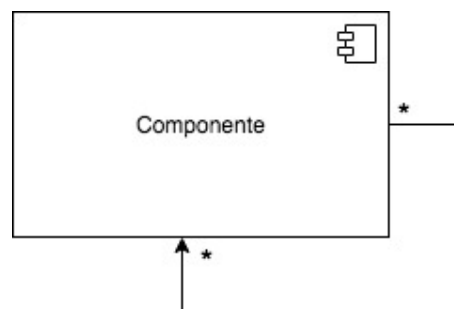


Figura 15 - Diagrama explicativo de composição de componentes

A contribuição para este projeto e para a organização em conjunto com o conceito de *atomic design*, ocorre ao nível do átomo. Isto significa, que o foco maior do projeto é garantir que os componentes/átomos contém elementos de acessibilidade e não possuem qualquer tipo de lógica sendo totalmente agnósticos de contexto, conseguindo assim adaptar-se a qualquer realidade e projeto. Esta contribuição ao nível do átomo pode ser observada na figura 16:

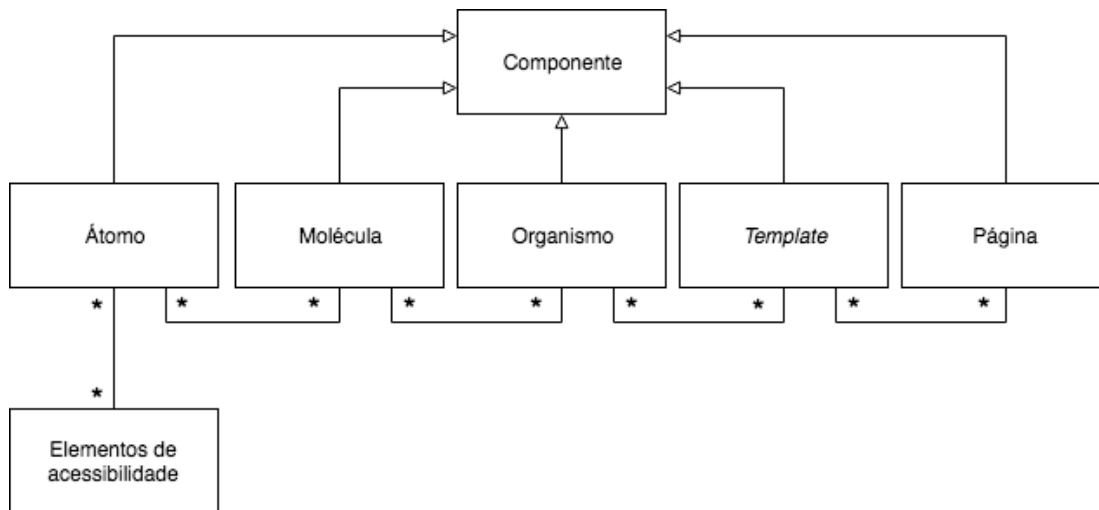


Figura 16 – Contribuição para o projeto ao nível do átomo em conjunto com o *atomic design*

### 4.2.3 Vista de implementação

A vista de implementação tem como objetivos mostrar as partes que constituem o software sob um ponto de vista arquiteturalmente significativo.

Nesta secção serão apresentados dois diagramas que explicam como é que a aplicação desenvolvida utiliza a metodologia associada ao conceito *atomic design* e um diagrama de sequência que retrata a interação entre os intervenientes do sistema, conseguindo esquematizar a propagação de informação entre componentes.

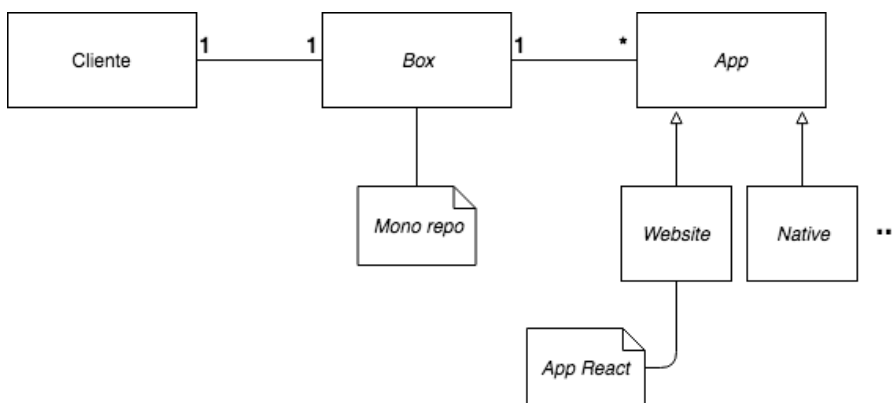


Figura 17 – Diagrama explicativo do projeto desenvolvido

Como é possível observar na figura acima, o projeto é um *website* é apenas uma das aplicações presentes no *Box* (que adota o padrão *Mono repo* cf. secção 4.2.1.1) que é o

produto, ao qual o cliente final tem acesso. Este *website* é o foco principal deste projeto e trata-se de uma aplicação desenvolvida em *React*, com recurso ao padrão *atomic design*. É neste fluxo desde o cliente ao *website* que os esforços devem estar concentrados e onde o projeto se foca. Assim a aplicação final condiciona a forma como os elementos são apresentados, ou seja, é o enquadramento de vários componentes.

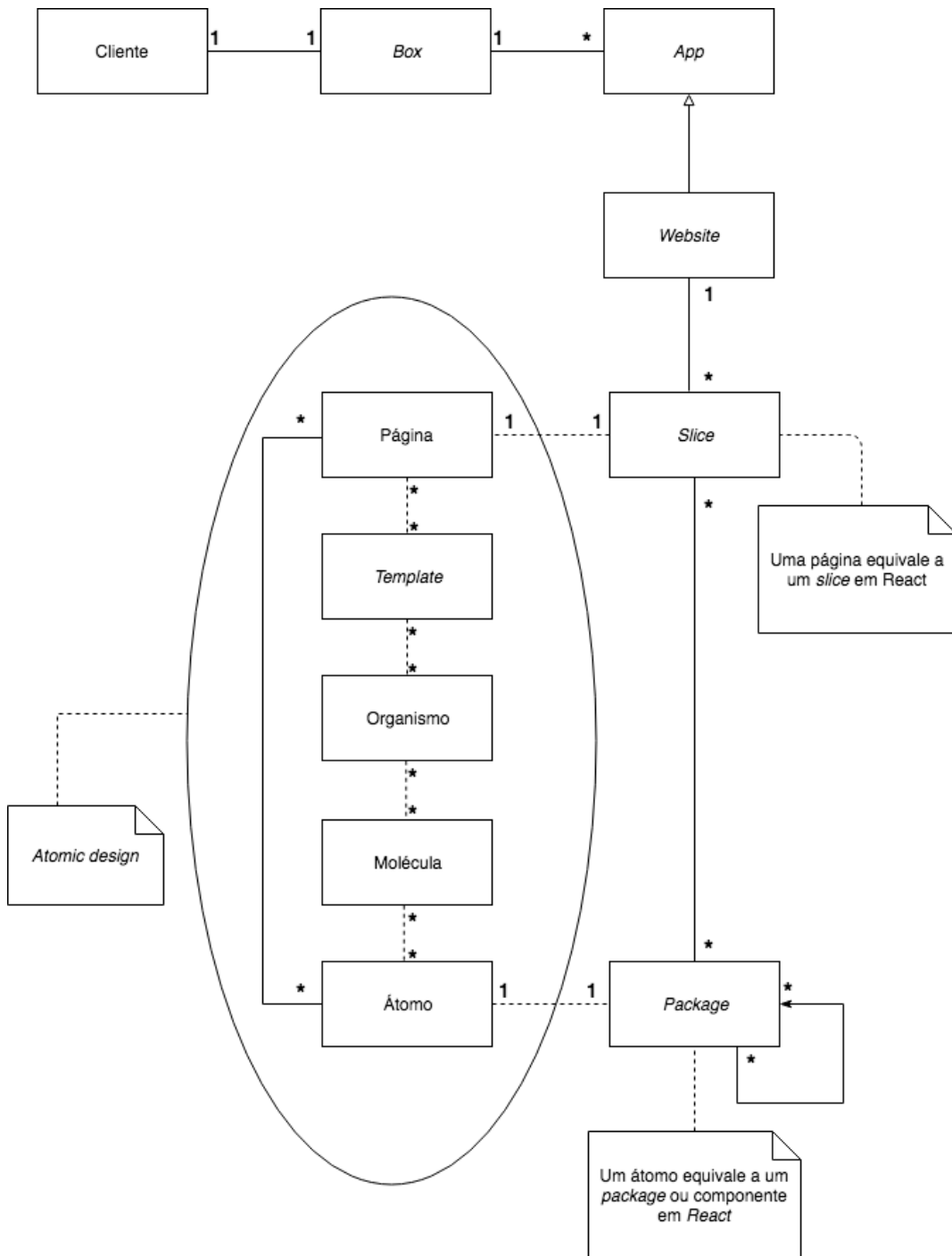


Figura 18 – Diagrama explicativo do uso de *atomic design* ao nível da aplicação

A aplicação que é um *website* contém um ou mais *slices*, que correspondem a páginas na analogia com o termo *atomic design*. Estas páginas podem ser um aglomerado de vários componentes, conforme referido na figura 11, sendo então um conjunto de átomos. Exemplos de páginas no projeto pode ser a lista de produtos e a página de detalhe de um produto. Quando a preocupação é ao nível dos componentes, onde está o foco principal do projeto devido ao interesse da reutilização dos mesmos, é possível dizer que correspondem aos átomos, ao fazer a analogia com o termo *atomic design*. Finalmente, é correto afirmar que os termos *package*, átomo e componente se referem à mesma entidade na medida em que representam a unidade mais singular do projeto, como por exemplo, um botão.

Portanto, a vista de implementação mostra que a terminologia lógica (de *atomic design*) não se mantém. E.g. as páginas são traduzidas para *slices* em *React* e os átomos são traduzidos para *packages/componentes* em *React*, como é possível observar na figura 18. Assim, o desenvolvimento deste projeto não é nada mais do que a implementação de uma aplicação em *React*, com recurso à metodologia *atomic design*, na qual são associados elementos de acessibilidade ao nível dos átomos para promover a sua reutilização e fácil manutenção.

#### 4.2.4 Vista de processo

A vista de processo descreve a dinâmica das partes que constituem o sistema, na realização das tarefas/responsabilidades que lhe estão atribuídas.

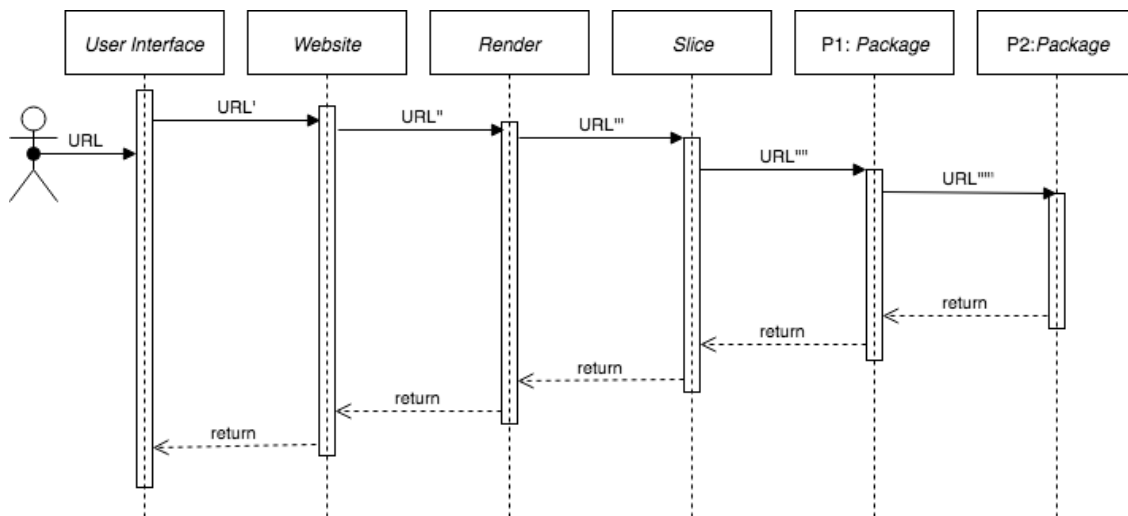


Figura 19 - Diagrama de sequência de interações entre o utilizador e o sistema

O diagrama da figura acima representa as mensagens, pedidos e interações entre os diversos intervenientes do projeto, a interface com o utilizador - *user interface*, o *website*, o *render*, os *slices* e os *packages*.

Assim que um utilizador interage com o *website*, este executa pedidos ao motor de renderização, que contém informação, como *slices*. Estes *slices* devolvem os dados e comunicam entre si até conseguirem renderizar toda a informação conforme o pedido e conseguir devolver resultados para sejam apresentados na interface gráfica, através dos *packages* pelos quais são compostos.

Embora este diagrama represente a comunicação das interações do utilizador com o *website* e consequentes interações entre *slices* e *packages* (dois *packages* neste caso), este pode ser adaptado para interagir com mais ou menos *slices* e mais ou menos *packages*, uma vez que a propagação entre eles é repetitiva a cada ação do utilizador.

Conforme verificado na figura 19, o funcionamento da renderização de *slices* e *packages* da aplicação explica-se da seguinte forma:

1. É selecionada a ação pretendida na interface com o utilizador;
2. É enviado um pedido ao *website*, para que decida o tipo de ações a determinar, o comportamento e quais os *slices/packages* a renderizar, uma vez que o servidor possui esta informação;
3. A ação encontrada determina qual o primeiro *package* (tipicamente uma página/slice) a renderizar e prepara a sua importação;
4. O primeiro *package* recebe o pedido de renderização e verifica se pode devolver a informação, ou se precisa de propagar o pedido para outros *packages* e assim consecutivamente até encontrar o último *package* que pode efetivamente devolver toda a informação necessária;
5. O último *package* recebe o pedido, trata e prepara a informação para devolver e organiza a devolução do mesmo através da propagação entre *package*;
6. Quando a propagação chega ao primeiro *package*, que recebeu o pedido inicial, este junta toda a informação e devolve-a ao *website*, com recurso a um *slice*, que por sua vez, apresenta os dados na interface gráfica, incluindo os elementos de acessibilidade.



# 5 Implementação

Este capítulo sintetiza a implementação/construção do projeto e apresenta algumas recomendações para a implementação das partes de acessibilidade. Todas estas recomendações estão contidas no projeto e são apresentadas neste capítulo tal como são apresentadas todas as conclusões e boas práticas aplicadas das diretrizes de acessibilidade (cf. 2.3.2).

Este capítulo tem vários objetivos:

- Apresentar os cuidados gerais a ter ao longo da construção duma aplicação;
- Apresentar os componentes desenvolvidos no âmbito de acessibilidade;
- Apresentar as páginas que foram desenvolvidas;
- Apresentar os processos de GIT e de contribuição utilizados.

## 5.1 Cuidados gerais

Este subcapítulo apresenta algumas implementações que devem ser tidas em conta ao longo de toda a aplicação, ou seja, em todas as páginas e componentes. São cuidados que, muitas vezes, não são difíceis de atingir, são considerados boas práticas e seguem as diretrizes de acessibilidade (cf. secção 2.3.2).

### 5.1.1 Semântica HTML

O uso de HTML semântico é um dos cuidados que todos os desenvolvedores devem ter e acaba por ser um cuidado básico que traz muitos benefícios a nível de acessibilidade, porque ajuda os utilizadores de leitores de ecrã a entender a estrutura do conteúdo das páginas web. Este cuidado não deve ser ignorado e, quando possível, deve ser utilizado código com significado semântico, como o uso correto e lógico de *headings* (*h1*, *h2*, ..., *h5*), *main*, *footer*, *ul* (*unordered list*), *p* (*paragraph*), entre outros.

No caso de *websites e-commerce*, há muito conteúdo que não é controlável na página, como é o caso de páginas geradas no servidor em que o desenvolvedor da UI não tem qualquer tipo de controlo, como os cabeçalhos ou conteúdos publicitários. No entanto, o objetivo do HTML semântico não é ser “tudo ou nada” e por isso o pretendido é sempre fazer o que é possível e que pode beneficiar a acessibilidade e as pessoas que usufruem das tecnologias assistivas.

A figura seguinte ilustra um bom exemplo semântico pois contém as *tags* `<header>`, `<nav>`, `<ul>` e `<li>`. Este exemplo está presente no *header* do projeto e é reutilizado em todas as páginas:

```
▼<header data-reactid="3">
  ▼<nav class="navbar navbar-expand-md navbar-light bg-light border-bottom" data-reactid="4">
    ▶<a class="navbar-brand" href="/us/" data-reactid="5">...</a>
    ▶<button type="button" data-toggle="collapse" data-target="#navbarSupport" data-aria-expanded="false" aria-label="Toggle navigation" class="navbar-toggler" data-reactid="6">...</button>
    ▼<div class="collapse navbar-collapse" id="navbarSupportedContent" data-reactid="7">
      ▼<ul class="navbar-nav mr-auto" data-reactid="12">
        ▶<li class="nav-item" data-reactid="13">...</li>
        ▶<li class="nav-item" data-reactid="17">...</li>
        ▶<li class="nav-item" data-reactid="19">...</li>
        ▶<li class="nav-item" data-reactid="21">...</li>
      </ul>
      ▶<ul class="navbar-nav align-items-center" data-reactid="23">...</ul>
    </div>
  </nav>
</header>
```

Figura 20 - Exemplo de HTML semântico no *header* da aplicação

Existem alguns elementos HTML que por recomendação, ao estarem presentes, acrescentam valor semântico e ajudam a estruturar todas as páginas de uma aplicação como: `<header>`, `<nav>`, `<main>`, `<article>`, `<aside>` e `<footer>`.

Elementos como a `<div>` e `<span>` são apenas úteis para controlo de *layout*, pois não têm qualquer tipo de significado semântico e não suportam o uso do teclado em nenhum *browser*, não acrescentando nada no que diz respeito a acessibilidade. Por exemplo, estas *tags* nunca devem ser usadas para renderizar um botão quando na realidade existe um elemento HTML nativo para esse efeito – o `<button>` que tem significado semântico.

Por sua vez, as tecnologias assistivas também usam a semântica do HTML para entender e detetar a melhor maneira de transmitir cada elemento ao utilizador. No caso dos leitores de ecrã, o HTML semântico dá contexto aos leitores de ecrã. Os utilizadores com visão podem usufruir dos elementos visuais na página para navegar até o conteúdo desejado de maneira linear e rápida, o que não se verifica nos utilizadores invisuais. Como os leitores de ecrã emitem áudio, estes elementos visuais não são tão úteis. Assim, os leitores de ecrã fornecem meios alternativos de navegação, permitindo que as pessoas invisuais naveguem entre diferentes tipos de conteúdo, como *links*, formulários, cabeçalhos, listas e parágrafos. Se todo o conteúdo das páginas *web* fosse composto por *divs* e *spans*, não era possível fornecer aos leitores de ecrã a oportunidade de contextualizar o conteúdo das páginas. A navegação por teclado também beneficia com uso correto do HTML semântico, como é o caso dos formulários (`<form>`), menus de opções, navegação (`<nav>`), vídeo e áudio onde é particularmente difícil aceder quando se depende apenas do teclado. Quando os formulários

semânticos são usados corretamente estes transmitem toda a informação que está associada ao campo de preenchimento em questão, informam se uma determinada caixa de seleção está ativa ou não, entre outros comportamentos que são herdados automaticamente com o uso desta técnica de acessibilidade. O facto de não se usar HTML semântico nos formulários pode fazer bastante diferença para um utilizador que navegue nas páginas por teclado pois pode trazer frustração por não saber onde se encontra, em que estado é que os elementos estão e se tem de preencher algum campo de texto, podendo levar ao abandono do *website*.

### 5.1.2 Título das páginas

O título das páginas, também conhecido por *page title*, é considerado uma forma de orientação para os utilizadores de tecnologias assistivas, uma vez que lhes permite identificar exatamente em que página estão. Neste caso, com a ajuda dos leitores de ecrã estes anunciam o título da página quando a mesma é carregada com sucesso.

Este elemento, denominado *title* (*tag <title>*) é encontrado na secção HEAD (*tag <head>*) do HTML de uma página *web* e normalmente contém uma descrição sucinta e clara do que a página representa, muitas vezes conseguindo fazer correspondência com o título visual que aparece na página, normalmente *tags* h1 (*heading* de nível 1).

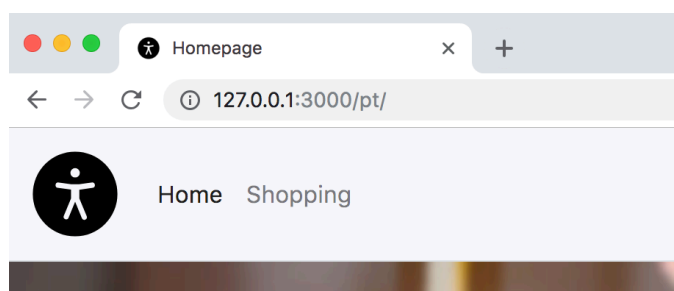


Figura 21 - Título da página inicial no separador do *browser*

O título da página (*page title*) tem algumas características:

- Aparece na barra de título das páginas de alguns *browsers*;
- Aparece nos separadores quando há várias páginas *web* abertas;
- Aparece nos resultados dos mecanismos de pesquisa (SEO);
- É usado como nome padrão para os favoritos e *bookmarks*;
- É lido pelos *screen-readers*.

Este elemento satisfaz um critério de sucesso das diretrizes de acessibilidade (cf. secção 2.3.2) que é o 2.4.2 que defende que as páginas *web* devem ter títulos que descrevam o seu tópico e/ou objetivo e pertence ao nível A (cf. secção 2.3.2) de acessibilidade. Ao cumprir este critério, é permitido aos utilizadores identificar de forma rápida e fácil as informações contidas numa determinada página *web*, é permitido aos utilizadores portadores de incapacidades visuais diferenciarem o conteúdo apenas pelo título da página quando têm

diversos separadores abertos, é permitido às pessoas portadoras de incapacidades cognitivas, como memória limitada de curto prazo ou incapacidade de leitura, beneficiarem da capacidade de identificar o conteúdo ao ouvir o título, bem como pessoas com incapacidades de mobilidade e que operem na *web* com tecnologias assistivas de áudio.

Neste projeto, os títulos das páginas estão todos sucintos, claros e adequados ao conteúdo das páginas com o objetivo de ajudar o utilizador final a perceber em que parte da aplicação está. Assim no projeto são apresentados como títulos das páginas os seguintes:

- “*Homepage*” - página inicial
- “*Listing page*” - página da lista de produtos
- “*Product detail page*” - página de detalhe de um produto
- “*Wishlist page*” - página da lista de desejos
- “*Basket page*” - página do carrinho de compras
- “*Checkout page*” - página de compra
- “*Page not found – 404*” - página não encontrada / página de erro

### 5.1.3 Landmarks

Os *landmarks* são úteis para utilizadores dos leitores de ecrã, porque ajudam a navegar de forma fácil nas páginas *web* de uma aplicação devidamente estruturada, facilitando a orientação o utilizador.

Alguns elementos presentes na especificação HTML5, como o *header*, *aside*, *navigation* e *main* já transmitem de forma nativa algumas informações especiais aos leitores de ecrã e são uma referência nos *browsers* mais recentes. Os *landmarks* também podem ser definidos pelo uso atributos ARIA (cf. secção 5.1.4) para reescrever a semântica de um elemento nativo ou personalizado. Existem ainda alguns *landmarks* que só podem ser definidos com o uso de um atributo ARIA.

Como referido, os utilizadores de leitores de ecrã beneficiam do uso correto dos *landmarks*, no entanto as pessoas sem incapacidades visuais e que usam o teclado podem também beneficiar dos mesmos, através de algumas ferramentas externas para *landmarks*, que permitem a navegação mais fácil pelos elementos principais e suas secções, ao invés de depender apenas do uso da tecla *tab* e dos *skip links* (cf. secção 5.2.1) (quando existentes) para navegar sequencialmente no conteúdo das páginas.

Os *landmarks* referem-se a áreas que são facilmente reconhecidas nas páginas/aplicações e definem as expectativas dos utilizadores quanto à importância do conteúdo que cada um deve conter. Assim identificam-se alguns *landmarks* que devem ser usados ao longo das aplicações:

- ***banner*** - área que contém informações globais que se repetem em todas as páginas de uma aplicação, em vez de apresentar informações específicas ou ações.

Normalmente, o logotipo da aplicação e a navegação principal são esperados como conteúdos dentro de um *banner*. Como não existe um elemento HTML nativo para o *banner*, os *browsers* modernos tratam a *tag header*, como um *banner* e o ideal é que haja apenas um *landmark* para o *banner* por página. Para garantir o uso deste *landmark* pode ainda ser usado a ARIA `role="banner"`.

- **complementary** – área que contém informações secundárias e que devem estar relacionadas com a página devendo ser claras e objetivas. Para garantir o uso deste *landmark* existe o elemento nativo HTML `<aside>` e pode ser usada, para complementar a informação, a `role="complementary"`.
- **contentinfo** – área que possui informações sobre a aplicação. Nos *browsers* modernos, exceto no Safari, eles inferem a *role contentinfo* na *tag footer*. Caso seja necessário, deve ser utilizado um *aria-label* (cf. secção 5.1.4) para que os elementos sejam facilmente distinguidos.
- **form** – os formulários são uma parte importante em todas as aplicações e por isso, devem possuir um *landmark* que ajude os utilizadores das tecnologias assistivas a encontrá-los com facilidade e a perceber o seu significado. Quando há vários *landmarks* para formulários numa só página, cada um deve ter o seu próprio título visível. Este título fornece um nome acessível (cf. secção 5.1.13) ao formulário por meio do uso de *aria-labelledby* (cf. secção 5.1.4), fazendo com que seja possível navegar através dos menus de *landmarks* oferecidos nos leitores de ecrã. Se o caso de ter o título do formulário visível não for possível por questões de *layout/design*, a solução é acrescentar um *aria-label* (cf. secção 5.1.4) para dar o nome acessível. Por sua vez, este nome acessível basta ser algo simples e sucinto como “contacto, login ou newsletter” para transmitir facilmente o propósito de um formulário.
- **main** – área que contém o conteúdo principal de uma página. Nos *browsers* modernos, o *landmark main* é garantido através do uso da *tag* HTML `<main>`. Para garantir o suporte por parte de todos os *browsers*, ou se o elemento HTML `<main>` não puder ser utilizado, existe a alternativa do ARIA `role="main"` para reforçar a interpretação de um elemento como *landmark*. O ideal é que exista apenas um *landmark* do tipo *main* por página, uma vez que ter múltiplos *landmarks main* seria contraditório e confuso no sentido em que os utilizadores o usam para encontrar o conteúdo principal de uma página de forma rápida. Finalmente, este *landmark* é um dos mais importantes, quando se usam em conjunto com os *skip links* (cf. secção 5.2.1), uma vez que estes permitem aos utilizadores que navegam por teclado contornar o conteúdo e avançar para as zonas das páginas que desejam, como avançar para o conteúdo principal neste caso.
- **navigation** – O elemento nativo HTML `<nav>` já tem associado um ARIA *role* (cf. secção 5.1.4) de navegação, que já é interpretado da forma correta na maioria dos *browsers*. Estes *landmarks* devem ser usados conforme seja necessário ao longo de uma página, e caso sejam múltiplos, devem ter nomes acessíveis (cf. secção 5.1.13) para facilitar a sua distinção. Por exemplo, no projeto existe uma navegação primária que possibilita a navegação entre páginas de produtos e a página inicial e possui outras navegações específicas por página que estão devidamente identificadas para serem úteis e fáceis de aceder, como é o caso da paginação na lista de produtos.

- **region** – O ideal é que todo o conteúdo das páginas deva estar presente nos *landmarks* apresentados anteriormente. No entanto, pode existir situações em que tal não seja possível, ou que seja necessário que determinada área seja destacada do restante conteúdo, de forma a que seja também encontrada de forma rápida pelos utilizadores de tecnologias assistivas. Para essa finalidade existe o *landmark region* que é um *landmark* genérico e que deve ser devidamente identificado através de um *aria-labelledby* ou *aria-label* (cf. secção 5.1.4) para permitir o rápido acesso e identificação. O uso deste *landmark* deve também ser limitado, uma vez que vários *region* numa página podem diluir a utilidade dos mesmos e pode dificultar a navegação entre os pontos realmente importantes das páginas. Se de facto existirem muitas *regions* por página, estas devem ser revistas e deve ser feita uma reflexão com a finalidade de perceber se estes conteúdos não ficariam também facilmente identificados com *tags* como `<section>` ou `<article>` com a associação de uma hierarquia de títulos (cf. secção 5.1.9) apropriada. Finalmente, não existe nenhum elemento HTML nativo que identifique este *landmark*, por isso deve ser usada a `role="region"` com o auxílio de nomes acessíveis para garantir o efeito desejado.
- **search** – O *landmark search* é um tipo específico de *region* pertencente aos formulários e possui um papel exclusivo que permite uma descoberta mais fácil entre todos os formulários que podem pertencer a uma página. Como não existe um elemento *search* nativo (é importante não confundir com `input type = "search"`), deve ser adicionado um ARIA `role="search"` no formulário para transmitir apropriadamente este *landmark*.

Como foi anteriormente mencionado, quanto mais *landmarks* existirem numa página *web*, especialmente *landmarks* do mesmo tipo, menos significado eles possuem na sua capacidade de descrever regiões exclusivas das páginas *web*. A forma padrão de um *landmark* ser exposto é através do seu *role*, como: *navigation*, *banner* e *main* que por sua vez, se forem únicos na página, já ajudam os utilizadores a identificar as áreas. No entanto, se forem múltiplos esta identificação é mais difícil e por isso uma descrição exclusiva pode ajudar a distinguir vários *landmarks* e fornece contexto ao utilizador.

Como exemplo, na figura seguinte é possível verificar o uso de um *aria-label* para acrescentar significado a um elemento de navegação na página de lista de produtos.

```

<nav aria-label="Pagination">
  <ul class="pagination justify-content-center">
    <li class="page-item disabled">...</li>
    <li class="page-item active">
      <a class="page-link" aria-current="true" aria-label="Current
        page, page 1" href="/pt/shopping#1">...</a>
    </li>
    <li class="page-item">...</li>
    <li class="page-item">...</li>
    <li class="page-item">...</li>
  </ul>
</nav>

```

Figura 22 - Utilização do *landmark navigation* com um *aria-label* apropriado

A utilização do *landmark* acima vai anunciar “*Pagination navigation*” quando um utilizador navegar no *website* através do leitor de ecrã ou quando abrir a lista de *landmarks* disponíveis do leitor de ecrã na página.

Em suma, utilizar *landmarks* com recurso a elementos nativos HTML é uma boa prática, no entanto com o objetivo de oferecer suporte a todo o tipo de *browsers* deve ser adicionada a *aria role* mais apropriada para o caso e, no caso de existir múltiplos *landmarks* do mesmo tipo por página, é sugerido que se use um *aria-label* para possibilitar a identificação e diferenciação.

Na tabela abaixo é possível verificar o *landmark* associado a cada elemento nativo HTML5:

Tabela 7 - Lista de elementos nativos HTML5 e respetivos *landmarks*

Elemento nativo HTML5	<i>Landmark</i> associado
<code>&lt;aside&gt;</code>	complementary
<code>&lt;footer&gt;</code>	<i>contentinfo</i> – o elemento <i>footer</i> não corresponde ao <i>landmark contentinfo</i> quando é descendente de um dos seguintes elementos: <i>article, aside, main, nav e section</i>
<code>&lt;form&gt;</code>	<i>form</i> – quando tem um nome acessível e usa atributos como <i>aria-labelledby, aria-label</i> ou título associados
<code>&lt;header&gt;</code>	<i>banner</i> – o elemento <i>header</i> não corresponde ao <i>landmark banner</i> quando é descendente de um dos seguintes elementos: <i>article, aside, main, nav e section</i>
<code>&lt;main&gt;</code>	main
<code>&lt;nav&gt;</code>	navigation
<code>&lt;section&gt;</code>	<i>region</i> - quando tem um nome acessível e usa atributos como <i>aria-labelledby, aria-label</i> ou título associados

#### 5.1.4 WAI-ARIA

WAI-ARIA (*Web Accessibility Initiative’s Accessible Rich Internet Applications*) ou ARIA é uma tecnologia que colmata muitos problemas de acessibilidade na web, adicionando novas formas de providenciar semântica ao código HTML, de forma a que os *browsers* e as tecnologias assistivas os reconheçam e anunciem o que está a acontecer recorrendo à informação extra. As funcionalidades oferecidas por esta tecnologia definem várias formas de modificar o conteúdo das páginas tornando-as mais acessíveis com a finalidade de ser

possível, para as pessoas com incapacidades, utilizar as tecnologias assistivas. Estas funcionalidades complementam o comportamento nativo dos elementos HTML para que as interações das aplicações possam ser interpretadas pelas tecnologias assistivas nos casos em que é necessária mais informação complementar.

Existem três funcionalidades distintas no uso de ARIA e que são exploradas neste projeto:

- Roles
- Properties/attributes
- States

Nenhum atributo ARIA afeta em nada as páginas *web*, tanto a sua estrutura como elemento da DOM, exceto as informações que são expostas às API de acessibilidade dos *browsers* (onde os leitores de ecrã obtêm as suas informações). Assim, o uso correto de atributos ARIA é essencial e útil para quatro áreas principais:

- **Landmarks** – Os ARIA *roles* podem atuar como *landmarks* que replicam o comportamento de alguns elementos HTML5 nativos (por exemplo, `<nav>`);
- **Atualizações dinâmicas do conteúdo de uma página** – os leitores de ecrã tendem em ter dificuldade a interpretar o conteúdo em constante mudança, por isso com a utilização de uma propriedade ARIA (*aria-live*) é possível informar os utilizadores dos leitores de ecrã que determinada área da página *web* é atualizada, como é o caso dos contadores da lista de desejos e do carrinho de compras no projeto;
- **Melhorar a navegação por teclado** – existem elementos HTML que já fornecem de forma nativa a navegação por teclado. No entanto, nos elementos em que isto não é possível, os atributos ARIA fornecem meios para permitir que outros elementos consigam receber o *focus*, com recurso ao *tabindex* (cf. secção 5.1.5);
- **Providenciar acessibilidade a elementos não semânticos** – existem algumas páginas *web* que são implementadas sem recurso a HTML nativo e que são desenvolvidas com base em encadeamento de várias *tags* `<div>`. Quando isto acontece, em termos de acessibilidade é muito difícil conseguir navegar entre estes elementos, pois eles não têm qualquer tipo de semântica. Assim, com recurso às *roles* ARIA é possível fornecer alguma informação extra e semântica, como usar *roles* do tipo *button*, *listbox* ou *tabgroup* e propriedades como o *aria-required*.

Existem algumas regras associadas ao uso de todo o tipo de atributos ARIA:

- Não se deve usar atributos ARIA se estes não forem estritamente precisos. Com o uso correto das *tags* HTML em muitos casos já não é preciso estes atributos;
- Não se deve mudar o significado semântico de um elemento HTML;
- Os elementos que são interativos devem ser possíveis de interagir com o uso do teclado;

- Deve evitar usar-se `role="presentation"`, visto que esta regra remove o comportamento nativo do elemento.

Uma vez mais, os atributos ARIA devem ser apenas usados quando é estritamente necessário. O ideal é que sejam usados os elementos HTML nativos para fornecer a semântica exigida pelos leitores de ecrã para informar aos utilizadores do que está a acontecer. Muitas vezes tal não é possível, porque é necessário dar mais informação aos utilizadores dos leitores de ecrã (informação com mais pormenor) do que aos utilizadores da navegação por teclado, que conseguem ter acesso à informação visual - nestes casos, os atributos ARIA podem ser uma ferramenta valiosa para aprimorar a acessibilidade.

#### 5.1.4.1 Roles

Os ARIA *roles* (assim conhecidos) definem o que é que um elemento HTML e qual é a sua função. Muitos destes *roles* estão diretamente relacionados com os *landmarks* (cf. secção 5.1.3) e duplicam o valor semântico aos elementos HTML nativos, como é o caso dos seguintes exemplos:

- `role="navigation"` (corresponde à *tag* HTML `<nav>`)
- `role="main"` (correspondente à *tag* HTML `<main>`)
- `role="complementary"` (correspondente à *tag* HTML `<aside>`)

Por outro lado, também existem ARIA *roles* que descrevem a estrutura dentro de uma página, facilitando a navegação entre o conteúdo e que estão muito presentes nas interfaces recentes das aplicações *web*, como é o caso do `role="banner"` e `role="search"`.

A tabela abaixo enuncia os ARIA *roles* que estão presentes no projeto:

Tabela 8 - Exemplo de ARIA *roles* utilizados ao longo do projeto

ARIA role	Funcionalidade	Exemplo
<code>role="alert"</code>	Esta <i>role</i> é usada para comunicar uma mensagem importante ao utilizador de forma imediata.	<pre>&lt;div role="alert" aria-live="polite"&gt;   &lt;span&gt;     Your wishlist is empty   &lt;/span&gt; &lt;/div&gt;</pre>
<code>role="contentinfo"</code>	Esta <i>role</i> é usada para identificar informações que se repetem ao longo de todas as páginas e, normalmente, é denominada <i>footer</i> .	<pre>&lt;footer role="contentinfo" aria-label="Footer"&gt;   &lt;h5&gt;Contacts&lt;/h5&gt;   &lt;ul&gt;     &lt;li&gt;Name:&lt;/li&gt;     ...   &lt;/ul&gt; &lt;/footer&gt;</pre>
<code>role="search"</code>	Esta <i>role</i> é também um <i>landmark</i> e contém	<pre>&lt;form role="search"&gt;   &lt;div&gt;     &lt;input type="search" placeholder="Search products"&gt;</pre>

ARIA role	Funcionalidade	Exemplo
	objetos que combinados melhoram a funcionalidade de pesquisa de um <i>website</i> .	<pre>&lt;button aria-label="Submit search term" type="submit"&gt;Search&lt;/button&gt; ... &lt;/div&gt; &lt;/form&gt;</pre>
<code>role="img"</code>	Esta <i>role</i> funciona como <i>container</i> <sup>12</sup> de elementos que juntos formam uma imagem, que é o caso das imagens SVG.	<pre>&lt;svg role="img" tabindex="-1" aria-hidden="true"&gt;   &lt;title id="Title"&gt;Title&lt;/title&gt; &lt;/svg&gt;</pre>
<code>role="presentation"</code>	Esta <i>role</i> é útil para remover o significado semântico de um determinado elemento e dos seus filhos.	<pre>&lt;table role="presentation"&gt;   &lt;tbody&gt;     &lt;tr&gt;       &lt;td&gt;Some text&lt;/td&gt;     &lt;/tr&gt;     ...   &lt;/tbody&gt; &lt;/table&gt;</pre>

#### 5.1.4.2 Properties

Os atributos ARIA (*aria attributes/properties*) definem propriedades dos elementos, que podem ser usados para lhes dar significado extra ou mais significado semântico. Como é o caso dos seguintes exemplos:

- *aria-required* (especifica que um elemento pertencente a um formulário precisa de ser preenchido para ser válido);
- *aria-labelledby* (especifica qual é o *id* do elemento HTML que funciona como especificação/*label* de um elemento em específico).

A tabela abaixo enuncia as ARIA *properties* que estão presentes no projeto:

---

<sup>12</sup> Contentor ou agregador de elementos

Tabela 9 - Exemplo de ARIA *properties* utilizados ao longo do projeto

ARIA <i>property</i>	Funcionalidade	Exemplo
aria-label	Define um valor que auxilia a identificar o elemento atual e ajuda o utilizador a identificar o elemento. Esta propriedade tem o mesmo propósito que a <i>aria-labelledby</i> .	<pre>&lt;a href="#banner" aria-label="Skip to homepage banner with actions"&gt;   Skip to banner &lt;/a&gt;</pre>
aria-labelledby	Define o elemento ou elementos que identificam o elemento atual. Proporciona aos utilizadores informação extra para identificar certos elementos na página.	<pre>&lt;svg aria-labelledby="logo-icon" role="img" tabindex="-1"&gt;   ...   &lt;title id="logo-icon"&gt;     A1ly project   &lt;/title&gt; &lt;/svg&gt;</pre>
aria-describedby	Define o elemento ou elementos que descrevem o elemento atual. Proporciona aos utilizadores informação extra para descrever o elemento.	<pre>&lt;div aria-describedby="info"&gt;   &lt;h1&gt;Bag&lt;/h1&gt;   &lt;p id="info"&gt;     Awesome page.   &lt;/p&gt; &lt;/div&gt;</pre>
aria-live	Define que certo elemento está a ser atualizado e é usado para definir a prioridade com que certa modificação num elemento vai ser anunciada. Esta propriedade pode ter dois valores: <i>polite</i> (as tecnologias assistivas notificam sobre as alterações, mas não interrompem a tarefa atual) e <i>assertive</i> (as tecnologias assistivas notificam imediatamente o utilizador).	<pre>&lt;div role="alert" aria-live="polite"&gt;   &lt;span&gt;     Your wishlist is empty   &lt;/span&gt; &lt;/div&gt;</pre>
aria-controls	Identifica o elemento ou elementos cujo conteúdo é controlado pelo elemento atual.	<pre>&lt;button type="button" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"&gt;   ... &lt;/button&gt;</pre>

#### 5.1.4.3 States

Os estados (*states*) ARIA são propriedades especiais que definem a condição atual de um elemento. Estes estados diferem das propriedades (cf. secção 5.1.4.2), no sentido em que as propriedades não mudam durante o ciclo de vida de uma aplicação, enquanto que os estados podem mudar, geralmente com mudanças realizadas através do *JavaScript*.

A tabela abaixo enuncia os *ARIA states* que estão presentes no projeto:

Tabela 10 - Exemplo de *ARIA states* utilizados ao longo do projeto

ARIA state	Funcionalidade	Exemplo
aria-disabled	Anuncia para os leitores de ecrã que certo elemento num formulário está inativo.	<pre>&lt;button aria-disabled="true"&gt;   Button text &lt;/button&gt;</pre>
aria-hidden	Indica se o elemento está exposto para a API de acessibilidade.	<pre>&lt;span aria-hidden="true"&gt;   Text &lt;/span&gt;</pre>
aria-busy	Indica que certo elemento está a sofrer alterações e que as tecnologias assistivas devem esperar até essas modificações estarem completas, para serem expostas aos utilizadores.	<pre>&lt;section aria-busy="true"&gt;   &lt;div&gt;Loading ...&lt;/div&gt; &lt;/section&gt;</pre>
aria-current	Identifica o elemento que representa o item atual num conjunto de elementos relacionados.	<pre>&lt;a aria-current="true" aria-label="Current page, page 1" href="/ag/shopping#1"&gt;   Page 1 &lt;/a&gt;</pre>
aria-expanded	Indica se o elemento, ou outro elemento descendente, está atualmente expandido ou não.	<pre>&lt;button type="button" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"&gt;   ... &lt;/button&gt;</pre>

### 5.1.5 Tab order & tabindex

Navegar no conteúdo de uma página *web* ou aplicação deve sempre seguir uma ordem lógica e deve espelhar a ordem visual da navegação e dos elementos interativos de uma página. Os utilizadores que navegam com o teclado (*keyboard users* – usam a tecla *tab*) esperam aceder ao conteúdo sequencialmente da esquerda para a direita e de cima para baixo, através dos elementos que possam receber *focus* na página. Quando se cria páginas *web* é importante que se verifique que a ordem dos elementos no código é a ordem correspondente à ordem visual apresentada. Para certificar este ponto deve-se testar as páginas tentando navegar nas páginas *web* apenas com o uso do teclado. Ao usar a tecla *tab*, deve ser possível aceder a todos os *links* e ações de uma forma previsível com base na posição visual do utilizador na página.

O critério de sucesso 2.4.3 das diretrizes *WCAG* (cf. secção 2.3.2) é sobre este tema, o *focus order*, isto é, sobre manter o *focus* dos elementos e a sequência do uso da navegação por teclado (uso da tecla *tab*) de uma forma lógica, útil e ordeira. Este critério defende que se deve fornecer uma ordem lógica no uso da navegação por teclado, mas que garantir apenas isto não é suficiente para cumprir este critério. Muitas vezes é necessário dar outra ordem aos

elementos, que não a natural, mas isso (forçar um *tab order*) pode facilmente levar a cometer erros de usabilidade e acessibilidade.

Em alguns casos, alguns *websites* podem querer especificar a ordem de navegação por teclado a acompanhar o conteúdo e as suas relações, sem seguir a ordem dos elementos do código. Nestes casos, este comportamento de ordem alternativa é possível especificando um atributo ***tabindex*** que pode receber valores positivos e negativos.

Quando a navegação dos elementos interativos é feita através da tecla *tab*, os elementos recebem o foco (*focus*) em ordem crescente do valor do seu atributo *tabindex* associado. Os elementos que possuem um valor de *tabindex* maior que 0 (zero) recebem o foco antes dos elementos com `tabindex="0"`. Depois de todos os elementos que contiveram o *tabindex* maior que 0, o resto dos elementos interativos da página recebem o *focus* na ordem em que estão definidos na página *web*.

O atributo *tabindex* possui três usos distintos:

- ***tabindex="1"*** (ou qualquer número maior que 0) – define uma ordem de navegação por teclado explícita, com a ordem definida no seu valor numérico, isto é, os elementos com `tabindex="4"` vão receber o *focus* antes dos elementos com `tabindex="5"`. Esta técnica deve ser evitada porque estes elementos recebem o *focus* do teclado antes dos elementos sem valor ou com `tabindex="0"`;
- ***tabindex="0"*** – permite que os elementos, para além dos *links* e formulários, também recebam o foco do teclado. Com este valor, o *tabindex* não altera a ordem do *tab*, mas coloca o elemento no fluxo lógico de navegação por teclado, como se fosse um *link* na página.
- ***tabindex="-1"*** – retira o elemento do fluxo lógico de navegação, permitindo no entanto que os elementos recebam o *focus* através de comportamento *JavaScript* associado.

Teoricamente, o uso do *tabindex* só deve ser usado nos casos em que:

- A ordem de navegação por teclado padrão não é a ideal;
- A ordem de navegação por teclado não pode ser alterada através da reorganização dos elementos no conteúdo e/ou alterando os estilos CSS para refletir o melhor arranjo visual.

Em termos de acessibilidade, e de forma a proteger as boas práticas do uso dos elementos que providenciam boa semântica, deve ser evitado o uso do atributo *tabindex* em conjunto com conteúdo não interativo e que não tenha comportamento padrão para navegação com o teclado. Um exemplo de uma má experiência é usar uma *tag* `<div>` para descrever um botão, em vez de usar de forma direta o elemento `<button>`. Estes componentes, que são forçados a ser interativos, não são listados na árvore de acessibilidade, o que faz com que as tecnologias assistivas não consigam aceder a este conteúdo. A melhor experiência é descrever os elementos com recurso aos elementos existentes com semântica (`<a>`, `<button>`, `<input>`, `<select>`, `<textarea>`, etc.) que já são interativos por padrão e já tem funções e estados que comunicam com as tecnologias assistivas.

Da mesma forma que o foco entre elementos deve ser sequencial e lógico, também a sua identificação (*focus ring/outline*) deve ser destacável e visualmente identificável. Este campo é geralmente identificado com uma borda fina no elemento em questão, quando o mesmo está ativo ou é pressionado e deve-se mover visivelmente entre elementos quando ativos.

Para além do facto de existir uma sequência lógica de navegação por teclado ser uma boa prática a nível de acessibilidade há uma enorme variedade de utilizadores que beneficia também deste feito:

- Utilizadores com incapacidades visuais, que usam leitores de ecrã ou ampliadores de ecrã;
- Utilizadores com destreza manual limitada, que preferem usar o teclado em vez do rato como periférico habitual;
- Utilizadores que só navegam nas páginas e no computador através da tecla *tab*.

Se a ordem dos elementos interativos não for uma sequência lógica, então pode acontecer que:

- Um utilizador de ecrã pode não conseguir encontrar o *link* para os interesses ou perfil pessoal;
- Um utilizador com ampliação de ecrã pode não conseguir orientar-se na página;
- Um utilizador que navega apenas com o teclado pode achar difícil saber qual é o elemento que tem o foco se ele não seguir a ordem lógica do *layout* da página.

Finalmente, o uso da navegação por teclado, em particular o uso da tecla *tab* é importante para os formulários. Os utilizadores esperam que a ordem de navegação de elementos de um formulário seja a mesma que a ordem que os elementos são apresentados no ecrã. Quando o último campo de um formulário for preenchido, os utilizadores pressionam a tecla *tab* no último elemento e o foco fica automaticamente nos botões de submeter o formulário (*submit*). No mundo *e-commerce* o uso da navegação por teclado também pode ser de interesse dos utilizadores que fazem compras *online* ou transações bancárias. Com uma boa navegação lógica por teclado, estes utilizadores beneficiam de não ter que alternar entre periféricos, isto é, entre o teclado e o rato e conseguem assim um fluxo lógico e natural.

### **5.1.6 Estrutura da página web**

A estrutura de uma página *web*, também conhecida como *page structure* é muito importante que seja bem pensada e organizada, pois permite a navegação e o processamento da informação mais eficientes. Com o uso do HTML e dos elementos ARIA apresentados (cf. secção 5.1.4) é possível melhorar a navegação e a orientação em páginas *web* e aplicações.

A estrutura de uma página *web* divide-se em quatro grandes áreas, que vão ser abordadas nas secções seguintes:

- *Page regions* / regiões da página;

- *Labeling regions* / identificar e nomear regiões;
- *Headings* / cabeçalhos;
- *Content structure* / estrutura do conteúdo.

O facto de uma aplicação *web* ter o conteúdo das páginas bem estruturado é importante, no sentido em que afeta muitos grupos de pessoas que podem beneficiar com isto, como:

- Pessoas com incapacidades cognitivas e de aprendizagem podem encontrar e priorizar mais facilmente o conteúdo da página;
- Pessoas que utilizam leitores de ecrã e que conseguem alternar entre o conteúdo principal as restantes secções que são importantes para elas;
- Os utilizadores da navegação por teclado que podem navegar nas páginas e nas secções com mais eficiência. Caso contrário, os utilizadores precisariam de pressionar a tecla *tab* múltiplas vezes para navegar por todos os *links* e elementos interativos em cada secção;
- Os utilizadores que utilizam *software* auxiliar que mostra apenas o conteúdo principal de uma página da *web*, como pessoas com incapacidades cognitivas, vão receber melhores resultados se a estrutura da página estiver organizada da melhor forma e corretamente;
- Pessoas com incapacidades visuais, incluindo pessoas com um nível de visão baixa, conseguem receber dicas e conteúdo auditivo que fornece orientação na página e no conteúdo;
- Os utilizadores de dispositivos móveis geralmente têm acesso ao modo de "leitura" incorporado nesses dispositivos, que irá mostrar o conteúdo principal da página, se este estiver marcado e organizado de forma correta;
- Pessoas que usam determinados *plugins* externos associados ao *browser* que podem usar as funções dos *landmarks* para ir para secções específicas de uma página.

Quem mais beneficia com o uso correto de uma boa estrutura de páginas são as pessoas com incapacidades que utilizam as tecnologias assistivas. No entanto, os mecanismos de pesquisa (*SEO*) também podem usar estes dados e estruturá-los de forma a conseguir indexar da melhor forma o conteúdo de uma página, para ser possível obter melhores resultados nas pesquisas.

### 5.1.7 Page regions

Numa página *web* deve sempre estar presente as várias *page regions* existentes, para que os *browsers* consigam identificá-las através das tecnologias assistivas. As *page regions* são as seguintes:

- *Header* – a maioria dos *websites* tem uma região no topo de cada página que contém informações de todo o *website*, como o logotipo, a área de pesquisa e a área de navegação. O elemento HTML5 nativo existente para identificar esta região da melhor forma é a *tag* `<header>`. Se este elemento for descendente das *tags* `<article>` ou `<section>`, o elemento `<header>` não fica associado à página inteira, mas apenas aquela região;
- *Footer* – de forma semelhante ao *header*, a maioria dos *websites* tem uma região na parte inferior de todas as páginas que contém informações de todo o *website*, como informações sobre contactos, direitos de autor, declarações de privacidade, entre outros. O elemento HTML5 nativo existente para identificar esta região da melhor forma é a *tag* `<footer>`. Se este elemento for descendente das *tags* `<article>` ou `<section>`, o elemento `<footer>` não fica associado à página inteira, mas apenas aquela região;
- *Navigation* – uma página *web* pode conter diversos elementos de navegação presentes que devem ser identificados com nomes acessíveis com recurso ao uso do atributo ARIA *aria-label*, com a finalidade de ser possível identificar cada menu de navegação. O elemento HTML5 nativo existente para identificar esta região da melhor forma é a *tag* `<nav>`;
- *Main* – uma página *web* deve conter apenas uma região *main* com a finalidade de identificar o conteúdo principal de uma página *web*. O elemento HTML5 nativo existente para identificar esta região da melhor forma é a *tag* `<main>`;
- *Complementary* – nas páginas *web* pode existir a necessidade de identificar regiões que suportam o conteúdo principal e que são secções separadas e com significado próprio, como é o caso de algo anotação ou nota lateral para explicar o conteúdo principal. O elemento HTML5 nativo existente para identificar esta região da melhor forma é a *tag* `<aside>`.

### 5.1.8 Labeling regions

Ao longo do presente documento já tem sido abordado que é necessário e uma boa prática providenciar formas de distinguir duas regiões do mesmo tipo, como é os casos das navegações existentes no projeto, como a navegação para as páginas e a navegação para o carrinho de compras, lista de desejos e área de pesquisa. Em alguns casos é necessário alterar a identificação padrão das regiões das páginas, por exemplo, identificar uma região do tipo `<aside>` como “Informação complementar”. No entanto, há algumas regiões exclusivas como o `<main>`, `<footer>` e o `<header>` que não precisam de identificação adicional, podendo até levar a que os utilizadores fiquem confusos pois é um comportamento fora do que é considerado habitual.

No sentido de providenciar informação adicional às regiões das páginas *web* existem duas abordagens possíveis – *aria-label* e *aria-labelledby* (cf. secção 5.1.4).

### 5.1.9 Headings

Os *headings* ou cabeçalhos comunicam a forma como o conteúdo da página está organizado com a finalidade de que os *browsers*, *plug-ins* e tecnologias assistivas possam obter informações de como navegar numa página *web*.

A organização de páginas da *web* por títulos ajuda os utilizadores a identificar da melhor forma a organização e estrutura da página. Visualmente, os títulos são apresentados como maiores e mais distintos do que o texto ao redor. O facto de os títulos, por defeito, aumentarem o texto dos elementos ajuda a guiar os olhos pela página, de forma a perceber o propósito da mesma mais rapidamente. Assim, usar *headings* e melhorar a sua aparência visual é especialmente útil para utilizadores com incapacidades cognitivas.

Se os *headings* forem usados corretamente numa página *web*, os utilizadores dos leitores de ecrã também beneficiam do seu uso. Estes utilizadores podem navegar nas páginas através dos seus títulos, a ouvir a lista de todos os títulos e avançar para um título desejado para começar a ler nesse ponto. Os utilizadores de leitores de ecrã podem usar estes títulos para avançar blocos repetidos de conteúdo, com menus ou barras laterais com informação complementar, por exemplo.

A par disto, em 2017, a organização *WebAIM*<sup>13</sup> perguntou aos utilizadores de leitores de ecrã como é que preferiam encontrar informações nas páginas *web* com bastante conteúdo. Quase 70% dos entrevistados disseram que preferiam usar os títulos da página para avançar e recuar entre o conteúdo (*WebAIM*, 2017). Com isto, é notório que, organizar as páginas com o recurso a *headings* é uma das melhores estratégias de acessibilidade disponíveis.

A estrutura de títulos das páginas *web* deve retratar de forma visual a sua relação, para que as pessoas possam ver a estrutura e os leitores de ecrã consigam identificar a estrutura para lê-la e identificar as relações entre elementos. Com isto, os utilizadores que conseguem ver a página decidem, em segundos, qual é o conteúdo que é ou não relevante para eles, ao ver os títulos, os subtítulos, e mantendo-se atento às palavras-chave significativas presentes no conteúdo. Por outro lado, as pessoas que no seu quotidiano usam os leitores de ecrã conseguem beneficiar com páginas que tem o conteúdo organizado através do uso de *headings*, pois conseguem facilmente navegar nesta.

Sem uma estrutura de títulos correta, os utilizadores são obrigados a fazer um esforço extra para encontrar as informações que estão à procura e isso não é o mais correto. Do ponto de vista da acessibilidade, a estrutura de títulos no código deve estar alinhada com a apresentação visual e deve fazer sentido como um “índice” para a página.

Os títulos são classificados desde o nível `<h1>` a `<h6>` e devem ser usados de forma hierárquica, com o `<h1>` a representar a ideia mais importante da página, e as subsecções organizadas com títulos de nível `<h2>`. Essas subsecções podem ser divididas com *tags* `<h3>` e assim por diante.

Na figura seguinte é possível verificar a relação dos títulos e das suas subsecções:

---

<sup>13</sup> *WebAIM – Web accessibility in mind*

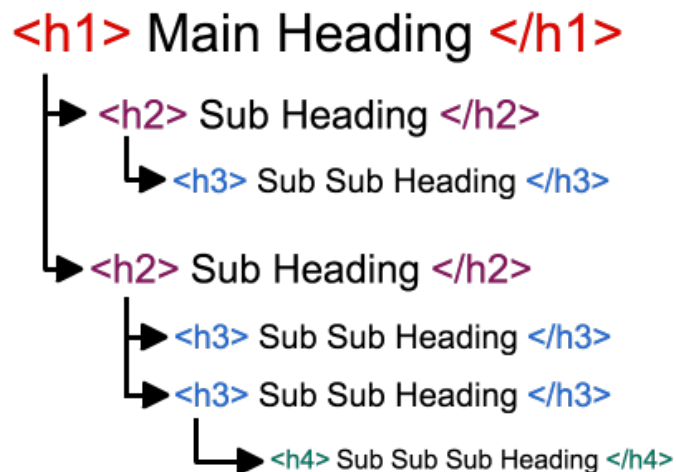


Figura 23 - Relação de *headings* e de respetivas subsecções (Nomensa)

Existem algumas recomendações/boas práticas relacionadas com *headings* que devem persistir no desenvolvimento de um *website*:

- A estrutura de títulos deve ser planeada antes de começar a desenvolver a página, com o objetivo de selecionar os níveis apropriados de títulos e manter a estrutura organizada no geral;
- Todas as páginas devem ter pelo menos um título de nível `<h1>` que forneça o título da página. Este título deve ser sempre um `<h1>` e a maioria dos utilizadores de leitores de ecrã preferem que apenas este elemento seja um `<h1>` (WebAIM, 2017), com o propósito de tudo ser mais objetivo;
- Um equívoco comum existente é que o `<h1>` deve ser o primeiro título na página, no entanto é normal e boa prática utilizar títulos nas *tags* `<section>` e `<article>`, que podem ocorrer antes do título do documento, como menus e informação lateral e complementar;
- Não se deve avançar entre níveis de títulos (avançar de `<h2>` para `<h5>`), no entanto é permitido avançar títulos na direção contrária (avançar de `<h5>` para `<h2>`) se a página assim o exigir. Do ponto de vista da acessibilidade, as pessoas que usam leitores de ecrã geralmente dependem da navegação por títulos, portanto, se a estrutura não for hierárquica, é possível que não entendam o relacionamento dos elementos;
- Os títulos não devem ser escolhidos com base na sua aparência visual, mas sim com base em que título é apropriado para determinada hierarquia;
- Não se forçar a aparência de um elemento para negrito em vez de utilizar um título. Um dos erros mais comuns de acessibilidade é tornar o texto em negrito quando um título é necessário. Embora o texto possa parecer um título, o código subjacente não está definido corretamente e os utilizadores dos leitores de ecrã não serão beneficiados;

- Não se deve usar títulos em excesso. Na maioria dos casos, o máximo que as páginas *web* precisam é de títulos `<h2>` e de forma ocasional títulos `<h3>`, em último caso e só em páginas excepcionalmente longas ou complexas é que é necessário o uso de `<h5>` e `<h6>`;
- Quando o assunto é sobre uma SPA<sup>14</sup> (*single page application*), como é o caso do presente projeto, o *focus* (cf. secção 5.1.11) do *browser* deve ser definido no cabeçalho `<h1>` após a página e o seu conteúdo carregar. Isto é, de cada vez que existe uma mudança de rota, o *focus* deve ir diretamente para o título da página presente na *tag* `<h1>`. Com esta técnica é possível evitar a repetição da leitura dos elementos que estão antes do `<h1>` como é o caso do *header*, que se encontra no topo da página e é repetido em todas elas;
- O planeamento é fundamental e, por isso, os desenvolvedores e os *designers* também devem organizar as páginas com recursos a *landmarks* (cf. secção 5.1.3). De acordo com a pesquisa da WebAIM de 2017, mais de metade dos utilizadores de leitores de ecrã usam esta funcionalidade. Alguns *landmarks*, como *complementary*, *navigation* e *regions*, devem possuir atributos do tipo *aria-label*. Se um título estiver presente, pode ser usado o atributo *aria-labelledby* para associar o título visual ao *landmark* a que pertence.

### 5.1.10 Content structure

Como já foi dito, quando se está a desenvolver um *website* é preciso ter atenção ao uso dos elementos nativos que acrescentam valor a nível de semântica, para que o *website* possa ser extensível. Esta semântica, quando é válida cria conteúdo reutilizável e mais com mais significado para as tecnologias assistivas. Assim, existem alguns elementos que ajudam a criar uma boa estrutura semântica no conteúdo das páginas *web*:

- **Articles** - O elemento nativo HTML5 `<article>` representa um *container* independente de uma página *web*. No projeto a *tag* `<article>` é utilizada para expor cada produto na lista de produtos;
- **Sections** - O elemento nativo HTML5 `<section>` define uma região geral de uma página *web* e é útil para agrupar o conteúdo por temas. No projeto a *tag* `<section>` é utilizada como *container* para todos os produtos na página de lista de produtos;
- **Paragraphs** – O elemento HTML `<p>` serve para identificar parágrafos de texto e devem ter um estilo consistente ao longo da aplicação para melhorar a legibilidade do texto, conseguindo proporcionar aos utilizadores mais controlo;
- **Lists** – Existem vários tipos de listas que permitem organizar e agrupar a informação de acordo com o seu conteúdo com o objetivo de fornecer orientação aos utilizadores. Os itens pertencentes a listas podem conter uma variedade de elementos

---

<sup>14</sup> *Single page application* – um *website* que renderiza o seu conteúdo em resposta a ações da navegação, com recurso a pedidos assíncronos.

HTML, como parágrafos, títulos, elementos de formulário, entre outros. Assim, existem vários tipos de listas:

- Listas não ordenadas são usadas quando a ordem dos itens não é relevante. Este tipo de lista consiste no uso do elemento `<ul>` e contém vários elementos descendentes do tipo `<li>`;
  - Listas ordenadas são usadas para apresentar informações sequenciais que são automaticamente enumeradas pelo *browser* e consistem no uso de um elemento `<ol>` e de vários elementos descendentes do tipo `<li>`;
  - Listas de descrição `<dl>` são grupos de termos e descrições relacionados que estão conectados. Cada grupo associa um ou mais termos (conteúdo das *tags* `<dt>`) a uma ou mais descrições (conteúdo dos elementos `<dd>`).
- **Quotes** – Identificar uma citação ajuda a esclarecer que o conteúdo é de outro autor. Com o uso dos elementos `<blockquote>` ou `<q>` as tecnologias assistivas conseguem transmitir aos utilizadores onde é que uma citação começa e termina, o que ajuda a evitar mal-entendidos. O elemento `<blockquote>` deve ser usado para citações mais longas e complexas e pode conter parágrafos, títulos e outros elementos da estrutura de texto e pode ser usado em conjunto com o elemento `<cite>` que serve para referir a fonte da citação. Para citações mais curtas, que geralmente estão contidas noutras frases, deve ser usado o elemento `<q>`;
  - **Figures** – Figures ou figuras são blocos com informações adicionais acerca do conteúdo principal da página. O conteúdo normalmente são listas, imagens ou tabelas. Cada figura é envolvida num elemento `<figure>` e identificada usando um elemento descendente do tipo `<figcaption>`;
  - **Images** – Estes elementos são úteis para criar compromisso visual para muitos utilizadores normalmente surgem acompanhados de texto alternativo (cf. secção 5.1.12) que ajuda a clarificar o propósito de uma imagem para pessoas com incapacidades de leitura;
  - **Tables** – Estes elementos fornecem uma forma útil de exibir os dados para que a estrutura da tabela possa ser comunicada a todos, quando acompanhado por um *layout* adequado que permita às pessoas entender os relacionamentos entre dados.

### 5.1.11 Focus e outline

A navegação por teclado é um dos aspetos mais importantes da acessibilidade na *web* e defende que todo o conteúdo deve ser possível aceder através do uso do teclado. Há muito tipos de incapacidades que podem beneficiar com o uso do teclado, como é o caso dos utilizadores com incapacidades visuais, utilizadores com incapacidades motoras, pessoas que sofreram algum tipo de acidente e que tenham ficado com os membros amputados e pessoas que usem teclados auxiliares, que não os tradicionais e que imite algumas funcionalidades de outros teclados.

Uma das maneiras mais úteis para os utilizadores saberem em que parte do *website* estão é o uso de indicadores do foco, também conhecido como *focus*.

Um utilizador que navegue por teclado geralmente usa a tecla *Tab* para navegar por elementos interativos das páginas - *links*, botões, campos para inserir texto de formulários, etc. Quando um elemento tem o "foco" do teclado, ele pode ser ativado ou manipulado com o teclado.

A par disto é importante que um utilizador de navegação por teclado, consiga perceber e ver um indicador visual do elemento que atualmente tem o foco do teclado. Este indicador de foco é simples e é fornecido automaticamente pelo *browser* e é normalmente mostrado como se fosse um contorno em torno do elemento que está com *focus*.

A boa prática associada ao *focus* dos elementos é relacionada com uma propriedade CSS denominada *outline*, que não deve ser ocultada. O uso da regra *outline:0* ou estilos semelhantes deve ser evitado porque este tipo de alterações impossibilita os elementos de receberem *focus* de forma visual, isto é, não fornecem qualquer tipo de informação visual acerca de que elemento é que está selecionado no momento.

O que pode ser feito além do comportamento padrão, é melhorar os estilos CSS para tornar o indicador de *focus* mais agradável a nível visual e com uma melhor experiência para o utilizador, tornando possível adicionar cores de fundo ou outro estilo à regra *outline*, conseguindo assim que este contorno consiga receber estilos CSS que correspondam ao *design* da página.

No caso do presente projeto, quando o utilizador está a usar da forma padrão o *website*, ou seja, com recurso ao rato como periférico, são retirados os estilos da regra *outline* com o intuito de não distrair o utilizador e não poluir o *layout*. Por outro lado, quando é detetado que o utilizador está a navegar com recurso a teclado, os estilos CSS para o indicador de *focus* são repostos, isto é, o indicador de *focus* é mostrado devidamente. Em seguida é possível ver os estilos que são reescritos quando é detetado que um utilizador está a navegar no *website* com a forma padrão:

```

<style>
  :focus {
    outline:0!important;
    box-shadow:none!important
  }

  ::-moz-focus-inner{
    border:0!important;
  }

  input:focus + label,
  input:active + label{
    outline:0!important;
    box-shadow:none!important
  }

  input:focus + label::after,
  input:active + label::after{
    outline:0!important;
    box-shadow:none!important
  }
</style>

```

Figura 24 - Estilos incluídos através do *Javascript* para retirar o *outline* a navegar com o rato

Nas figuras seguintes é possível verificar o aspeto visual do mesmo componente, neste caso o *link* para a *Homepage*. A seguinte imagem ilustra um *link*, aquando da utilização da navegação através do teclado, usando a tecla *tab*.

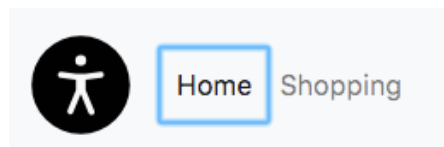


Figura 25 - *Link* para *homepage* quando é utilizada a navegação através do teclado

Por fim, o facto de ter o *focus* visível faz parte de um critério de sucesso das diretrizes *WCAG* (cf. secção 2.3.2) que defende que qualquer interface de utilizador tem de conter um modo de funcionamento que seja possível navegar com o teclado e quando isso acontece o *focus* do elemento tem de estar visível. Ao cumprir este critério de sucesso é também cumprida uma das bases para ser possível atingir o nível *AA* de acessibilidade (cf. secção 2.3.2).

### 5.1.12 Alternative text

A técnica *alternative text*, texto alternativo em português, é o primeiro princípio da acessibilidade na *web*, no que diz respeito às diretrizes *WCAG* (cf. secção 2.3.2) e trata-se de um dos princípios que são mais difíceis de implementar adequadamente quando o caso se trata de imagens dinâmicas (sem controlo por parte do desenvolvedor).

As páginas *web* estão repletas de imagens com texto alternativo em falta, incorreto ou insatisfatório, no entanto, como muitas coisas na acessibilidade da *web*, a determinação de textos alternativos apropriados é uma das formas de dar melhor contexto e complemento a

uma imagem, conseguindo assim fornecer uma melhor interpretação por parte dos utilizadores. Desta forma, a premissa básica do texto alternativo é fazer com que todos os utilizadores possam ler as mesmas informações e ter as mesmas interações com uma página da *web*, independentemente de poderem ou não ver as imagens. O texto alternativo é frequentemente usado por utilizadores com incapacidades visuais com recurso a leitores de ecrã, onde a descrição de uma imagem é lida no seu lugar.

Posto isto, providenciar um texto alternativo pode ser feito de várias formas:

- Texto alternativo no atributo *alt* da imagem – este texto pode ser armazenado no atributo *alt* da tag `<img>` e deve ser uma breve descrição que informa o conteúdo da imagem ou as informações que esta quer transmitir;
- Um valor vazio no atributo *alt* – se o intuito de uma imagem for meramente decorativo, o atributo *alt* pode ser definido com um conteúdo vazio, desta forma: `alt=""`. Assim, é possível que os utilizadores de leitores de ecrã consigam avançar a imagem e não saber nada sobre ela, o que é bom no sentido em que não são obrigados a receber informação de elementos que apenas são decorativos e que não apresentam grande valor para um utilizador com incapacidades visuais;
- Recurso ao atributo ARIA *aria-hidden* – com o mesmo intuito que o ponto anterior, adicionar o atributo `aria-hidden="true"` permite que os utilizadores que usam leitores de ecrã consigam avançar a imagem quando a mesma não apresenta valor, a não ser decorativo, para a perceção do *website*;
- Recurso a elementos auxiliares – muitas vezes, a função ou a informação que uma imagem está a transmitir está presente nos elementos de texto da página em volta da imagem. Nestes casos, o atributo *alt* pode ser algo mais simples e assim, o utilizador de leitores de ecrã sabe que a imagem está presente e sabe o que é, sendo capaz de obter as informações que está a receber do contexto fornecido pelos elementos auxiliares.

Isto significa que o atributo *alt* não é o único mecanismo para fornecer o conteúdo e a função da imagem. Estas informações também podem ser fornecidas nos elementos de texto adjacentes à imagem ou na página que contém a imagem.

O uso de um bom texto alternativo através do atributo *alt* serve vários propósitos:

- Fornecer informações alternativas para uma imagem se um utilizador, por algum motivo, não puder visualizá-la (devido a uma conexão lenta, a um erro no atributo *src* ou se usar um *screen-reader*);
- Fornecer um melhor significado semântico à página e uma melhor descrição para as imagens que podem ser lidas por mecanismos de pesquisa (*SEO*).

O princípio fundamental que diz respeito aos textos alternativos é que os leitores de ecrã não conseguem analisar uma imagem e determinar o que a imagem apresenta, sem recurso a estas ajudas. Como desenvolvedores e marcas que defendem o uso de acessibilidade nos seus projetos, o texto alternativo deve ser fornecido ao utilizador para providenciar uma melhor interpretação das imagens que sejam consideradas relevantes nos *websites*.

No projeto, nas imagens que são estáticas foi possível providenciar uma descrição que garanta uma boa interpretação, no entanto nas imagens que são dinâmicas, isto é, que são providenciadas através de uma resposta da API, não é possível mudar o texto alternativo porque não é possível identificar que imagem é que vai estar em que elemento, no sentido de conseguir melhorar a sua descrição e isto levaria a problemas de manutenção e de escalabilidade no sentido de conseguir expandir esta solução para todas as imagens dinâmicas de um *website*. Nesse sentido, existem duas soluções que conseguem ajudar a resolver este problema:

- Conseguir descrever melhor as imagens no momento da sua criação (para conseguir obter melhores resultados nos pedidos à API);
- Conseguir utilizar plataformas externas de leitura e descrição de imagens.

### 5.1.13 Nomes acessíveis

Um nome acessível é o nome de um elemento de uma página web e o valor que lhe está associado é comunicado aos utilizadores de tecnologias assistivas. Sem a correta implementação desta técnica as pessoas que dependem dessas tecnologias podem ter dificuldade em entender ou interagir com grande parte do conteúdo da página

Muitos elementos HTML possuem um nome acessível, que pode derivar de diferentes fontes e é exposto na árvore de acessibilidade (cf. secção 2.3.4) do *browser*. Esta informação é usada pelas tecnologias assistivas para identificar o elemento. Assim, os utilizadores que usam os leitores de ecrã conseguem ouvir o nome acessível que foi atribuído ao elemento quando o foco muda para esse elemento.

Os nomes acessíveis dos elementos podem derivar de várias formas:

- Conteúdo do elemento;
- Atributo associado ao elemento (atributo *alt*, atributos ARIA *aria-label* e *aria-labelledby*).

O nome acessível para um elemento pode ser alterado usando os atributos ARIA (cf. secção 5.1.4). Os atributos [aria-label](#) e [aria-labelledby](#) funcionam de maneiras diferentes, mas o importante a lembrar é que o ARIA supera tudo, isto é, o nome acessível fornecido com o HTML nativo será substituído pelo nome acessível fornecido com os atributos ARIA.

Em suma, os nomes acessíveis devem ser escolhidos com cuidado, no caso de existirem dois elementos numa página que possuam exatamente a mesma responsabilidade, o nome acessível deve ser o mesmo, como é o caso de dois *links* que direcionam exatamente para a mesma página. Por outro lado, se existirem seis botões numa página que possuem responsabilidades diferentes, cada um deve ter um nome acessível único e distinto dos restantes.

#### 5.1.14 Contraste de cores

Alguns utilizadores têm dificuldade em perceber o texto se houver muito pouco contraste entre este e o plano de fundo. O cuidado com as cores e o contraste das mesmas deve ser abordado desde o início dos projetos, ainda na primeira fase que envolva aspetos visuais como *layout* e *design*.

Ao projetar o esquema de cores de um *website* é importante certificar se há contraste suficiente entre os elementos das páginas. Tanto no processo de *design* como no processo de implementação há ferramentas que podem ser usadas para verificar o contraste entre duas cores, como:

- *Colour contrast analyser* - permite verificar o contraste entre duas cores. Estas duas cores podem ser inseridas através dos seus valores hexadecimais. Esta ferramenta fornece informações sobre o contraste entre essas cores, incluindo classificações de aprovação ou reprovação para vários os níveis de conformidade com as diretrizes WCAG (Connor, 2014);
- *WebAIM Color Contrast Checker* – permite ter acesso às mesmas funcionalidades que o *Colour contrast analyser* e tem ainda uma funcionalidade que permite “clarear” e “escurecer” as cores adicionadas até obter uma aprovação de bom contraste entre cores (Connor, 2014).

Para além destas ferramentas, existem ainda ferramentas de auditoria de acessibilidade onde é possível verificar quanto é que um *website* é acessível, tendo em conta o contraste das cores. No caso do projeto, nas ferramentas de auditoria incorporadas no *browser Chrome* o nível de contraste das cores do projeto é satisfatório e as cores são perceptíveis.

#### 5.1.15 Tratamento de erros

Fornecer *feedback* ao utilizador, quer seja proveniente de uma ação na página (adicionar um produto ao carrinho de compras), ou do envio dos dados de um formulário é muito importante a nível de acessibilidade. Este *feedback* pode ser realizado de várias formas, sendo normalmente apresentado com recurso a elementos visuais e cores (o que não é necessariamente o mais correto), perto dos elementos de submissão ou dos botões. O problema neste campo é que também estes elementos precisam de ser acessíveis para conseguirem providenciar a informação correta aos utilizadores de tecnologias assistivas.

Estes elementos que representam os erros, também chamados de notificações, devem de ser concisos e claros. Neste caso particular, as mensagens de erro devem ser fáceis de entender e fornecer instruções simples sobre como o erro pode ser resolvido. Da mesma forma, e introduzindo um novo conceito, as mensagens de sucesso também devem existir após estas ações serem bem-sucedidas, com o objetivo de anunciar a confirmação de uma conclusão de tarefa.

Utilizar os atributos ARIA *role="alert"* e *aria-live* é uma técnica e boa prática para identificar erros nas páginas *web* e satisfaz o critério de sucesso 3.3.1 – Identificação de erros, das diretrizes WCAG (cf. secção 2.3.2) que satisfaz o nível A de acessibilidade. Esta diretriz defende que quando um erro é detetado automaticamente, o elemento que está com erro

deve ser identificado e o erro deve ser descrito para o utilizador conseguir entender como deve agir perante a página. Assim, o objetivo desta técnica é notificar os utilizadores, através das tecnologias assistivas, de que um erro na página aconteceu. O atributo `aria-live` possibilita que uma TA (como o leitor de ecrã) seja notificada quando as mensagens de erro são injetadas nas páginas, conseguindo que o conteúdo da ARIA *region* `aria-live` seja lido automaticamente pelo leitor de ecrã, sem que o mesmo tenha de estar no elemento que possui o atributo ARIA (W3C).

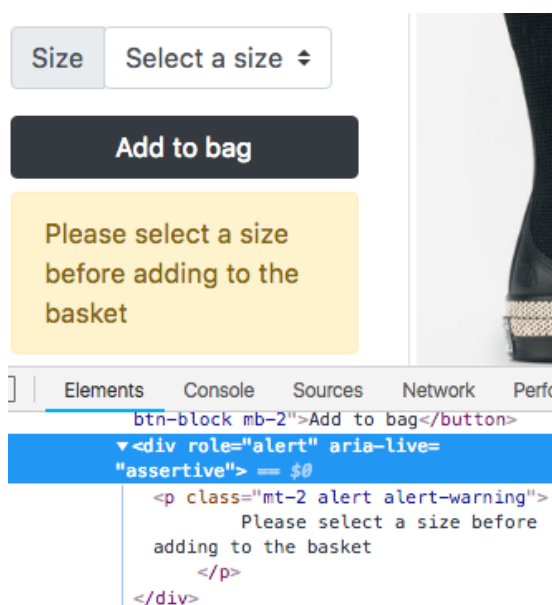


Figura 26 - Tratamento de erro ao adicionar um produto ao carrinho de compras

No exemplo acima, retirado do projeto, é possível verificar que o facto de não existir um tamanho selecionado quando se tenta adicionar ao carrinho de compras, resulta num erro. Este erro não só é identificado visualmente, com também pelas tecnologias assistivas, com o recurso à técnica mencionada acima: uso do `role="alert"` e `aria-live="assertive"`.

## 5.2 Componentes

Nesta secção vão ser abordados todos os componentes existentes no projeto. Os componentes aqui descritos podem ser componentes criados no projeto que já foram alvos de isolamento e partilha com a comunidade ou que ainda vão ser partilhados, ou então componentes que já existiam na comunidade e vão ser reutilizados. Tal como já foi referido na secção 4.2 e de acordo com a metodologia *atomic design*, utilizada no projeto, os componentes podem ser vistos como *packages* ou átomos.

### 5.2.1 Skip link

O conteúdo principal geralmente não é a primeira coisa apresentada numa página *web*. Os utilizadores de leitores ecrã e de navegação via teclado geralmente precisam navegar por uma lista longa de navegação com sub-listas de *links*, a pesquisa do *website* e outros elementos

antes de chegar ao conteúdo principal. Este comportamento é particularmente difícil para utilizadores com incapacidades visuais ou motoras, tendo como conclusão que é impraticável exigir que os utilizadores executem qualquer ação repetidas vezes antes de chegar ao conteúdo principal.

É claro que pessoas com visão que usam o rato como periférico de movimento para navegar na página não apresentam estes problemas. Estes utilizadores conseguem navegar na página e avançar/ignorar o conteúdo que quiserem de forma fácil conseguindo avançar diretamente para um *link* desejado. E é ao observar este comportamento que surgem os *Skip links*, com a ideia de ignorar a navegação ou conteúdo para dar aos utilizadores de teclados e tecnologias assistivas a mesma capacidade de ir diretamente para o conteúdo principal que os utilizadores com visão que usam periféricos como o rato.

Neste projeto foi criado um componente denominado *Skip Link* que tem como propósito uma ideia bastante simples: fornecer um *link* no topo da página que torna possível o utilizador navegar para outra parte/alvo do *website*, como o conteúdo principal ou o *footer*. Na maior parte, não é uma tarefa complexa à partida, embora exista mais de uma maneira de atingir o objetivo. Para criar este componente foram pensados alguns requisitos/funcionalidades que o mesmo tinha de satisfazer:

- Fornecer *skip links* no topo das páginas;
- Fornecer *skip links* em outros lugares úteis ao longo das páginas;
- Tornar os *skip links* invisíveis até receber o foco do teclado.

Este componente foi criado no âmbito do projeto, já foi isolado num repositório partilhado com o objetivo de promover a partilha e já está a ser usado noutros projetos que não este.

Este componente foi feito para aplicações *React* e é composto por três módulos principais:

- *Identifiers* – objeto que fornece um conjunto de identificadores padrão (mais usados) nas aplicações. O principal objetivo deste módulo é providenciar *ids* comuns entre o *Skip Link* e o *Target Element* (*main*, *footer*, etc) para que os dois tenham o mesmo *id* como referência;
- *SkipLink* – componente com as propriedades necessárias para redirecionarem para um elemento *TargetElement*, via *href*;
- *TargetElement* – componente que renderiza uma *tag HTML* específica como o *id* do *Skip Link* para o qual quer redirecionar o utilizador e já possui os atributos necessários de acessibilidade (*tabindex* e *id*).

Para este componente ser renderizado no projeto é preciso fornecer ao mesmo algumas propriedades obrigatórias, como o *children* (conteúdo que o componente deve apresentar, como "*Skip to main content*") e o *to* (que é um *id* de um elemento para o qual se pretende avançar).

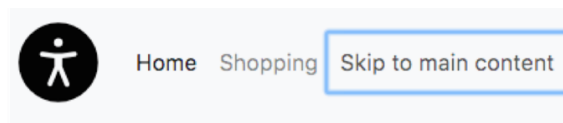


Figura 27 – Resultado visual do uso do componente *Skip Link*

Nas imagens acima é possível verificar a utilização do componente no código e o resultado visual final na aplicação. O componente mostrado nas imagens é usado na página *Homepage* e serve para permitir ao utilizador navegar rapidamente para áreas importantes do *website*: *main*, *banner* e *footer*.

### 5.2.2 Visually Hidden

Os desenvolvedores normalmente usam a regra de CSS `display: none` para esconder conteúdo específico na página. Infelizmente, esta ação simples e comum pode ser problemática para os leitores de ecrã e deve ser repensada.

Existem situações do mundo real nas quais é necessário esconder certos elementos visualmente (por exemplo, a descrição de um formulário), mas manter o texto do elemento disponível para ser anunciado pelos leitores de ecrã. Para fazer isto existe um padrão chamado *clip-pattern* que possibilita este comportamento - ocultar o conteúdo visualmente e ainda fornecer o conteúdo para os leitores de ecrã. Assim, surge a necessidade de criar o componente *Visually Hidden* no projeto.

Mesmo que o foco principal deste componente seja esconder o conteúdo, se este for *focusable* deve ser mostrado quando o mesmo recebe o estado *focus* da aplicação. Caso contrário, um utilizador que navegue através do teclado pode não saber qual é o elemento que está atualmente em *focus*.

Este componente foi criado no âmbito do projeto, já foi isolado num repositório partilhado com o objetivo de promover a partilha e já está a ser usado noutros projetos que não este. O mesmo foi feito para aplicações *React* e tem como objetivo ocultar visualmente o conteúdo de elementos HTML, mas mostrar o seu conteúdo aos leitores de ecrã. Em termos de acessibilidade, é necessário fornecer texto / elementos para os leitores de tela que estejam visualmente ocultos. É o oposto do atributo `aria-hidden` e é extremamente útil para fornecer informações específicas, que não são tão úteis na lógica da visão. Um bom caso exemplo é o preço com desconto de um determinado produto - normalmente, o leitor de ecrã anuncia o preço original e o preço de saldo, de forma seguida e sem qualquer indicação de qual é o preço atual e o preço original de venda. Desta forma, o componente *VisuallyHidden* permite adicionar estas informações adicionais sem “poluir” a interface visual.

Com isso, os elementos que podem receber o estado de *focus*, ficam ocultos visualmente por defeito e aparecem quando os utilizadores navegam até eles usando a tecla *tab* do teclado.

Para este componente ser renderizado no projeto é preciso passar para o mesmo a propriedade *children* (conteúdo que o componente deve esconder visualmente, como *“Price”* ou `<a href="#next-product">Skip to next product</a>`), que é obrigatória.



Figura 28 – Resultado visual do uso do componente *VisuallyHidden* num elemento *focusable*

Na imagem acima é possível verificar o resultado visual final na aplicação quando se usa o mesmo componente num elemento *focusable*, isto é, que pode receber o estado *focus*. O componente mostrado nas imagens é usado na lista de produtos, em cada produto para anunciar informação adicional para o preço e para providenciar uma forma rápida de ir para a página de produto, escondendo o conteúdo do *link* da mesma, que aparece se o utilizador estiver a navegar por teclado.

### 5.2.3 Botão

As ações nos *websites* constituem uma das partes mais importante, uma vez que aparecem diversas vezes, quer como botão de submissão de um formulário, quer para preencher uma morada e guardá-la, quer para as ações associadas a um produto, como adicionar ao carrinho de compras.

Este componente já existia num repositório partilhado com objetivo de promover a sua reutilização e manutenção. Neste projeto, o mesmo foi reutilizado para as diversas ações contidas no *website* e foi alvo de pequenas mudanças para melhorar a sua acessibilidade. Este componente foi feito para aplicações *React* e é composto por algumas propriedades:

- *component* – *tag* HTML ou componente que irá ser renderizado. Por padrão é a *tag* `<button>` que é renderizada;
- *loading* – propriedade isolada que define o estado de *loading* do botão. Se o botão estiver com o estado *loading* são adicionados alguns atributos automaticamente como o `tabindex="-1"` e o `aria-busy="true"`;

Este componente já inclui algumas propriedades de acessibilidade, no entanto, outras devem ser passadas aquando do uso do componente, como o uso do *aria-Label* para providenciar mais contexto para os leitores de ecrã.

### 5.2.4 Product card

O componente *Product Card* surgiu da necessidade de apresentar as informações de um produto, múltiplas vezes numa lista de produtos e possui todos os elementos que suportam a

renderização das informações do produto como nome, descrição, preço original, preço em saldo, marca, entre outros e deve ainda conter os cuidados de acessibilidade.

Foi com o foco nestas necessidades que surgiu o componente *ProductCard* que recebe como propriedade as informações do produto e renderiza as mesmas de forma lógica. O componente foi criado no âmbito do projeto, mas ainda não foi isolado num repositório partilhado e ainda só é utilizado no âmbito do presente projeto.

Para fazer este componente não existe nenhum elemento, nem um atributo ARIA que defina um padrão de *design* para o mesmo e é isso que o torna tão interessante, pois requer muita criatividade. Cada elemento deste cartão de produto deve ser pensado para incluir acessibilidade, tendo como objetivo saber a informação do produto e conseguir navegar entre produtos de forma rápida. Algumas informações deste componente são apenas elementos textuais e outros são elementos que por sua vez já trazem, de forma nativa, algum comportamento e regras de acessibilidade.

O componente *ProductCard* contém algumas informações e elementos que vão ser apresentados de seguida, de acordo com a sua ordem no código:

- *Link* para a página de detalhe do produto visualmente escondido, com recurso ao uso do componente *VisuallyHidden*. Este *link* permite, quer aos utilizadores que navegam por leitores de ecrã quer aos que navegam por teclado, encontrar o *link* de forma rápida que por sua vez já dá informações sobre o nome e a marca do produto. Assim, se o produto for uma **camisola branca** da marca **MarcaX**, o *link* escondido vai anunciar no leitor de ecrã e apresentar visualmente quando encontrado com a tecla *tab* (porque recebe o estado de *focus*): “**Ver camisola branca por MarcaX**”. Um exemplo deste *link* é apresentado na figura abaixo, no segundo produto apresentado.
- *SkipLink* para avançar de forma rápida para o próximo produto sem passar pelo resto das informações presentes no *ProductCard* atual. Este *SkipLink* é bastante útil na medida em que uma lista de produtos é difícil de navegar pela quantidade de produtos que apresenta, com isto é possível passar de produto em produto de uma forma rápida e eficaz. Um exemplo deste *link* é apresentado na figura abaixo, no terceiro produto apresentado;
- Imagem dinâmica do produto, que por sua vez é também um *link* para a página de detalhe do produto, oferecendo a possibilidade ao utilizador do periférico rato de conseguir aceder também a esta página da forma considerada padrão, o clique no *ProductCard*. Esta imagem possui um texto alternativo igual ao nome do produto que por sua vez é escondido para os leitores de ecrã não o anunciarem, visto que esta informação já foi anunciada anteriormente através do *link* escondido. Este *link* com a imagem contém o atributo `aria-hidden="true"` para evitar duplicar o conteúdo que já foi lido anteriormente pelos *links* escondidos;
- *Heading* de nível 2 com a informação do nome do produto, para possibilitar uma ordem lógica de *headings* na página e possibilitar que o utilizador navegue através dos *headings*, isto é, através de produtos, com recurso aos atalhos dos leitores de ecrã.

- Elemento visualmente escondido (*VisuallyHidden*) para anunciar que a marca do produto vai ser lida e para dar contexto à informação seguinte que é a marca do produto;
- Elementos visualmente escondidos (*VisuallyHidden*) para anunciar os preços associados ao produto (preço ou preço original e preço em saldo se o produto estiver com desconto). Esta informação vai ser lida antes do valor numérico dos preços referidos e serve para dar contexto à informação que vai ser anunciada de seguida pelos leitores de ecrã;
- Botão para adicionar/remover à lista de desejos, dependendo se o produto já se encontra na mesma ou não. Este botão possui uma particularidade na forma como é anunciado pois anuncia também o nome do produto que vai ser adicionado/removido à lista de desejos de forma a lembrar o utilizador de qual é o produto em questão. Assim, como forma de exemplo em vez de anunciar “Remover produto da lista de desejos”, anuncia “Remover camisola branca da lista de desejos”, dando contexto ao utilizador do que vai acontecer.

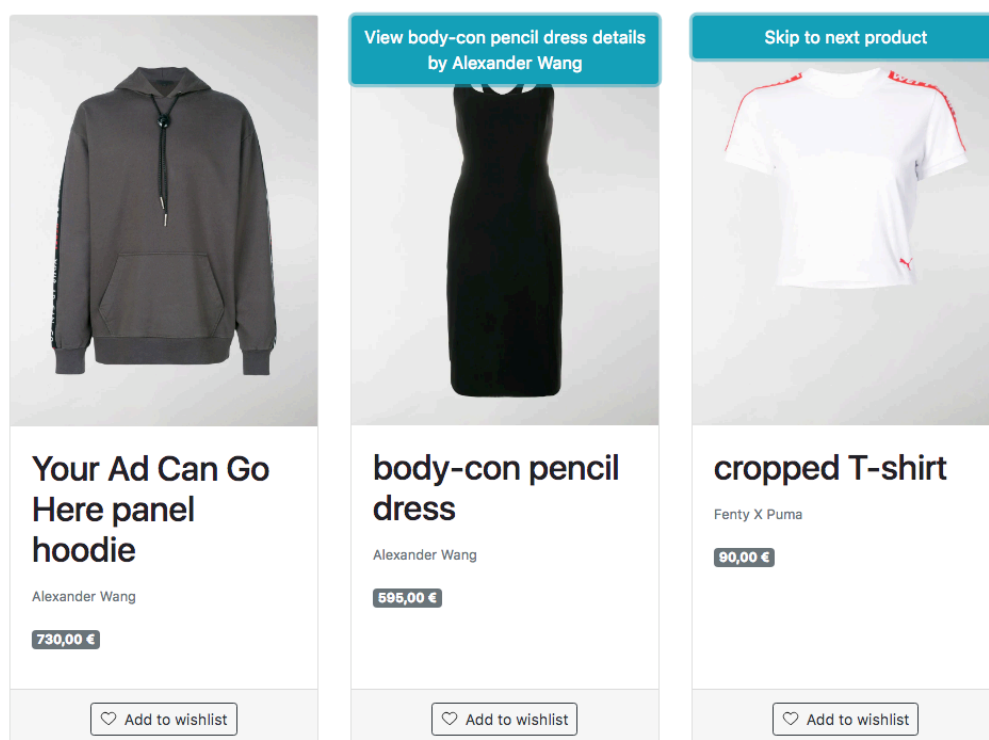


Figura 29 - Representação do *ProductCard* com o *link* para a página de detalhe do produto e com o *link* para avançar para o próximo produto

### 5.2.5 Select

Outro elemento comum nos *websites* são os elementos `<select>` que fornecem uma forma de controlar um menu com diversas opções.

Este componente já existia num repositório partilhado com objetivo de promover a sua reutilização e manutenção. Neste projeto, o mesmo foi alvo de pequenas mudanças para melhorar a sua acessibilidade e foi reutilizado para as diversas listas de opções contidas no *website*, como escolha de quantidade, escolha de tamanhos de um produto e escolha de um filtro de ordenação na lista de produtos. Este componente foi feito para aplicações *React* e é composto por algumas propriedades. De seguida são nomeadas as mais relevantes:

- *initialValue* – valor (*value*) da tag `<option>` selecionado por defeito na primeira renderização;
- *labelContent* - elemento que define o conteúdo para a *label* que descreve o *select* em questão;
- *options* – lista de objetos `<option>` com texto (*text*) e valor (*value*) para cada;
  - *value* – valor único e numérico associado à *option*, identificando-a;
  - *text* – conteúdo textual da *option* em questão;
- *placeholder* - texto que é apresentado quando ainda nenhuma opção está selecionada, ou seja, a primeira opção, com o estado *disabled* e selecionada se o parâmetro *initialValue* não for fornecido.

O uso típico deste componente deve possuir um atributo *id* para permitir ser associado à `<label>` apresentada para fins de acessibilidade, bem como um atributo *name* para representar o elemento. Cada opção do menu é definida por um elemento `<option>` dentro da tag HTML `<select>`.

Cada elemento `<option>` deve ter um atributo de valor (*value*) que define o valor a ser enviado para certo serviço quando essa opção é selecionada; se nenhum atributo de valor estiver incluído, o valor assumido por defeito é o texto (*text*) do elemento. Este componente também aceita a maioria dos atributos ARIA que são declaradas nas devidas circunstâncias, como `aria-describedby` e `aria-invalid` quando o componente possui um erro associado e `aria-required` se o seu preenchimento for obrigatório.

### 5.2.6 SvgIcon

O componente *SvgIcon* já existia num repositório partilhado com o objetivo de promover a sua reutilização e manutenção. Neste projeto, o mesmo foi alvo de pequenas mudanças para melhorar a sua acessibilidade e foi reutilizado para renderizar as diversas imagens do tipo SVG contidas no *website*.

Este componente foi feito para aplicações *React* e é composto por algumas propriedades:

- *svg* – *source*/fonte do objeto a ser lido e executado, isto é, a imagem SVG em si;
- *role* – atributo ARIA *role* que deve ser atribuído. Se nenhum valor for associado é o valor padrão do componente que é aplicado: `role="img"`;
- *title* – título acessível para associar à tag `<title>` dentro da tag `<svg>` do componente;

- *desc* – descrição acessível para associar à *tag* `<desc>` dentro da *tag* `<svg>` do componente;
- *uniqueId* – *id* único associado à *tag* `<svg>`.

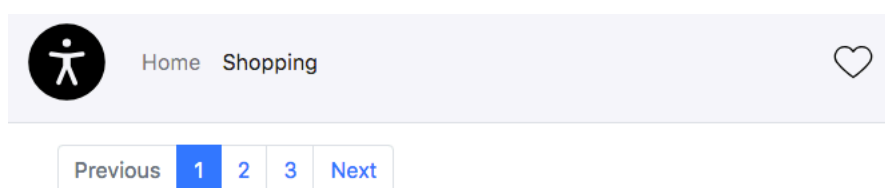
### 5.2.7 Pagination

O componente *Pagination* existe no contexto do projeto e é expectável que o mesmo seja isolado num repositório partilhado com o objetivo de promover a sua reutilização e manutenção.

Este componente segue algumas boas práticas:

- Os *links* de paginação estão dentro de uma *tag* `<nav>` com o respetivo *aria-label* e possuem o objetivo de providenciar aos utilizadores o reconhecimento da mesma como paginação;
- ARIA `role="navigation"` e `aria-label="Pagination"` para transmitir uma ideia clara aos utilizadores de que o *landmark navigation* é usado com o objetivo de paginação;
- Todos os *links* de navegação foram devidamente identificados com o atributo *aria-label*, tendo como exemplo, “Ir para a página anterior (número X), ir para a página número um, ir para a página número dois, ir para a próxima página (número Y)”. Esta técnica ajuda os utilizadores de tecnologias assistivas a receber informação clara da página para a qual estão prestes a navegar.
- Uso do atributo `aria-current` para identificar qual é o elemento da paginação que está ativo.

Na seguinte imagem é possível ver o aspeto visual do componente *Pagination* no projeto, na lista de produtos.



## New arrivals

Figura 30 - Aspeto visual do componente *Pagination*

### 5.2.8 Notificações de rotas

A necessidade deste componente surge da necessidade de a aplicação ser uma SPA, ou seja, é um *website* que renderiza o seu conteúdo em resposta a ações da navegação, sem fazer nenhum pedido ao servidor para obter um novo HTML, mantendo sempre a mesma página.

Para colmatar o facto da mudança de página não ser anunciada (uma vez que não há um “*reload*” da página, o leitor de ecrã não sabe que a página mudou porque apenas houve mudanças na história do *browser*). Como tal, este componente é apenas constituído por um elemento que facilita o uso do atributo ARIA `aria-live="assertive"` e texto visualmente escondido (com recurso ao componente *VisuallyHidden*). O seu funcionamento passa por atualizar o conteúdo desta *region* com o atributo *aria-live* para “A carregar” / “Loading” quando a página começa a carregar e posteriormente anuncia o título da página que acabou de ser carregada. Basta esta alteração no conteúdo para a *region aria-live* notificar aos leitores de ecrã do que está a acontecer e posteriormente aos utilizadores destes. Como o título das páginas aparece em primeiro lugar como um *heading* de nível 1, o leitor de ecrã anuncia o título da página, indicando então ao utilizador onde se encontra.

Nas páginas, geralmente o título da página é um *heading* de nível 1 com o conteúdo textual que for mais indicado. Em todos estes *headings* de nível 1 de todas as páginas houve um cuidado extra em adicionar o atributo `tabindex="-1"` com o intuito de este ser o primeiro elemento a receber o estado de *focus* com a finalidade de ajudar os utilizadores de tecnologias assistivas a orientarem-se nas páginas, tendo sempre este comportamento presente em todas elas, aquando da sua renderização. Todo este conjunto de funcionalidade serve, no fundo, para imitar o comportamento normal de um *website* que não é uma SPA, em que cada página faz *refresh* e é anunciado o seu título.

## 5.2.9 Header

Tal como falado anteriormente na secção 5.1.3, acerca dos *landmarks* de uma aplicação o *header* é a secção inicial de qualquer página *web*.

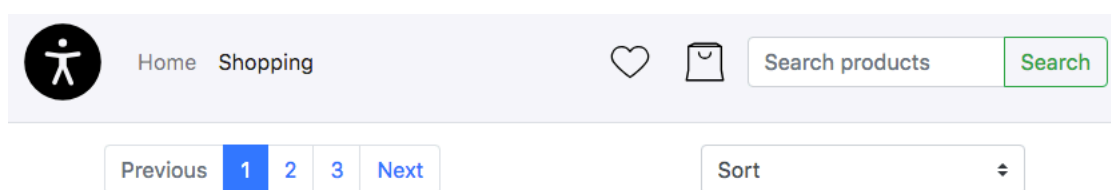
Neste projeto o *header* contém vários elementos, como é o caso do logótipo (com ligação para a *homepage*/página principal), a navegação principal para as páginas e a navegação para as páginas de *bag* e *wishlist*, juntamente com a área de pesquisa. Estas navegações encontram-se separadas por dois elementos HTML `<nav>` com os respetivos nomes acessíveis para serem identificadas facilmente pelos utilizadores, tornando-se *landmarks* distintos e tornando a navegação do utilizador final mais fácil e organizada.

Sobre a navegação principal, a navegação por páginas, existe a indicação da página atual em que o utilizador se encontra, com recurso ao atributo `aria-current`, e um conjunto de *skip links* para facilitar a navegação aos utilizadores de tecnologias assistivas (cf. secção 2.4.1).

Já na outra navegação, encontra-se um *link* para a *wishlist*, que visualmente é composto por um ícone (componente *SvgIcon*) e uma contagem de quantos itens lá estão presentes, com uso do atributo `aria-live="polite"`, para conseguir notificar o utilizador de quando este número muda, através das ações de adicionar ou remover produtos à *wishlist*, sem interromper o fluxo da aplicação e o que estava a ser anunciado na altura. Visualmente estes elementos ficam bem, mas para o leitor de ecrã foi necessário esconder o ícone (`aria-hidden="true"`) e criar mensagens próprias para que um utilizador com incapacidade visual entenda facilmente o que estes representam, nomeadamente a indicação de quantos itens estão na *wishlist* (lista de desejos) ou, caso não exista nenhum, a indicação de que a *wishlist* está vazia.

Ao lado deste *link*, encontra-se o *link* para o *bag* (carrinho de compras), que tem exatamente o mesmo comportamento e características que a *wishlist*, mudando convenientemente o texto que é anunciado.

Por fim, encontra-se um pequeno formulário de pesquisa, sem comportamento *JavaScript* associado, composto por um *input* e um botão de submissão. A este formulário foi adicionado um atributo `role="search"` para o utilizador saber exatamente do que se trata o elemento, ao *input* não foi acrescentado nenhum nome acessível em particular, uma vez que o *placeholder* existente ("*Search products*") já cumpre essa função e ao botão apenas foi acrescentado um nome acessível para melhor representar qual a sua função ("*Submit search term*").



## New arrivals

Figura 31 - Header da aplicação

Na imagem acima é possível ver o exemplo do *header* da aplicação e a forma como o mesmo é representado visualmente.

### 5.2.10 Footer

Tal como falado anteriormente na secção 5.1.3, acerca dos *landmarks* de uma aplicação o *footer* é a secção final de qualquer página *web*.

Neste projeto o *footer* contém vários elementos como os *links* para as redes sociais, *links* para assuntos legais, *link* para os *accessibility statements* (cf. secção 5.5) e os contactos.

A nível de acessibilidade, é necessário reforçar a *tag* `<footer>` com o atributo `role="contentinfo"`, acompanhado de um `aria-label` para transmitir a informação correta em diferentes *browsers*.

Relativamente às redes sociais, uma vez que têm ícones e texto associado, o seu nome acessível é diferente do que se interpreta visualmente para providenciar uma melhor identificação dos ícones e a sua finalidade. Já na parte dos contactos, uma vez que visualmente se depreende de que tipo de contacto se trata (*email*, telefone), para o leitor de ecrã esta informação não é tão clara e por isso foi acrescentado texto adicional para melhor contextualizar, dando uso ao componente *VisuallyHidden*.



### Go social

 Facebook

 Instagram

 Twitter

### Legal

[Terms and conditions](#)

[Cookies policy](#)

[Privacy policy](#)

### Contacts

Catarina Couto

[catarcouto@live.com.pt](mailto:catarcouto@live.com.pt)

916655958

Figura 32 - Footer da aplicação

## 5.3 Páginas

Nesta secção vão ser abordadas todas as páginas existentes no projeto. Estas páginas podem servir de exemplo e base para a criação de outras páginas noutros projetos, uma vez que no seu conteúdo são utilizados os componentes desenvolvidos ao longo do projeto. Todas as páginas possuem como primeiro elemento um *heading* de nível 1 com o atributo `tabindex="-1"` para este ser o primeiro elemento a ser anunciado ao entrar nas páginas. Tal como já foi referido na secção 4.2 e de acordo com a metodologia *atomic design*, utilizada no projeto, as páginas podem ser vistas como *slices*.

### 5.3.1 Homepage

A *homepage* é, de forma geral, a primeira página que um utilizador vê ao entrar num *website*. Esta página normalmente serve para dar ao utilizador uma ideia do conteúdo do *website* e a sua finalidade, facilita a navegação para outras páginas do *website*, fornece *links* para artigos/produtos do *website* e ainda disponibiliza uma área de pesquisa.

No projeto, esta página de entrada funciona como capa do projeto e é aqui que se encontra a introdução e contextualização, bem como um botão para começar a navegar e explorar a lista de produtos. No conteúdo principal desta página surgem diversos apontamentos de acessibilidade, nomeadamente o texto alternativo da imagem de fundo, uma secção alvo dos *skip links (banner)* do *footer* e um botão com um nome acessível diferente do que está visível para conseguir providenciar ao utilizador mais contexto. Na imagem seguinte é possível ver o aspeto visual da *homepage* no projeto.

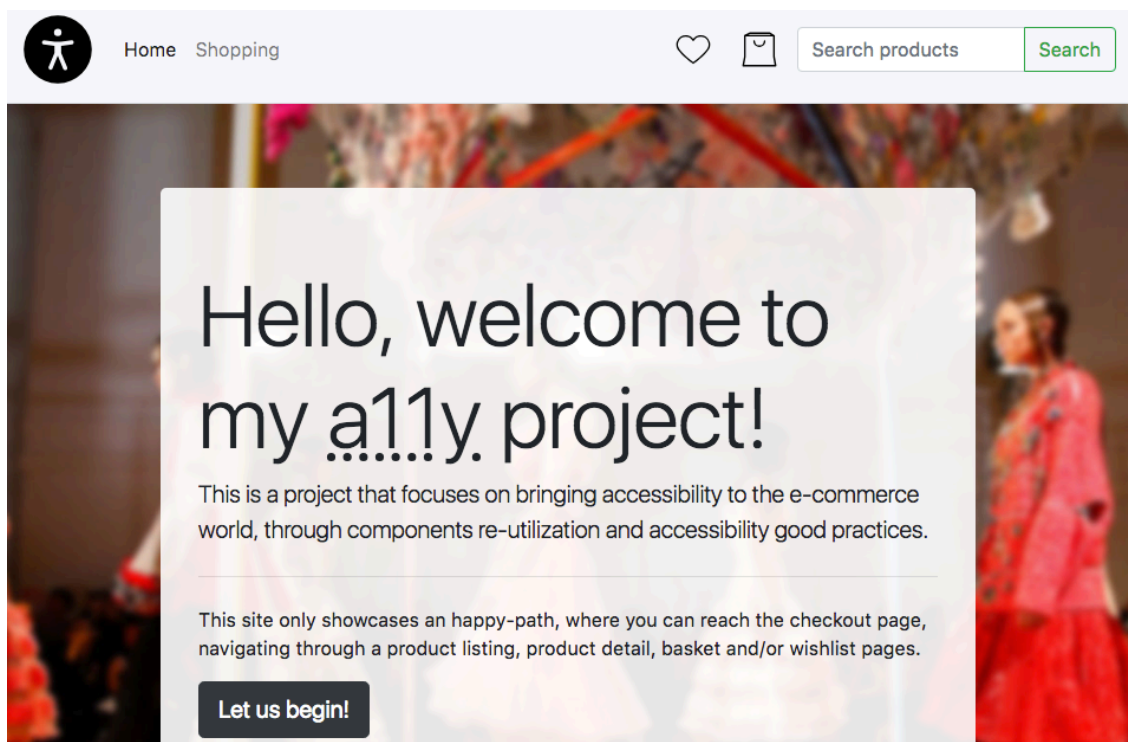


Figura 33 - *Homepage* da aplicação

### 5.3.2 Lista de produtos

Nesta página encontra-se a listagem de produtos, logo a seguir ao componente de paginação (cf. secção 5.2.7) e ao uso do componente *Select* (cf. secção 5.2.5) para a ordenação (*sort*) da lista de produtos. Esta lista é composta por vários componentes do tipo *ProductCard* (cf. secção 5.2.4), com um *link* para a página de detalhe do respetivo produto.

Relativamente à ordenação, tem também um nome acessível diferente do que está visível, visivelmente aparece “*Sort*”, mas para os leitores de ecrã é anunciado “*Sort the product’s list*”.

Outra particularidade desta página é que o conteúdo é carregado de forma assíncrona, o que torna especialmente útil o uso do componente Notificações de Rotas (cf. secção 5.2.8), na medida em que notifica o leitor de ecrã que a página está carregada e pronta apenas depois do conteúdo estar totalmente carregado. Na imagem seguinte é possível ver o aspeto visual da lista de produtos no projeto.

## New arrivals

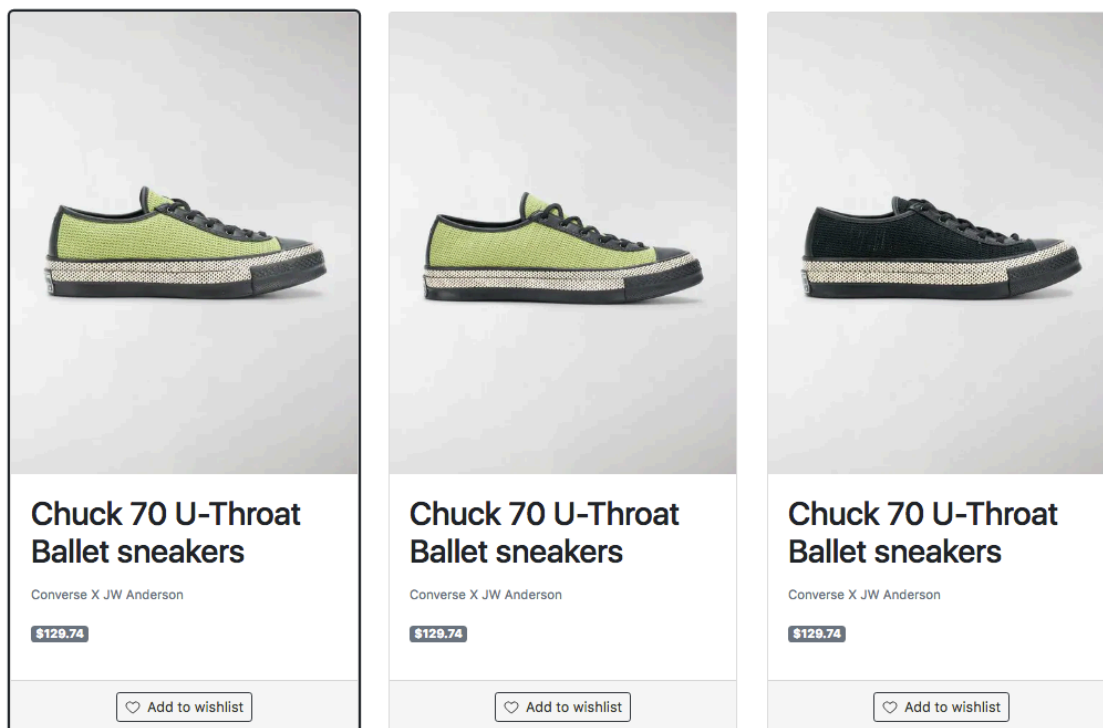


Figura 34 – Lista de produtos da aplicação

### 5.3.3 Detalhe de um produto

Nesta página é apresentada toda a informação de um determinado produto, desde nome, marca, descrição e imagens, bem como permite adicionar ao carrinho de compras (*bag*) (depois de escolhido o tamanho) e à lista de desejos (*wishlist*). Esta página tem também a particularidade de apresentar um conteúdo diferente caso o produto não tenha tamanhos em *stock*.

Já a nível de acessibilidade, encontram-se alguns detalhes:

- O botão de adicionar ao *bag* tem um nome acessível diferente do visível, porque é boa prática voltar a referir que produto está a ser adicionado, através do uso do atributo *aria-label*. É de salientar que quando um produto é adicionado ao *bag* é acionada a notificação do *header*, proveniente da *region aria-live*;
- Existe uma mensagem de erro com o devido tratamento (cf. secção 5.1.15) para quando se tenta adicionar ao *bag* sem se ter escolhido o tamanho, sendo esta mensagem um elemento com uso do atributo *aria-live*, para assinalar imediatamente que ocorreu um erro;
- O botão de adicionar à *wishlist* tem as mesmas características que o do *bag*, na medida em que também tem o nome do produto para providenciar um nome

acessível. De salientar que quando um produto é adicionado à *wishlist* é acionada a notificação do *header*, proveniente da *region aria-live*;

- Toda a descrição detalhada do produto está dividida pelos vários painéis de uma área colapsável com recurso aos atributos *aria-controls* e *aria-expanded*;
- As imagens apresentadas possuem um texto alternativo.

Na imagem seguinte é possível ver o aspeto visual do detalhe de um produto no projeto.

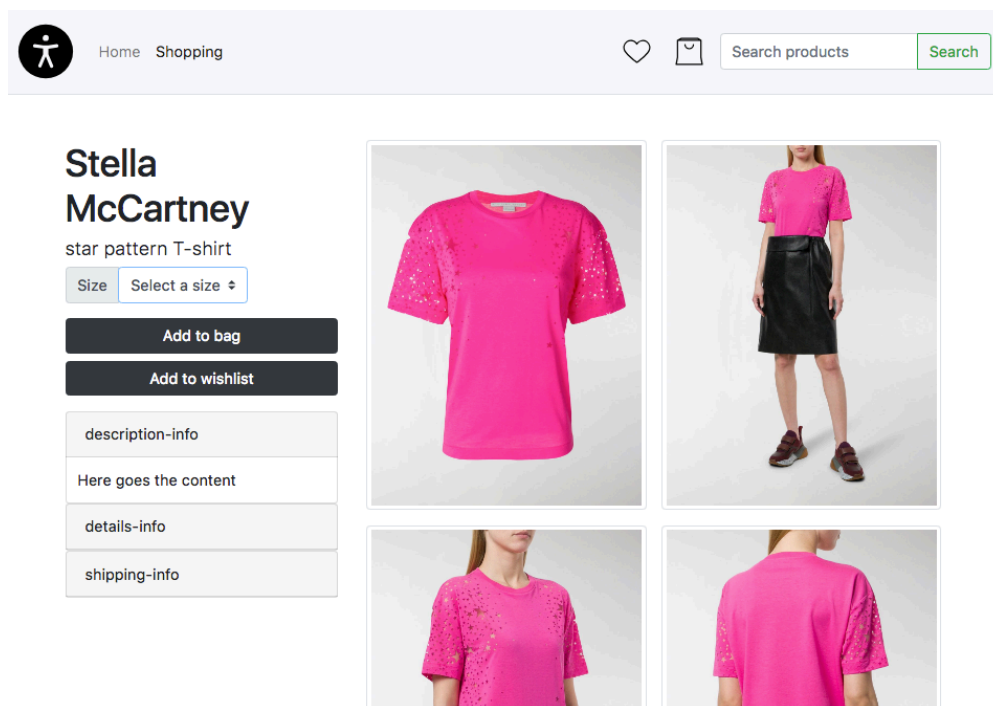


Figura 35 – Detalhe de um produto no projeto

### 5.3.4 Lista de desejos

A lista de desejos também conhecida como *wishlist* é a página onde são apresentados os produtos que foram adicionados à *wishlist* pelo utilizador. Esta lista é composta apenas por uma lista de *ProductItems* (componente semelhante ao *ProductCard*, com um *layout* horizontal), ou, no caso de não existirem produtos, é apresentada uma informação textual a avisar o utilizador da inexistência de produtos e um *link* para ir para a lista de produtos.

Na imagem seguinte é possível ver o aspeto visual da *wishlist* no projeto.

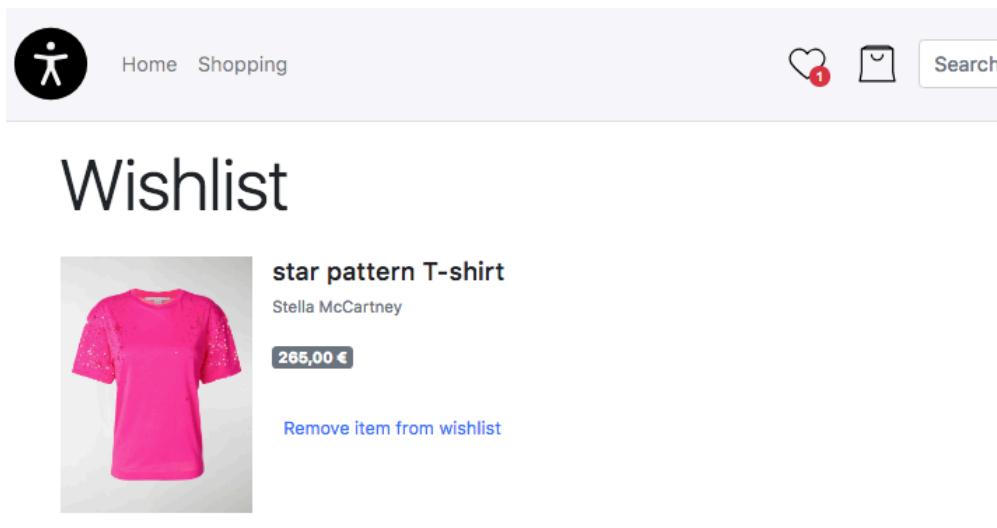


Figura 36 – *Wishlist* do projeto

### 5.3.5 Carrinho de compras

O carrinho de compras também conhecido como *bag* é a página onde são apresentados os produtos que foram adicionados ao *bag* pelo utilizador. Esta lista é composta apenas por uma lista de *ProductItems* (componente semelhante ao *ProductCard*, com um *layout* horizontal), ou, no caso de não existirem produtos, essa mesma informação seguida de um *link* para voltar à página lista de produtos. Esta listagem é também acompanhada de um sumário, que consiste numa tabela com as parcelas a pagar pelos produtos escolhidos - é importante ser uma tabela e não apenas elementos `<div>`, porque existem atalhos e formas específicas para os leitores de ecrã navegarem por uma tabela. Abaixo desta tabela encontram-se dois *links*, um para voltar à página da lista de produtos e outro para iniciar o processo de compra, o *checkout*.

Na imagem seguinte é possível ver o aspeto visual do *bag* no projeto.

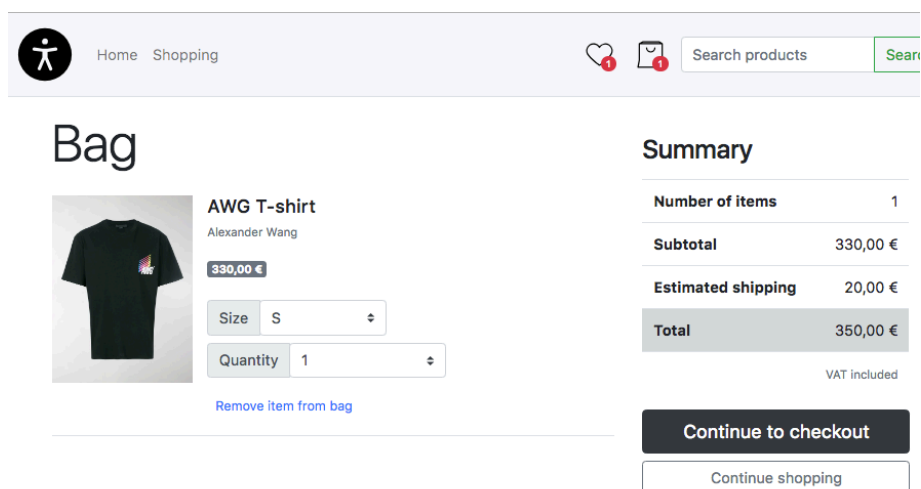


Figura 37 – *Bag* do projeto

### 5.3.6 Checkout

O *checkout* é a página final da aplicação, onde hipoteticamente se poderia fazer compras. É a página que comprova que a prova de conceito desejada para o projeto foi efetivamente conseguida, e o seu conteúdo textual revela exatamente isso.

A acompanhar esta descrição, tem, tal como no *bag*, o mesmo sumário com informação das parcelas a pagar, juntamente com uma pequena listagem dos produtos a comprar. Na imagem seguinte é possível ver o aspeto visual do *checkout* no projeto.

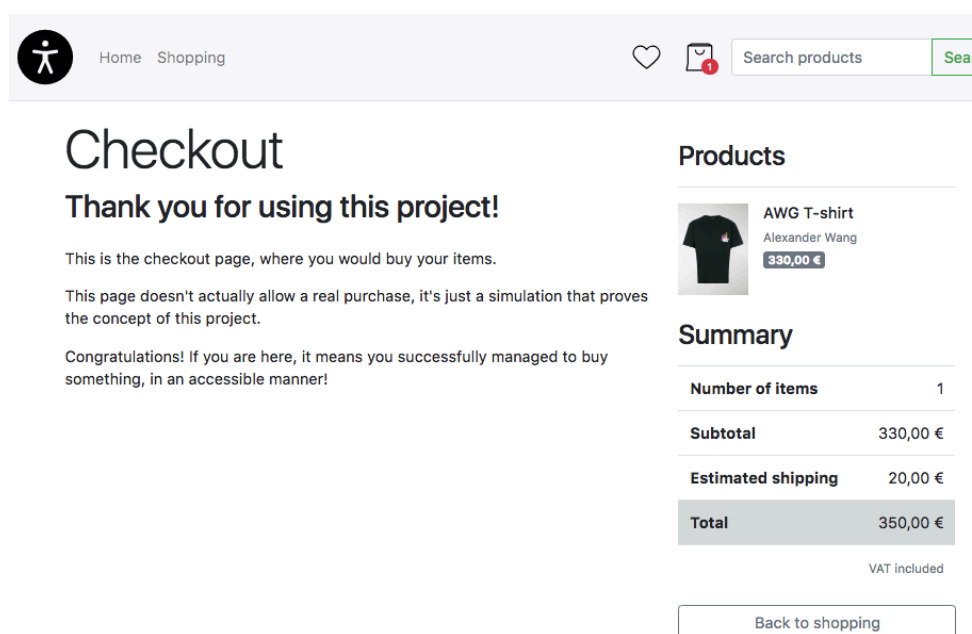


Figura 38 - Página de *checkout* do projeto

### 5.3.7 Página de erro

A página de erro é provavelmente a página mais simples de todas, na medida em que simplesmente apresenta uma mensagem a indicar que a página não foi encontrada, acompanhada de um *link* para voltar à lista de produtos.

Na imagem seguinte é possível ver o aspeto visual da página de erro no projeto.

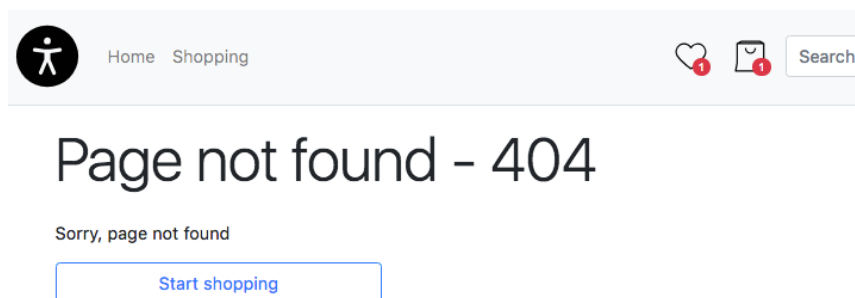


Figura 39 - Página de erro do projeto

## 5.4 Atalhos e opções das tecnologias assistivas

A tecnologia assistiva utilizada para testar este projeto foi o leitor de ecrã. O leitor de ecrã escolhido foi o *VoiceOver* (cf. secção 2.4.1), leitor de ecrã integrado no sistema operativo *MacOS*, aquando do uso dos *browsers Safari* e *Google Chrome*.

Para uma melhor utilização desta tecnologia, alguns atalhos foram bastante utilizados, nomeadamente:

- Command + F5 – Ligar e desligar o VoiceOver;
- Ctrl + Option + U – Acesso ao “*Web Rotor*”, menu que lista vários painéis com atalhos para *Headings*, *Landmarks*, *Form elements*, *Links*, entre outros;
- Setas do teclado - Mover para o item adjacente na lista/ecrã, menu ou conteúdo HTML (mesmo que não seja *focusable*, para permitir a sua leitura);
- Tab e Shift + Tab - Mover para o próximo/anterior conteúdo HTML com interação, respetivamente;
- Tecla espaço - Interação com botões, áreas colapsáveis ou semelhante;
- Tecla Enter - Interação com *links*.

## 5.5 Accessibility statements

Segundo a Léonie Watson, os *accessibility statements*, também conhecidos como declarações de acessibilidade estão cada vez mais a ser implementados nos *websites* (Léonie Watson, 2009) e definem o nível de acessibilidade que o *website* pretende alcançar. Estas declarações são normalmente incluídas no *footer* de um *website*, como um *link* que redireciona para a página onde estão incluídas todas as declarações de acessibilidade do *website*, como forma de garantir que esta informação está disponível e presente em todas as páginas.

A página que apresenta os *accessibility statements* tem como funções:

- Apresentar informações claras sobre o nível-alvo de acessibilidade que o *website* se compromete a cumprir e quais os métodos usados para atingir esse nível;
- Permite que os proprietários dos *websites* reconheçam as áreas do *website* em que os objetivos de acessibilidade não tenham sido totalmente atingidos e estabeleça o plano proposto para resolver esse problema;
- Sugere uma declaração poderosa de compromisso, conseguindo com que as pessoas que visitam o *website*, especialmente as portadoras de incapacidades, apreciem o reconhecimento público de que a acessibilidade é um fator-chave e importante para o *website*.

Aquando do *feedback* da Léonie Watson (cf. secção 6.3.1) foi possível perceber a importância que a mesma dá aos *accessibility statements* quando estes estão presentes, pois ajudam a perceber que foi feita uma pesquisa com o intuito de providenciar um *website* acessível aos utilizadores. A mesma defendeu ainda que estas declarações são procuradas por pessoas com incapacidades para saber ao que é que o *website* se comprometeu a cumprir e para saberem como agir quando querem reportar algum erro/sugerir uma melhoria. A par disto, a própria Léonie Watson deu a conhecer o *Accessibilty Statement Generator*, que permite gerar estas declarações de forma a serem integradas no *website*, e que foi utilizado no presente projeto para gerar os *accessibility statements*.

As declarações de acessibilidade e as páginas de ajuda que são normalmente incluídas na mesma área do *footer*, são frequentemente confundidas, pelo que para simplificar e facilitar o seu uso, estas páginas devem ser separadas. As pessoas que procuram ajuda nem sempre consideram que uma declaração de acessibilidade é o local lógico a visitar. Assim, as declarações de acessibilidade devem incluir os seguintes componentes:

- Declaração de um compromisso aberto para fornecer um *website* acessível;
- Um resumo das principais diretrizes e padrões de acessibilidade que o *website* segue;
- Quaisquer exceções não implementadas conhecidas para o nível pretendido de acessibilidade;
- Informações de contacto para os utilizadores conseguirem relatar dificuldades sentidas no uso do *website* ou sugerir melhorias.

### 5.5.1 Compromisso aberto

Normalmente é o parágrafo de abertura das declarações de acessibilidade e deve conter declarações claras, objetivas e sucintas das intenções do *website* e dos seus proprietários/marca associada para com o tema acessibilidade.

No projeto, na página das declarações de acessibilidade é possível ler o seguinte (traduzido para português):

“O projeto de acessibilidade e os seus proprietários estão empenhados em fornecer um *website* acessível ao maior público possível, independentemente da tecnologia ou

incapacidade apresentada. Estamos a trabalhar ativamente para aumentar a acessibilidade e a usabilidade do nosso *website* através da adesão a muitos dos padrões e diretrizes de acessibilidade disponíveis.”

### 5.5.2 Diretrizes e padrões de acessibilidade

O próximo passo lógico é incluir informações sobre as diretrizes e os padrões que o *website* usa para maximizar a acessibilidade na *web*. As diretrizes WCAG (cf. secção 2.3.2) refletem os padrões comuns mais usados pelos desenvolvedores de *websites* acessíveis. Este parágrafo deve explicitar claramente o nível de acessibilidade na *web* que o *website* pretende alcançar e deve explicar o que são as WCAG e que estão em conformidade com as suas orientações, conseguindo assim ajudar pessoas com incapacidades.

No projeto, na página das declarações de acessibilidade é possível ler o seguinte (traduzido para português):

“Este *website* está em conformidade com o nível AA das Diretrizes de Acessibilidade ao Conteúdo da Web 2.1 (WCAG 2.1) do World Wide Web Consortium (W3C). Estas diretrizes explicam como tornar o conteúdo da *web* mais acessível para pessoas com incapacidades. A conformidade com essas diretrizes ajuda a tornar a *web* mais acessível para todas as pessoas.”

Deve ainda existir um parágrafo que informe os utilizadores sobre os padrões mais técnicos que o *website* segue, isto é, em que *browsers* funciona e explicar que vai continuar a ter o mesmo compromisso quando algum *browser* novo for lançado.

No projeto, na página das declarações de acessibilidade é possível ler o seguinte (traduzido para português):

“Este *website* foi construído usando código compatível com os padrões W3C para HTML e CSS. O site é exibido corretamente nos *browsers* atuais e o uso de código HTML / CSS é compatível com os padrões. Assumimos um compromisso de que nos *browsers* que possam ser lançados futuramente o mesmo conteúdo também vai ser exibido corretamente.”

### 5.5.3 Exceções

Às vezes, apesar do esforço, há áreas do *website* que estão fora das metas de acessibilidade da *web* pretendidas. Este parágrafo serve para reconhecer esses pontos de falha e pode ajudar as pessoas a entender as limitações do *website*. Estas exceções não devem ser usadas como uma forma fácil de fugir à implementação de acessibilidade e todas as áreas onde a acessibilidade não é fornecida devem ser indicadas, juntamente com o trabalho futuro que é idealizado para resolver a falta de acessibilidade nesses pontos.

No projeto, na página das declarações de acessibilidade é possível ler o seguinte (traduzido para português):

“O projeto de acessibilidade esforça-se para aderir às diretrizes e padrões existentes de acessibilidade e usabilidade, no entanto nem sempre é possível fazê-lo em todas as áreas do

*website*, como é o caso de algumas imagens de produtos que possuem um texto alternativo que não está de acordo com os parâmetros de acessibilidade definidos.”

#### 5.5.4 Informações de contacto

Por último, é importante fornecer às pessoas a capacidade de contactar o *website* sobre problemas específicos que possam ocorrer. Um endereço de *email* simples ou um formulário de contacto são exemplos de uma boa estratégia para este caso.

No projeto, na página das declarações de acessibilidade é possível ler o seguinte (traduzido para português):

“Estamos de forma contínua a procura soluções e melhorias que levem todas as áreas do presente *website* ao mesmo nível de acessibilidade geral da web. Nos entretantos, se detetar alguma dificuldade em aceder ao projeto de acessibilidade, não hesite em contactar.”

## 5.6 Reutilização e manutenção

Nesta secção é possível verificar o contributo deste projeto para a comunidade, quer a nível de componentes já disponibilizados, como possíveis componentes a isolar. É ainda possível verificar que também neste projeto foram utilizados outros componentes já existentes, com o objetivo de seguir as boas práticas da equipa.

Para a criação/isolamento, manutenção e a própria reutilização de componentes já existentes de *React*, todo o trabalho foi realizado com recurso ao repositório partilhado entre equipas denominado *react-box* (*monorepo* cf. secção 4.2.1.1). Já para componentes *redux*, que envolvem lógica de negócio e tratamento de dados, na perspetiva apenas de reutilização, todos os componentes usados são consumidos diretamente da *redux-box* (*monorepo* cf. secção 4.2.1.1) para obter os dados dinâmicos da aplicação, como é o caso da lista de produtos, carrinho de compras, lista de desejos, respostas a pedidos à API de serviços, entre outros.

### 5.6.1 Criação de componentes e sua utilização

Com este projeto foi possível chegar à criação de vários componentes (cf. secção 5.2), sendo que dois já foram isolados na *react-box*, porque ainda não existiam e faziam falta para melhorar a experiência a nível de acessibilidade - *Visually Hidden* e *Skip Link*. Estes componentes não só se mostraram úteis no contexto deste projeto, como também, com o seu isolamento para a *react-box*, provaram ser úteis noutros projetos de marcas, onde já foram reutilizados, conseguindo assim atingir o objetivo de reutilização e boas práticas de partes do presente projeto e conseguindo contribuir para uma fácil reutilização e manutenção dos mesmos.

## 5.6.2 Uso de componentes

Além da criação de novos componentes necessários ao projeto, outros já estavam criados e foram também reaproveitados no projeto, uma vez que já cumpriam os requisitos de acessibilidade e eram úteis.

Foram então reaproveitados componentes como o *Button*, *SubfolderLink*, *Responsive Image*, *Select*, *SvgIcon*, entre outros. O processo de integração destes componentes no projeto passou por verificar se os mesmos cumpriam o propósito da acessibilidade e integrar o mesmo no código demonstrando, mais uma vez, as vantagens de utilizar um componente isolado: evitar código repetido em projetos diferentes.

## 5.6.3 Manutenção de componentes

Com o uso de componentes já existentes, é natural que em alguns deles falte alguns pormenores ou funcionalidades, ou até mesmo que contenham pequenos erros, tornando alguma funcionalidade incompleta ou menos bem conseguida. Neste caso, não só foi necessária a criação de alguns componentes, como também foi requerida a manutenção de outros, de forma a poder responder às necessidades em questão deste projeto. Exemplos disto são o *react-head*, onde foi feita uma alteração para permitir manipular o `<title>` de uma página, o *select*, onde foi corrigido o valor padrão selecionado inicialmente no caso de existir *placeholder*, o *SvgIcon* que foi alvo de alterações para que o parâmetro recebido para o *id* do atributo ARIA *aria-labelledby* fosse obrigatório. Neste processo, foram identificadas algumas mudanças que os componentes já existentes poderiam sofrer e, quando possível, essa mudança foi realizada no repositório partilhado. Por outro lado, quando não era possível realizar as melhorias no repositório partilhado, com o objetivo de não criar demasiadas dependências que poderiam fazer com que o presente projeto atrasasse o código desses componentes foi copiado para o projeto e melhorado, tendo como objetivo a inclusão dessas melhorias nos repositórios partilhados como trabalho futuro.

## 5.7 Processo de contribuição

Esta secção documenta o processo de contribuição que define como é que se deve proceder quando é pretendido sugerir alguma funcionalidade ou correção a um repositório. Existem alguns processos a seguir, de forma à informação estar toda disponível e visível para qualquer desenvolvedor. Desta forma é possível evitar duplicados, ou simplesmente partilhar o conhecimento sobre algum assunto, conseguindo usar a mesma forma de comunicação entre todos, já que existem processos e diretrizes de contribuição a seguir.

### 5.7.1 Templates

Tanto para o relato de um *issue* (problema encontrado) ou de um *merge request* (pedido de contribuição) é necessário fornecer algumas informações base, que são chave para um bom entendimento por parte de qualquer pessoa e que seguem algumas regras. Como tal existem três tipos de *templates* (dois para *issues* (*bug* ou *feature*) e um para *merge request*), de forma a contemplar que informação é necessária para cada caso:

- Issues
  - *Bug* - onde é reportado algo que não está a funcionar de acordo com o expectável (são pedidas informações como o tipo de *bug*, uma descrição, como se reproduz, a plataforma onde aconteceu, o que foi feito para resolver, entre outros);
  - *Feature* - onde é reportado algo que se gostaria de ver adicionado a determinado componente, tipicamente são funcionalidades novas, ou onde se dá sugestões de melhoria de determinada funcionalidade.
- Merge request
  - Existe o *template "default"* que possui toda a informação necessária para este tipo de contribuição (se a mensagem de *commit* segue as diretrizes, se foram adicionados testes, qual o propósito do pedido de contribuição, quais foram as alterações realizadas, entre outros) com o intuito de ser consistente entre todos os *merge requests* e pedidos de contribuição.

### 5.7.2 Issues

*Issues* é onde os contribuidores podem manifestar a sua intenção - seja para relatar um erro, sugerir uma melhoria, propor uma funcionalidade ou simplesmente abrir uma discussão sobre alguma dúvida do projeto.

Tipicamente é o primeiro local onde um contribuidor acede, e onde consulta se a sua ideia de contribuição já foi sugerida ou não. Depois da confirmação de que a mesma ainda não existe, a contribuição é aberta, para que fique disponível para qualquer pessoa poder visualizar, comentar e dar opinião. O próximo passo pode acontecer de duas formas: ou o responsável do repositório endereça o *issue* e o resolve, aceitando a contribuição, ou espera que alguém da comunidade tome a iniciativa de rever a contribuição e mais tarde aceitá-la. Em qualquer uma das formas, a lista de *issues* funciona como uma base de dados/lista de assuntos que são necessários endereçar, resolver e contribuir.

### 5.7.3 Merge requests

Os *merge requests* (também frequentemente conhecidos como *pull requests*) acontecem na fase posterior aos *issues* (apesar de que os *issues* não são obrigatórios, isto é, um não implica o outro). Um *merge request* é a implementação de algo (seja consequência de um *issue*, uma funcionalidade nova ou apenas uma pequena correção) que fica disponível para validação por parte da comunidade - o código passa por uma nova fase, a fase de *code review*.

Esta fase é importante para assegurar a qualidade do código e garantir que as diretrizes são cumpridas, pois com outras pessoas a rever é possível evitar que alguns erros sejam introduzidos. Durante o *code review*, a pessoa que está a rever pode comentar cada linha de código, ou simplesmente deixar comentários ao *merge request* em si, e é esta a forma de comunicação típica entre a comunidade e quem implementou a funcionalidade.

Terminado o *code review* (processo que é determinado pelo responsável do repositório, mas que tipicamente acontece depois da aprovação de duas pessoas - aprovação essa que é conseguida através da atribuição de *thumbs up*<sup>15</sup>), o *merge request* pode então ser fechado (*merged*) ou seguir para aprovação de *Quality Assurance*<sup>16</sup>, dependendo do seu conteúdo.

#### 5.7.4 Commits

Para conseguir contribuir com um *merge request*, é preciso guardar as alterações a propor. É aqui que surgem a responsabilidade dos *commits* - um *commit* em GIT é o equivalente ao “Guardar” de qualquer outra aplicação. No entanto, este guardar tradicional deve ser considerado como uma operação do sistema de ficheiros para sobrescrever um ficheiro já existente ou gravar um novo. Alternativamente, o *commit* do GIT é uma operação que atua sobre um conjunto de ficheiros e pastas (Atlassian).

Estes *commits* ficam guardados no histórico do repositório, e como tal precisam de ser identificados com uma mensagem (além da *commit hash* que é gerada pelo GIT). Como a comunidade toda pode contribuir, e sem regras de contribuição para *commits*, é natural que as mensagens sejam totalmente díspares de pessoa para pessoa. Para evitar este problema, foi adotada uma regra/convenção para todos escreverem da mesma forma - que neste caso se designam por *conventional commits*.

Esta convenção contém os seguintes elementos estruturais, para comunicar a intenção à comunidade:

- *fix*: um *commit* deste tipo resolve algum problema (*bug*) no código;
- *feat*: um *commit* deste tipo introduz uma nova funcionalidade ao código;
- *chore*: um *commit* deste tipo indica que algo no código foi alterado, mas sem corrigir nada ou introduzir uma nova funcionalidade (exemplos disto são *refactors* ou atualização de componentes);
- *breaking change*: um *commit* que tenha esta informação no conteúdo da sua mensagem vai introduzir uma diferença no produto final- o que significa que qualquer utilizador poderá ter que efetuar mudanças para continuar a ter o comportamento habitual do seu componente, conseguindo que este funcione corretamente e que esteja atualizado. Esta informação é válida para qualquer tipo de *commit* (*fix*, *feat* e *chore*).

Outra vantagem desta convenção é que está diretamente relacionada com a estratégia de versionamento de aplicações utilizada, o *semver* - sob este esquema, os números das versões e a forma como eles mudam transmitem informação sobre que alterações foram realizadas ao

---

<sup>15</sup> Imagem gráfica com um contador que representa o número de pessoas que aprovaram determinado acontecimento, neste caso, o número de pessoas que já aprovaram o *merge request*.

<sup>16</sup> *Testers*, isto é, pessoas pertencentes à equipa que garantem a qualidade de um produto/serviço e que realizam diversos testes para tal feito.

código subjacente e mostra as modificações entre uma nova versão e a anterior. O versionamento das aplicações seguem o esquema *MAJOR.MINOR.PATCH*, sendo que:

- *MAJOR* - versão que introduz mudanças na API ou resultado final (relacionado com o “*breaking change*” dos *conventional commits*);
- *MINOR* - versão que introduz funcionalidades de forma compatível (relacionado com o “*feat*” dos *conventional commits*);
- *PATCH* - versão que introduz correções de forma compatível (relacionado com o “*fix*” dos *conventional commits*).

## 5.8 Testes unitários

Testes unitários são métodos de teste nos quais unidades individuais de código são testadas para determinar se estão aptos para uso, sem problemas (*bugs*) ou efeitos secundários que não se relacionem da forma correta e esperada com o resto da aplicação. São também muito úteis (e frequentemente usados) para garantir o bom funcionamento depois de um *refactor*. Não só assegura que o correto comportamento se mantém, como dá confiança ao desenvolvedor e restante equipa de conseguir melhorar o código, sabendo que não está a danificar a funcionalidade. Isto demonstra a importância que este tipo de testes representa, considerando também que é a primeira e mais básica camada de testes à qual o desenvolvedor se sujeita.

### 5.8.1 Tecnologia e método

Neste projeto, tal como em alguns dos projetos que utilizam *React*, as tecnologias utilizadas para realizar testes unitários são *Enzyme*<sup>17</sup> e *Jest*<sup>18</sup>. Já a nível de metodologia, é seguido o padrão **Arrange Act Assert**<sup>19</sup>, ainda que não seja forçado ou esteja efetivamente representado em todo e qualquer teste. Neste padrão, cada teste é agrupado em 3 secções:

- *Arrange* - organização de todas as pré-condições e dados para os testes;
- *Act* - Atuar no objeto ou método em teste com os dados;
- *Assert* - Declarar que os resultados esperados ocorrem e documentar nos relatórios.

---

<sup>17</sup> Conjunto de utilitários em *Javascript* para *React*, que torna mais fácil manipular e percorrer os componentes de *React*.

<sup>18</sup> Plataforma de testes universal com a capacidade de adaptação a qualquer biblioteca *Javascript* ou *framework*.

<sup>19</sup> Padrão para organizar e formatar o código nos métodos dos testes unitários.

### 5.8.2 Coverage associada

A *coverage* fica a cargo do *Istanbul*<sup>20</sup> e é uma indicação da percentagem de código que foi abrangido pelos testes unitários, dividindo-se em 4 áreas:

- *Statements* – quantidade de declarações do código que são executadas;
- *Branches* – quantidade de declarações condicionais que criam ramos de código que podem não ser executados (por exemplo, *if/else*). Esta métrica informa quantas das ramificações foram testadas;
- *Functions* – quantidade das funções/métodos definidos que foram chamados;
- *Lines* – quantidade de linhas de código que foram executadas.

No caso do presente projeto, a *coverage* só é considerada válida se estiver acima de um total de 80%. Caso contrário, o teste é considerado não válido. Isto é importante para garantir que grande parte do código é testada, diminuindo os erros que possam ser introduzidos e garantindo a qualidade do produto.

---

<sup>20</sup> Executa o código *Javascript* que lhe é associado com contadores, para ser possível verificar e acompanhar o desempenho dos testes unitários.

## 6 Avaliação

Este capítulo descreve o processo de avaliação do software e servirá para apurar a eficácia para a equipa de desenvolvimento e satisfação do desenvolvedor (de novos *websites*) face ao protótipo desenvolvido, no seu estado final.

A avaliação terá em conta os interesses das diferentes partes interessadas (*stakeholders*):

- A equipa de desenvolvimento de software, “clientes” internos – colaboradores da *Farfetch*, e que pretendem organizar e reutilizar o código e algoritmos, de forma a que se consiga ter especial atenção à acessibilidade;
- Os clientes finais que são os utilizadores dos *websites e-commerce*, e que pretendem aceder a *websites* que cumprem certos requisitos de acessibilidade, conseguindo assim usufruir de uma utilização acessível e fácil;
- Cliente/loja, que pretende ter acessibilidade implementada nos seus *websites e-commerce* cumprindo as leis e regulamentos em vigor no país de operação.

### 6.1 Abordagem

Descreve-se nesta secção a abordagem específica a seguir na avaliação dos interesses de cada uma das partes interessadas.

#### 6.1.1 Desenvolvedores

O interesse deste público-alvo é perceber se a organização e reutilização de código e algoritmos, com especial atenção a acessibilidade, foi conseguida e efetuada com sucesso. Com objetivo de aferir do cumprimento dos objetivos associados à equipa de desenvolvimento foram realizadas as seguintes avaliações:

- Para avaliar se a reutilização de código através de componentes funcionou, para a aplicação final foi verificado se os projetos/componentes estão a ser usados noutros

projetos, que não o próprio e se os componentes do projeto final estão a ser reutilizados, com sucesso, no próprio projeto.

- Para verificar os restantes requisitos não funcionais, foi realizado um inquérito de satisfação aos desenvolvedores da equipa que acompanharam o projeto, com algumas métricas capazes de verificar em que medida é que o projeto foi satisfatório, ou não e se cumpriu os requisitos (organização, reutilização de código e disponibilização de componentes acessíveis) e o esperado.

### 6.1.2 Clientes finais

Com objetivo de aferir do cumprimento dos objetivos associados aos clientes finais foram realizadas as seguintes avaliações:

- Para avaliar a satisfação dos utilizadores finais, foram realizados testes de usabilidade da aplicação (testar com *screen-reader* e só com o uso do teclado), a fim de perceber se a aplicação cumpre os requisitos e verificar se o utilizador final ficou ou não satisfeito. Estes testes de usabilidade foram realizados pelos utilizadores finais, seguindo um guião previamente fornecido com etapas a cumprir em conjunto com o inquérito;
- Na mesma ótica que os testes de usabilidade, foram realizadas experiências de utilização do *website* e protótipo com utilizadores finais com incapacidades, neste caso a Léonie Watson. Esta experiência permitiu identificar as práticas mais utilizadas por pessoas com incapacidades visuais e descobrir limitações e melhorias que o projeto podia incorporar, visto que estes testes não foram realizados no estado final da aplicação.

### 6.1.3 Cliente/loja

O público-alvo cliente/loja, são os representantes do segmento de clientes deste projeto (cf. secção 3.7). Visto que o interesse deste público-alvo é perceber em que medida o *website*/protótipo final é acessível e em que medida foi vantajoso o uso de acessibilidade no seu projeto, não serão realizados métodos de avaliação nem serão realizadas com pessoas cliente/loja incluídas no processo.

Contudo, foram realizadas as seguintes avaliações:

- O *website* do cliente/loja foi testado a nível de acessibilidade, seguindo a ótica e interesses do cliente/loja, i.e., assegura o cumprimento de lei e o uso de tecnologias assistivas.
- Ao avaliar os interesses dos clientes finais, está-se implicitamente a avaliar o interesse último do cliente/loja (i.e., a satisfação do cliente final) - verificar se a acessibilidade permite a utilização do *website*.

## 6.2 Preparação de experiências

Nesta secção é descrita a preparação das experiências a realizar para auferir a avaliação dos interesses de cada uma das partes interessadas.

### 6.2.1 Experiências realizadas

As experiências para auferir a avaliação foram as seguintes:

- Elaboração de guiões para testes de usabilidade ao protótipo desenvolvido e respetivos componentes, sendo a parte inicial do inquérito;
- Inquéritos de satisfação para os desenvolvedores, a fim de perceber se estão satisfeitos com os resultados obtidos e com o projeto desenvolvido em prol da equipa *Black & White* e em prol dos utilizadores finais, com a finalidade de apurar se os *websites* estão acessíveis ou não e o que é possível melhorar;
- Uso de verificadores de acessibilidade (*checkers*), como consequência de leitura do guião, com o intuito de verificar se os *websites* finais, que usam o protótipo e os componentes, ficam acessíveis ou não e se é possível melhorar no campo da acessibilidade;
- Uso de componentes já isolados em projetos para os clientes finais.

Sistematizando, foram realizadas as seguintes avaliações por *stakeholder* (Tabela 11):

Tabela 11 – Avaliações realizadas por *stakeholder*

	Guiões	Inquéritos	Checkers
Desenvolvedores	Sim	Sim	Sim
Clientes finais	Sim	Sim	Sim
Cliente/Loja	Não	Não	Não

### 6.2.2 Guião

Para melhor interpretação, por parte dos desenvolvedores e dos clientes finais, foi disponibilizado um guião, no início do inquérito, com instruções sobre o que era suposto ter em atenção aquando do uso da aplicação e durante o preenchimento do inquérito.

Este guião foi entregue na parte final do projeto aos desenvolvedores e clientes finais, com a finalidade de conseguir que fizessem um percurso no *website*.

Foi ainda entregue, durante o desenvolvimento, a um utilizador final (Léonie Watson), com o propósito de obter os pontos a que as pessoas invisuais dão mais atenção, conseguindo

recolher algumas melhorias que foram entretanto colmatadas até aos outros testes finais. Este guião foi distribuído de forma verbal, para que o processo de testes e interpretação do que é o projeto e o seu objetivo, fosse simplificado e claro.

Na seguinte imagem é possível verificar o guião que foi distribuído pelos *stakeholders*:

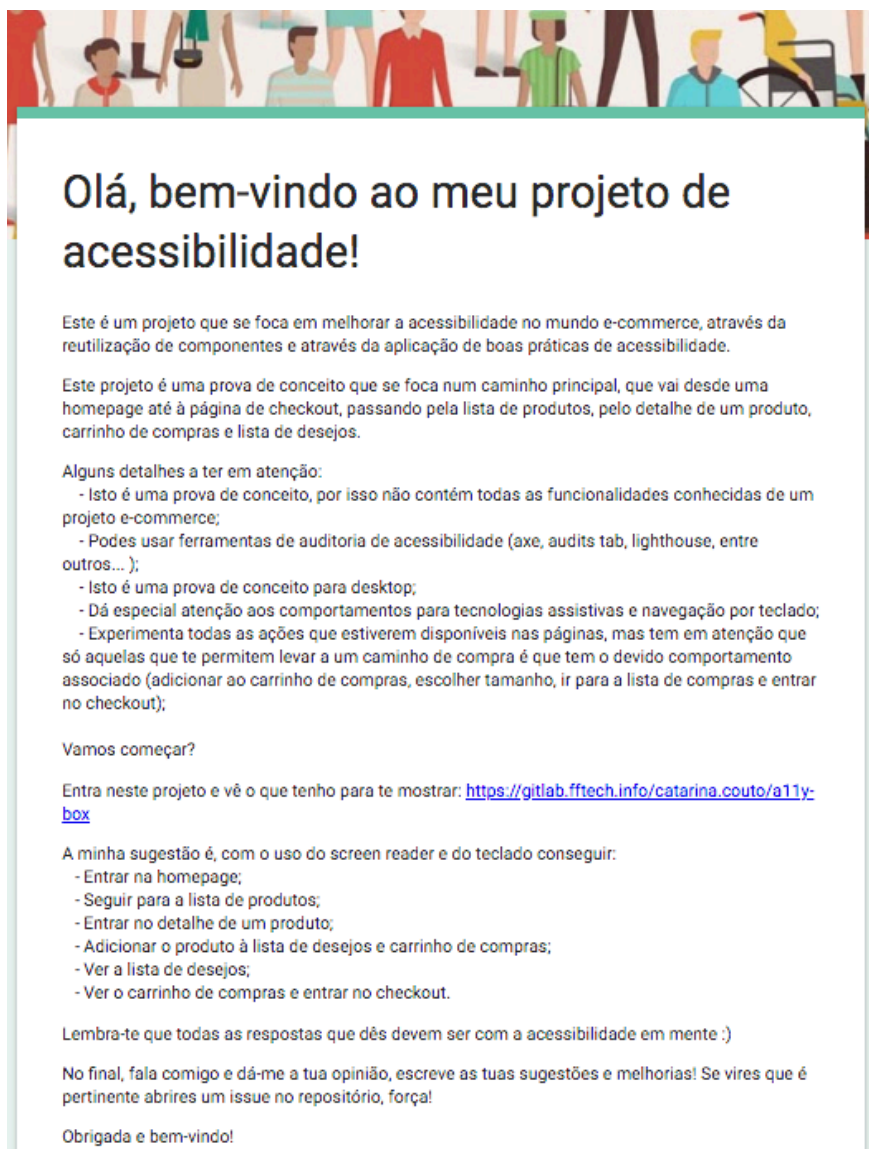


Figura 40 – Guião distribuído pelos *stakeholders*

### 6.2.3 Inquéritos

Para perceber a satisfação, por parte dos desenvolvedores e dos clientes finais, foi disponibilizado um inquérito, com perguntas (cf. secção 6.3.2) sobre as várias áreas do protótipo disponibilizado.

Este inquérito foi entregue na parte final do projeto aos desenvolvedores e clientes finais, com a finalidade de recolher a satisfação sobre o percurso no *website* até atingir o momento de compra, de forma acessível.

Para a realização dos inquéritos e apuração de respostas acerca do trabalho desenvolvido, foi escolhido um tipo de inquérito simples e económico que permitisse acompanhar as opiniões e sugestões dos inquiridos. Estes inquéritos têm a vantagem de se permitir obter a informação necessária dos clientes e utilizadores da aplicação sobre o produto que estão a utilizar (Mateus, 2017). Os inquéritos possuem as seguintes características:

- Curtos, simples e objetivos para levar o inquirido a responder ao essencial;
- Poucas perguntas por página, para evitar a distração do inquirido;
- Estimativa de tempo de resposta entre 30 a 45 minutos, visto que é necessário realizar alguns testes de usabilidade;
- Realizados com suporte *online*, para melhor organização de respostas e elaboração de resultados, com recursos a plataformas existentes, como é o caso do *Google Forms*;
- Inquéritos mistos, ou seja, contêm respostas de questão aberta (o inquirido tem total liberdade em estruturar a resposta) e respostas de questão fechada (o inquirido está limitado às opções de resposta apresentadas no questionário) (Mateus, 2017).

#### 6.2.3.1 Público-alvo

Na conceção de um questionário e de avaliação de uma aplicação através de testes um dos aspetos a ter em consideração é o público-alvo. A sua correta determinação leva a que todas as perguntas e processos de teste sejam pensados com pormenor e dirigidos ao público com a linguagem e conceitos mais corretos e apropriados.

Nesta secção serão identificadas as características do público-alvo de cada parte interessada. O facto de alguns dos públicos-alvo inquiridos já mostrarem familiarização com termos na área de informática, transforma os métodos de avaliação num estudo mais objetivo e mais rigoroso, pois permite obter respostas com carácter mais crítico, o que se torna uma vantagem no sentido de apurar melhorias à aplicação e retirar conclusões sobre o trabalho realizado.

##### 6.2.3.1.1 Desenvolvedores

O público-alvo que diz respeito aos desenvolvedores faz parte da equipa na qual o projeto está inserido, isto é, da equipa *Black & White*.

O interesse deste público-alvo é perceber se a organização e reutilização de código e algoritmos, com especial atenção a acessibilidade, foi conseguida e efetuada com sucesso.

Todos os intervenientes possuem as mesmas características:

- Idades entre os 22 e 40 anos;

- Desenvolvedores (várias categorizações: juniores, intermédios e séniores) pertencentes à equipa *Black & White* ou equipas que pertencem à *Farfetch* e contribuem para os seus processos;
- Interesse em acessibilidade;
- Interesse em tecnologia;
- Familiarização na área de informática;
- Origem Portuguesa;
- Idioma português.

#### 6.2.3.1.2 Clientes finais

O público-alvo dos clientes finais, são os utilizadores finais da aplicação. Neste caso os utilizadores finais da aplicação são os utilizadores internos (colaboradores da organização) e utilizadores externos com uso de tecnologias assistivas.

O interesse deste público alvo é verificar se consegue aceder a um *website* que cumpra certos requisitos de acessibilidade, conseguindo usufruir de uma navegação acessível e fácil.

Todos os intervenientes possuem as mesmas características:

- Idades entre os 22 e 40 anos;
- Desenvolvedores (várias categorizações: juniores, intermédios e séniores) pertencentes à equipa *Black & White* ou equipas que pertencem à *Farfetch* e contribuem para os seus processos;
- Verificadores de qualidade de aplicações – *testers/quality assurance* (várias categorizações: juniores, intermédios e séniores) pertencentes à equipa *Black & White* ou equipas que pertencem à *Farfetch* e contribuem para os seus processos;
- Pessoas incapacitadas visualmente/fisicamente que são utilizadores externos;
- Interesse em acessibilidade;
- Interesse em tecnologia;
- Familiarização na área de informática;
- Origem Portuguesa;
- Idioma português.

#### 6.2.3.1.3 Cliente/loja

O público-alvo do cliente/loja, são os representantes do segmento de clientes deste projeto (cf. secção 3.7).

Visto que o interesse deste público-alvo é perceber em que medida o *website*/protótipo final é acessível e em que medida foi vantajoso o uso de acessibilidade no seu projeto, não serão realizados métodos de avaliação nem serão realizadas com pessoas incluídas no processo.

O possível *website* do cliente/loja, vai ser testado a nível de acessibilidade, seguindo a ótica e interesses do cliente/loja - verificar se a acessibilidade está ou não bem implementada na solução final e se assegura o cumprimento de lei e o uso de tecnologias assistivas.

## 6.3 Avaliação estatística/objetiva

Nesta secção são apresentados os resultados obtidos a partir do inquérito e da interação com a Léonie Watson, apresentando as respetivas respostas.

### 6.3.1 *Feedback* com a Léonie Watson

Léonie Watson começou a explorar o mundo da *web* em 1993, e transformou este mundo na sua carreira profissional em 1997 (apesar de ter começado a perder a visão ao longo deste espaço temporal), contribuindo para a comunidade de acessibilidade e boas práticas na *web*.

Léonie Watson é parte integrante de projetos que são referência no mundo *web*, como o *The Paciello Group (TPG)*, *W3C Advisor Board*, *W3C Web Platform Working Group*, onde é responsável por muitas especificações e decisões, incluindo os atributos ARIA em HTML e a API de acessibilidade - *HTML Accessibility API Mappings (AAM)*. Contribui ainda para vários projetos como o *Inclusive Design Principles* e coorganiza a conferência #ID24 – *Inclusive Design 24* em conjunto com os membros do TPG. Já participou na escrita técnica de alguns livros sobre acessibilidade e agora é *Senior Accessibility Engineer* no TPG, contribuindo em forma de consultoria para melhorar alguns projetos e providenciar *workshops* a grupos/empresas que estejam interessados em melhorar os projetos a nível de acessibilidade.

Hoje em dia, mesmo tendo perdido completamente a visão, Léonie Watson é completamente adepta das tecnologias assistivas, uma referência no mundo da *web*, especialmente em acessibilidade e é frequentemente encontrada em conferências abordando diversos temas como os melhores padrões *web* a usar e práticas para definir um *design* inclusivo (com tecnologias como *SVG*, *HTML*, *ARIA* e *JavaScript*).

Foi assim que conheci a Léonie e que a mesma se tornou uma referência para este projeto. A Léonie Watson veio ao Porto a 4 e 5 de Setembro de 2018 não só com o propósito de providenciar *workshops* de acessibilidade para a organização, mas também de rever o presente projeto e conseguir ajudar com boas práticas para o domínio *e-commerce*, ajudando com a sua perspectiva como pessoa com incapacidades visuais permanentes e como utilizadora de um leitor de ecrã.

Na primeira fase foram abordadas perguntas sobre todas as páginas de um projeto *e-commerce* e que cuidados é que cada uma deve ter a nível de acessibilidade. Estes cuidados/boas práticas sugeridas podem ser consultadas na tabela seguinte, onde é possível verificar se tal requisito está implementado ou se ficará para trabalho futuro:

Tabela 12 – Boas práticas sugeridas por Léonie Watson

Nº	Boa prática sugerida	Implementado	Trabalho futuro
1	Todas as páginas devem possuir um <i>heading</i> de nível 1 (cf. secção 5.1.9).	Sim	Não
2	A lista de produtos do <i>bag</i> deve ter em atenção o número de produtos. Se for menor que 3 devem ser apresentados de forma individual, com recurso a um <i>ProductItem</i> (cf. secção 5.3.5).	Sim	Não
3	A lista de produtos do <i>bag</i> deve ter em atenção o número de produtos. Se for maior que 3 devem ser apresentados em forma de tabela.	Não	Sim
4	Na página de <i>bag</i> , o sumário deve ser apresentado em formato de tabela para usufruir das vantagens do leitor de ecrã nestes casos (cf. secção 2.4.1).	Sim	Não
5	Os botões da lista de desejos devem ter um nome acessível e descritivo, incluindo o nome do produto em questão (cf. secção 5.3.4).	Sim	Não
6	Os botões do carrinho de compras devem ter um nome acessível e descritivo, incluindo o nome do produto em questão (cf. secção 5.3.5).	Sim	Não
7	A lista de desejos deve ter uma opção para mover o produto para o carrinho de compras.	Não	Sim
8	Os <i>loading</i> das páginas devem anunciar “ <i>Loading</i> ” e “ <i>Loading completed</i> ” (cf. secção 5.2.8).	Sim	Não
9	Todas os dados que mudam dinamicamente com as ações do utilizador e que devem ser notificados devem possuir uma boa descrição textual e o atributo <i>aria-live</i> .	Sim	Não
10	Todas as páginas devem ter um título de página semelhante ao <i>heading</i> de nível 1 presente na mesma (cf. secção 5.1.9).	Sim	Não
11	Possuir uma estrutura lógica de <i>headings</i> (cf. secção 5.1.9).	Sim	Não
12	Na página de produto a descrição deve ser	Não	Sim

Nº	Boa prática sugerida	Implementado	Trabalho futuro
	pormenorizada.		
13	Na página de produto as imagens devem possuir um texto alternativo que descreva objetivamente a imagem.	Parcialmente, porque as imagens são dinâmicas, tal como a sua descrição que deve ser melhorada pela equipa de conteúdo.	Sim
14	O <i>footer</i> deve ter um <i>link</i> para a página sobre <i>accessibility statements</i> .	Sim	Não
15	O <i>website</i> deve ter uma página exclusiva para <i>accessibility statements</i> (cf. secção 5.5).	Sim	Não

Segundo a Léonie Watson, as boas práticas apresentadas na tabela acima protagonizam as funcionalidades mais utilizadas e procuradas pelos utilizadores invisuais. Algumas delas já foram abordadas como pontos fulcrais a nível de acessibilidade, quer no estado de arte (cf. secção 2.3) quer na implementação do protótipo (cf. secção 5.1).

Com a observação da tabela é possível perceber que existem boas práticas que estão implementadas no projeto e outras que não foram implementadas, ou foram implementadas parcialmente por estratégias de prova de conceito ou por limitações encontradas:

- Nº 3 – Não implementada porque o foco da página para o carrinho de compras neste projeto era conseguir iniciar o processo de compra e por isso, a lista de produtos desta página foi sempre apresentada da mesma forma, com recurso ao componente *ProductItem*. No entanto, esta boa prática deve ser considerada como trabalho futuro no sentido de melhorar a acessibilidade para as tecnologias assistivas com o uso de tabelas para quando o número de produtos é superior a 3;
- Nº 7 – Não implementada porque o foco da página para a lista de desejos era garantir o uso de todas as boas práticas gerais (cf. secção 5.1), considerando que esta funcionalidade seria pouco relevante para o fluxo da aplicação. No entanto, caso fosse implementada seria muito parecida a um botão da página atual, ou ao *link* presente na lista de desejos com a função de eliminar o produto da *wishlist*. Ainda assim, ao implementar esta funcionalidade no futuro é necessário ter em conta que para mover um produto para o *bag*, o mesmo tem que possuir quantidade e tamanho associados, o que leva à implementação de mais dois componentes *Select* (cf. secção 5.2.5) para escolher estas propriedades, muito à semelhança do que acontece na página para o *bag*;
- Nº12 e Nº13 – Tanto as descrições dos produtos como o texto alternativo associado às imagens do mesmo são dinâmicas e carecem de conteúdo que forneça uma boca

descrição textual. Assim, como trabalho futuro tanto as descrições dos produtos, como os textos alternativos das imagens devem ser melhoradas/trabalhadas para atingir melhores resultados.

Conforme a avaliação da Léonie Watson e transcrevendo as suas palavras, todas as suas compras e tarefas quotidianas com recurso às tecnologias assistivas na *web* e é aqui, ao fazer compras *online*, que se debate com grandes barreiras. No entanto, ao testar o projeto atual, ela conseguiu atingir o momento de compra com recurso ao leitor de ecrã e partilhou que a experiência do *website* foi muito boa, afirmando que se fosse um projeto já publicado na *web* ela compraria no mesmo. Foram poucas as ressalvas que a mesma fez a nível de melhorias, que já foram colmatadas:

- A descrição textual do *skip link* para o conteúdo geral da *homepage* (*banner*) precisava de ser melhorada, na medida em que não providenciava informação relevante e útil para o *screen-reader*;
- Para os utilizadores que navegam a utilizar o teclado era útil incluir o componente para o comportamento de paginação (cf. secção 5.2.7) no início da lista de produtos para facilitar a sua descoberta e uso;
- Rever os requisitos que a mesma considera importantes e tentar implementar ou solucionar da melhor forma.

Por outro lado, a mesma referiu que apreciou alguns cuidados que estavam impressos na aplicação:

- no *header* - a nível de *Skip Links* e do formulário para a pesquisa;
- no *footer* - por existir o *link* para a página de *accessibility statements*;
- na página de lista de produtos - navegação entre produtos através dos *skip links* e da apresentação de informação complementar para os *screen-readers*;
- no *checkout* – pela forma como o elemento sumário está dentro de um elemento HTML *table* facilitando a leitura de informação;
- no *bag* e na *wishlist* – pela forma como os produtos e informação dos mesmos é apresentada;
- em toda a aplicação - pelos nomes acessíveis dos botões, pelas notificações ao mudar de página e ao adicionar produtos ao *bag* e à *wishlist*, pelos *page title* associados a cada página e pelos *landmarks* e *headings* bem estruturados e com descrições.

Finalmente, a Léonie Watson, para além de dizer que utilizaria a prova de conceito presente no projeto se o mesmo fosse um projeto publicado, disse ainda que o trabalho realizado quer a sensibilizar as pessoas da equipa para o tema acessibilidade, como a isolar componentes com o intuito de providenciar bases para o tema para todos os projetos foi muito bem conseguido e executado com muito mérito.

### 6.3.2 Resultados de inquéritos

Foram realizadas dezassete perguntas a 14 pessoas: 11 perguntas numa escala de 1 a 5 (nada satisfeito a muito satisfeito, respetivamente) que avaliam a satisfação do utilizador em diversas áreas, 2 perguntas de sim/não como resposta para perceber se o utilizador conseguiu atingir o momento de compra através do teclado e do *screen-reader*, 4 perguntas de resposta aberta sobre que ferramentas de auditoria foram utilizadas, quais os resultados obtidos relativamente às ferramentas de auditoria, quais os *browsers* utilizados para testar a aplicação e a última para obter sugestões de melhoria ou observações.

Para cada pergunta associada a uma escala foram elaborados gráficos com o intuito de melhor se compreender os resultados obtidos de uma forma geral e para as outras foi feito um apanhado de todas as respostas, conseguindo chegar a conclusões de limitações e trabalho futuro.

Qual é o seu grau de satisfação (0 a 5) no *header*?

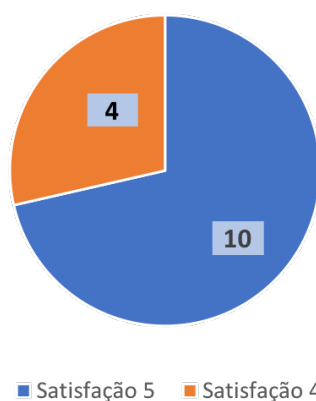


Figura 41 – Inquérito Pergunta 1: Grau de satisfação no *header*

Qual é o seu grau de satisfação (0 a 5) no *footer*?

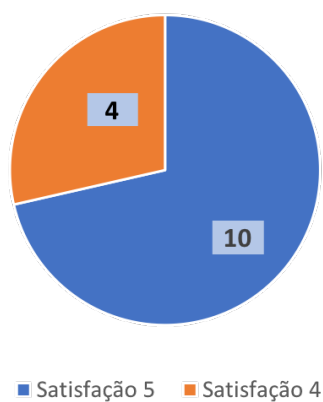


Figura 42 – Inquérito Pergunta 2: Grau de satisfação no *footer*

Qual é o seu grau de satisfação (0 a 5) na *homepage*?

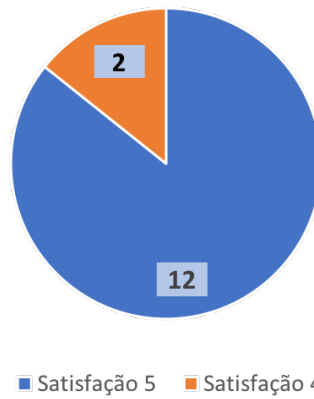


Figura 43 – Inquérito Pergunta 3: Grau de satisfação na *homepage*

Qual é o seu grau de satisfação (0 a 5) na lista de produtos?

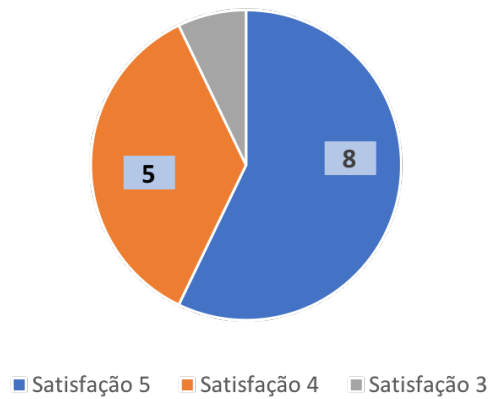


Figura 44 – Inquérito Pergunta 4: Grau de satisfação na lista de produtos

Qual é o seu grau de satisfação (0 a 5) no detalhe do produto?

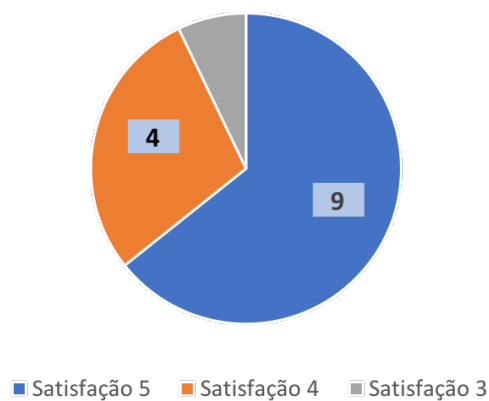


Figura 45 – Inquérito Pergunta 5: Grau de satisfação no detalhe do produto

Qual é o seu grau de satisfação (0 a 5) na *wishlist*?

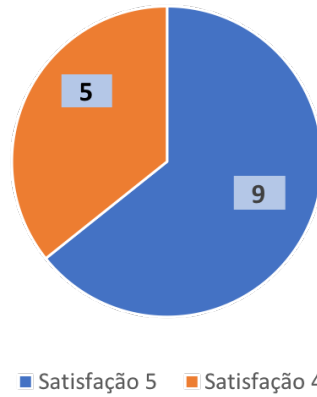


Figura 46 – Inquérito Pergunta 6: Grau de satisfação na *wishlist*

Qual é o seu grau de satisfação (0 a 5) no *bag*?

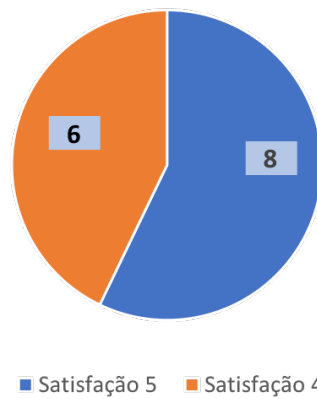


Figura 47 – Inquérito Pergunta 7: Grau de satisfação no *bag*

Qual é o seu grau de satisfação (0 a 5) no *checkout*?

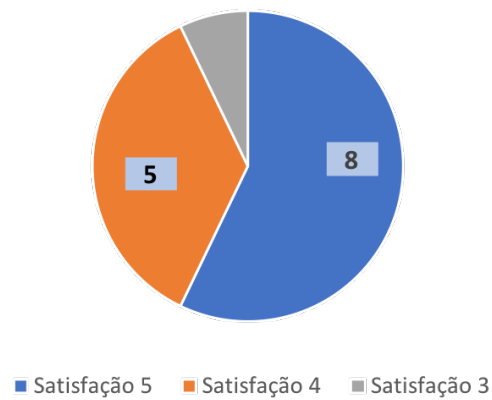


Figura 48 – Inquérito Pergunta 8: Grau de satisfação no *checkout*

Qual é o seu grau de satisfação (0 a 5) com o fluxo do uso do *screen-reader* do projeto?

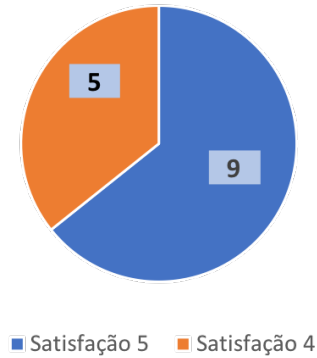


Figura 49 – Inquérito Pergunta 9: Grau de satisfação com o fluxo do *screen-reader* do projeto

Qual é o seu grau de satisfação (0 a 5) com o fluxo do uso do teclado no projeto?

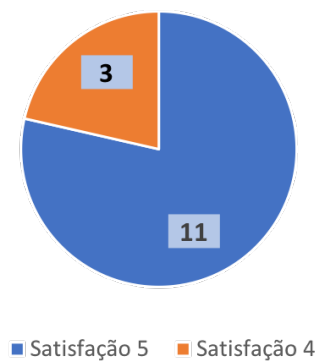


Figura 50 – Inquérito Pergunta 10: Grau de satisfação com o fluxo do uso do teclado no projeto

Que ferramentas de auditoria que usou para testar o projeto?

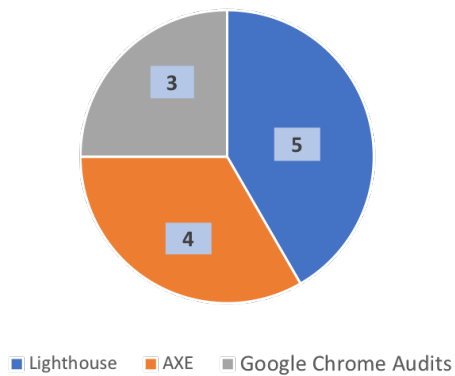
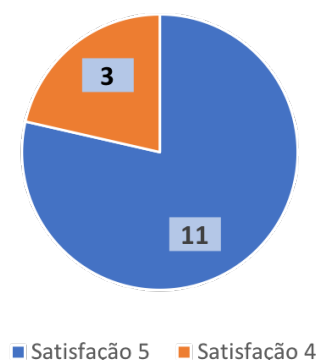


Figura 51 – Inquérito Pergunta 11: Ferramentas de auditoria utilizadas para testar o projeto

À pergunta nº12, sobre quais foram os **resultados que obtiveram ao testar o projeto com ferramentas de auditoria** e a opinião sobre os mesmos, foram obtidas várias respostas:

- ⇒ Os problemas reportados foram coisas pequenas – falta de estrutura lógica de *headings* em algumas páginas;
- ⇒ No cômputo geral está num nível de acessibilidade bastante elevado;
- ⇒ Os resultados foram extremamente positivos, tendo como único ponto a assinalar o contraste de cores dos *links* do *header* (os que não estão ativos no momento): a cor cinza parece ser muito próxima à cor do fundo;
- ⇒ O projeto atingiu a pontuação máxima: 100% em acessibilidade e sem falhas de semântica ou SEO, confirmam o cuidado que existiu em garantir acessibilidade;
- ⇒ O projeto atingiu resultados com várias ferramentas de auditoria muito positivos: sempre acima dos 80%.

Qual é o seu grau de satisfação (0 a 5) com o estado final do projeto?



À pergunta nº13, sobre se os utilizadores tinham **sugestões de melhoria**, foram recolhidas algumas respostas:

- ⇒ Incorporar uma estratégia de retroceder para as páginas anteriores;
- ⇒ Melhorar o contraste na página de produtos:
- ⇒ A estrutura de *headings* devia ser dinâmica por página;
- ⇒ Os botões associados a ações de adicionar e remover produtos deviam conter o nome do produto na sua descrição;
- ⇒ O botão “seguinte” na paginação devia anunciar “Página seguinte” para ser mais descritivo;
- ⇒ O valor do atributo `aria-labelledby="product-wishlist"` dos produtos da lista deve ser único;

- ⇒ Quando dá erro ao adicionar um produto ao *bag* sem tamanho, devia anunciar o erro;
- ⇒ Os valores do sumário do *bag* e do *checkout*, na coluna direito não anunciam o conteúdo da forma correta;
- ⇒ O botão de ir para o *checkout* no *bag*, devia anunciar o preço total na sua descrição;
- ⇒ As mensagens de elementos e páginas em loading devem ser melhoradas;
- ⇒ Em termos gerais, a notificação para o *screen-reader* das transições entre páginas pode ser melhorada.

À pergunta nº14, sobre se os utilizadores tinham **observações**, foram recolhidas algumas respostas:

- ⇒ Os *skip links* funcionam na perfeição permitindo navegar entre as diferentes zonas de conteúdo;
- ⇒ Considero o resultado muito positivo, pois consegui fazer todo o fluxo de compra sem qualquer problema, com recurso ao teclado e ao *screen-reader* de uma forma bastante fácil concluindo que este projeto se traduz numa prova de conceito bem conseguida;
- ⇒ É importante destacar o cuidado existente com a informação que é anunciada aos *screen-readers* no projeto, no sentido que a maior parte dos elementos tem descrições associadas de forma a melhorar o seu entendimento;
- ⇒ A forma de abordar a navegação entre produtos é bastante inovadora e criativa. Melhora consideravelmente a eficácia de navegação na lista de produtos;
- ⇒ As descrições detalhadas de um produto em modo colapsável estão muito bem executadas pois dão indicação se o item está expandido ou colapsado;
- ⇒ Um dos principais problemas em SPA, em termos de acessibilidade, é o facto de na transição de páginas o atributo `<title>` não ser alterado. Neste projeto, isto não se verifica;
- ⇒ Muito fácil e intuitivo navegar pela loja.

Em seguida são mostradas algumas respostas que achei que seria importante revelar, visto que se demonstram desenquadradas das restantes, demonstrando alguma falta de conhecimento por parte dos utilizadores ao realizar os testes, visto que não conhecem todas as funcionalidades das tecnologias assistivas:

- ⇒ No *bag*, não é possível aceder ao sumário da compra, só a cada produto individualmente;
- ⇒ Apesar de ter pouca experiência na utilização deste tipo de tecnologias assistivas, consegui navegar pela loja;

- ⇒ Não consegui ler muitos elementos com o *screen-reader*, não conseguindo que os mesmos fossem anunciados.

Para finalizar, é importante mostrar as respostas da pergunta nº15, sobre a **opinião geral do projeto, a sua importância, impactos positivos e negativos** associados e observações:

- ⇒ Este projeto pode e deveria servir de base para qualquer plataforma *e-commerce* que se preocupe em providenciar a melhor experiência aos seus utilizadores. Este projeto tem exemplos criativos de como abordar problemas simples, bom contraste de cores, entre outros;
- ⇒ Considero que este projeto é bastante importante, porque acho que é fundamental qualquer pessoa conseguir navegar em sites *de e-commerce* e conseguir efetuar uma compra sem dificuldades;
- ⇒ Este trabalho aumenta a visibilidade e noção sobre o que é a acessibilidade e o que se pode fazer para melhorar o acesso de todos os utilizadores a qualquer *website* ou serviço na *internet*. Com estas implementações o público-alvo será mais extenso o que significará provavelmente mais vendas.
- ⇒ Este trabalho é muito importante, pois para além de conseguirmos com que pessoas com dificuldades consigam usar as plataformas de *e-commerce* também conseguimos criar uma experiência melhor para quem usa exclusivamente tecnologias assistivas.
- ⇒ É certo que a aplicação dos padrões deste protótipo em projetos de contexto real terá um impacto muito positivo. É dos projetos mais importantes que tive oportunidade de testar e curiosidade de aprender mais. O valor que confere a qualquer projeto *web* é imensurável, a qualidade de execução é exemplar e a facilidade de uso surpreendente.
- ⇒ É um projeto importante e sensibilizador. Obrigada pela dedicação e esforço.
- ⇒ Este projeto é extremamente importante para uma empresa global. Temos que ser inclusivos e facilitar a compra a todos. Com as boas práticas espelhadas neste projeto é mais fácil ter diretrizes para o desenvolvimento dentro da organização.
- ⇒ No geral, está um trabalho excelente! Estou bastante satisfeito, consegui chegar ao processo de *checkout* sem utilizar o rato. Consegui selecionar o tamanho do produto pretendido, naveguei por várias páginas até encontrar o produto que mais me aliciou, adicionei-o à *wishlist*, e posteriormente cheguei ao processo de *checkout* sem qualquer problema.
- ⇒ Tenho de confessar que estou impressionado com a facilidade com que consegui fazer as tarefas. O projeto está de facto muito bom e cumpre tudo ao que foi proposto. Excelente iniciativa e ótimo trabalho. Parabéns!
- ⇒ É fundamental não menosprezarmos a acessibilidade na *web*. O facto de uma pessoa possuir uma incapacidade não pode ser um entrave ao usar qualquer tipo de interface, no mínimo para as interações mais básicas. Daí que este projeto tenha a

maior das importâncias nem que seja para dar o mote para que se criem mais iniciativas neste tema. Está um excelente trabalho.

Depois de realizados os inquéritos, a um total de 14 pessoas (11 *Developers* e 3 *Quality Assurance*), foi necessário auferir a eficácia dos mesmos e do modelo de avaliação.

Os resultados da análise estatística dos dados relativos ao questionário serão realizados nas seguintes categorias:

- Satisfação do uso das boas práticas de acessibilidade;
- Satisfação do fluxo da aplicação com o uso de tecnologias assistivas;
- Satisfação e resultados através do uso de ferramentas de auditoria;
- Vontade do utilizador e satisfação geral.

A escala utilizada nas perguntas dos inquéritos é de 0 a 5 em que os valores fazem corresponder: 1 – nada satisfeito, 2 – pouco satisfeito, 3 – razoavelmente satisfeito, 4 – satisfeito e 5 – muito satisfeito.

### **6.3.3 Satisfação do uso das boas práticas de acessibilidade**

Quanto ao conteúdo do projeto e objetivos, este ponto é o mais fulcral e importante de estudo pois avalia a satisfação do mesmo como um todo, tendo em conta o cumprimento do objetivo geral do projeto: providenciar acessibilidade em projetos *web*.

Para a avaliação neste parâmetro foram estudadas as perguntas 1) a 8) explicitadas na secção 6.3.2. Quando questionados sobre a satisfação dos elementos *header* e *footer*, de um modo geral, 10 inquiridos (71,4%) afirmam que estão muito satisfeitos e 4 afirmam que estão satisfeitos (28,6%). É interessante verificar que os utilizadores ficaram agradados com o produto final apresentado, tendo 12 respondido que estão muito satisfeitos (85,7%) com a página de homepage, sendo que os outros 2 estão satisfeitos (14,3%). Oito dos inquiridos (57,1%) diz estar muito satisfeito com o produto final da lista de produtos, 5 satisfeitos (35,7%) e um utilizador diz estar apenas razoavelmente satisfeito, não existindo para esta pergunta mais respostas nas outras categorias. Podemos constatar que 9 inquiridos (64,3%) assegura que está muito satisfeito com o aspeto final da página de detalhe de um produto, 4 estão satisfeitos (28,6%) e apenas um razoavelmente satisfeito. No que diz respeito à página de *wishlist* os resultados são muito positivos: 9 inquiridos (64,3%) dizem estar muito satisfeitos e os restantes 5 ficaram satisfeitos (35,7%). Na hora de responderem sobre a satisfação com a página do *bag* os resultados foram animadores: 8 afirmam que estão muito satisfeitos (57,1%) e 6 afirmam estar satisfeitos (42,9%). Para finalizar, quando falamos da satisfação na página de *checkout*, 8 (57,1%) inquiridos demonstram ter ficado muito satisfeitos, 5 (35,7%) satisfeitos e um razoavelmente satisfeito.

Estes resultados mostram que o projeto foi bem conseguido e que atingiu valores, junto dos utilizadores, de um modo geral, positivos, tendo os utilizadores ficado satisfeitos o mesmo e com as boas práticas implementadas ao longo das páginas.

#### **6.3.4 Satisfação do fluxo da aplicação com o uso de tecnologias assistivas**

Para a avaliação deste parâmetro foram estudadas as perguntas 9) e 10) que são sobre a satisfação com o fluxo do uso do *screen-reader* para testar o projeto 9 dos 14 inquiridos (64,3%) afirma está muito satisfeito e 5 dizem estar satisfeitos (35,7%). Após esta análise, concluiu-se o fluxo do uso do *screen-reader* na aplicação foi bem aceite pelos utilizadores, visto que todas as respostas foram positivas. Quando inquiridos sobre a satisfação com o fluxo do uso do teclado para testar o projeto, os resultados também foram positivos, pois 11 afirmam que estão muito satisfeitos (78,6%) e os restantes 3 dizem estar satisfeitos (21,4%). Estes resultados são ótimos porque face ao esforço impresso no projeto para providenciar elementos de acessibilidade para que o funcionamento com as tecnologias assistivas fosse natural e melhorado, as opiniões recolhidas foram ao encontro das expectativas. No entanto, foram recolhidas algumas respostas de escrita livre, nas quais é possível observar que alguns dos membros do público-alvo não possuem formação/conhecimento suficiente para conseguir utilizar as tecnologias assistivas e todas as suas funcionalidades, constituindo este facto para ser uma limitação do projeto.

#### **6.3.5 Satisfação e resultados através do uso de ferramentas de auditoria**

Para a avaliação deste parâmetro foram estudadas as perguntas 11) e 12) que são sobre a satisfação através do uso de ferramentas de auditoria para testar o projeto, apenas 13 dos inquiridos realizou estes testes. Destes, 5 dos 13 afirma ter utilizado o *Lighthouse* como ferramenta de auditoria, 4 dos que responderam a esta pergunta usaram a ferramenta *AXE* e os restantes 3 deram uso ao *Google Chrome Audits*. As respostas de escrita livre dadas acerca dos resultados obtidos com as ferramentas de auditoria, demonstram que o projeto inclui vários cuidados de acessibilidade, uns mais básicos e outros mais complexos, com o propósito de cumprir algumas diretrizes do tema, conseguindo fornecer aos utilizadores melhores formas de navegar e utilizar um *website*.

#### **6.3.6 Vontade do utilizador e satisfação geral**

Nas questões 13), 14) e 15) em que os inquiridos podiam dar sugestões de melhorias e funcionalidades que gostariam de ver desenvolvidas no futuro houve muita adesão e colaboração por parte dos mesmos.

Muitas das sugestões de melhoria e observações fornecidas ao longo de todas as respostas constituem novas funcionalidades e comportamentos a nível de acessibilidade, que devem ser consideradas como trabalho futuro, visto não ter sido possível incluir as mesmas no tempo de vida da prova de conceito inerente ao projeto apresentado neste documento. A par disto, a opinião geral dos inquiridos sobre o projeto é que o mesmo é uma grande contribuição para a organização e para a equipa, pois pode ser visto como base e referência para outros projetos já que cumpre muitos requisitos e diretrizes de acessibilidade.

Sobre esta característica de avaliação, os resultados também foram positivos e foram recebidos muitas críticas/sugestões construtivas contribuindo assim para que, no futuro, seja realizado um trabalho melhor e mais rico nos conteúdos que o utilizador deseja ver satisfeitos e melhorados.



## 7 Conclusões e trabalho futuro

Neste capítulo são apresentadas conclusões acerca do projeto realizado, bem como uma descrição de todos os objetivos atingidos no projeto e todas as limitações e trabalho futuro, visando algumas melhorias na aplicação desenvolvida.

Por fim, é realizada uma apreciação final acerca do projeto como um todo e a aprendizagem obtida a nível pessoal.

### 7.1 Objetivos alcançados

O grande objetivo deste projeto foi a pesquisa e implementação de boas práticas de acessibilidade na web, a nível de projetos *e-commerce*. Esta implementação assenta na criação e reutilização de componentes que na sua essência já são acessíveis permitindo que aquando da sua reutilização esta base já esteja garantida.

Esta implementação de acessibilidade no protótipo tinha como principal objetivo conseguir chegar ao fluxo de compra com uso exclusivo de teclado e do *screen-reader*, como tecnologias assistivas, começando na *homepage*, passando pela lista de produtos e detalhe dos mesmo, *bag* e *wishlist*, conseguindo chegar ao *checkout*.

O pressuposto deste projeto era pesquisar boas práticas e implementá-las num protótipo garantindo a possibilidade de chegar ao momento de compra. Este objetivo considera-se cumprido com sucesso, uma vez que foram implementadas todas as páginas supracitadas em conjunto com várias técnicas de acessibilidade. Com o protótipo não só foram criados componentes, como também foram isolados, estando neste momento em utilização em diferentes projetos para clientes finais, garantindo a sua reutilização, facilidade de uso e manutenibilidade.

O trabalho desenvolvido resultou na concretização de um protótipo com componentes e adoção de boas práticas de acessibilidade em diferentes páginas de um *website e-commerce*, respondendo assim ao objetivo proposto na secção 1.3. Com isto, é importante ainda salientar que este projeto fica como referência futura para desenvolvimento e irá servir como projeto de testes para a equipa não só para experimentar novas abordagens de acessibilidade,

antes de serem incluídas em projetos reais de clientes, como para promover a formação e sensibilização do tema acessibilidade para a equipa.

Em suma, todos os objetivos definidos para este projeto foram alcançados, assim como todas as tarefas ao longo do tempo foram concretizadas.

## 7.2 Limitações

Uma vez que este projeto incide sobre a implementação de acessibilidade em *websites e-commerce*, é necessário que toda a equipa sinta a responsabilidade e necessidade do tema, conseguindo implementar diferentes abordagens e, conseqüentemente, testá-las. Este comportamento não existe de forma exemplar na equipa, na medida em que existe pouco conhecimento e interesse em diversas áreas (desenvolvimento, testes e gestão) no tema, verificando-se uma notória dificuldade quando confrontados com o presente projeto. Ao testar o projeto, com recurso a tecnologias assistivas, foi notável uma particular inexperiência (cf. secção 6.3.4 e 6.3.5) sobre o que é que testar acessibilidade implica e inclui.

Outra limitação encontrada está relacionada com o conteúdo dinâmico, nomeadamente as imagens e produtos que carecem de boa descrição e texto alternativo, levando a que do lado do projeto seja impossível providenciar melhores resultados.

## 7.3 Trabalho futuro

Apesar do protótipo apresentado disponibilizar já algumas das funcionalidades mais importantes no comércio *online*, existem outras páginas e funcionalidades que podem ser desenvolvidas e prevê-se que no futuro sejam implementadas no mesmo projeto utilizando o mesmo design arquitetural preconizado neste projeto.

Todas as melhorias sugeridas fruto dos testes e inquéritos devem ser tomadas em consideração e serem adotadas de acordo com os princípios da aplicação. No que diz respeito à apuração da satisfação dos utilizadores, esta foi feita junto dos utilizadores da equipa (colaboradores) e pessoas com incapacidades (externas à equipa), pelo que devem ser feitas mais avaliações quando mais funcionalidades estiverem disponíveis para obter um *feedback* real, de alguém que necessite de utilizar estas tecnologias.

Pondo isto em prática seria necessário implementar/melhorar algumas funcionalidades que são fruto dos inquéritos e ideias que surgiram ao longo do projeto que são consideradas uma mais valia para a organização:

- Estratégia para disponibilizar uma sequência de *headings* dinâmica, por página;
- Isolamento dos componentes criados no protótipo (cf. secção 5.2);
- Estudo da plataforma Azure para reconhecimento de conteúdo de imagens para determinar textos alternativos de forma automática;

- Implementação do comportamento de novas funcionalidades/componentes (paginação, filtros, pesquisa, conteúdo editorial, etc.);
- Incluir nas *pipelines* de projetos ferramentas de auditoria de acessibilidade garantindo que esta prática é assegurada;
- Definir estratégia para incluir a página de *accessibility statements* (cf. secção 5.5) nos projetos.

## 7.4 Apreciação final e pessoal

Considero que, ao longo do projeto, o meu conhecimento sobre desenvolvimento *web* e acessibilidade tenha aumentado consideravelmente e que aprimorei algumas técnicas que usava, bem como conheci novos métodos de trabalho que desconhecia até então e que hoje considero serem cruciais para projetos desta envergadura e domínio. Tenho completa noção de que aperfeiçoei o meu conhecimento do contexto legal e de negócio, das linguagens utilizadas e que aprendi técnicas indispensáveis para o desenvolvimento deste tipo de projetos.

A nível pessoal, sinto que melhorei bastante a capacidade de trabalho individual e pesquisa, mas também a capacidade de trabalho em equipa. Foram ainda desenvolvidas capacidades de elaboração de documentos relativos a um projeto, como o presente documento, que sistematiza os pressupostos tomados e as decisões tomadas ao longo deste, o que faz com que o mesmo, em conjunto com o repositório do projeto, possa ser uma mais-valia para a empresa no sentido de ser consultado sempre que sejam precisos esclarecimentos acerca das tecnologias, decisões e implementação de funcionalidades.

Sobre o tema, sinto que depois da inclusão deste projeto na equipa, a preocupação com acessibilidade é crescente, visto que já existem planos para futuras formações com o objetivo de promover e sensibilizar os temas que foram abordados neste documento. Este projeto revelou ser um contributo importante, no sentido em que descreve uma nova forma de desenvolver *websites* que atingem níveis de acessibilidade elevados, com recurso a componentes isolados e acessíveis.

Com este protótipo, em que a acessibilidade foi preocupação desde o início, posso afirmar com toda a certeza (conhecendo outros projetos que já iniciaram) que o custo e facilidade são consideravelmente reduzidos. Além disto, ficou claro que desenvolver acessibilidade não precisa de adotar uma abordagem completa desde o início, mas por contraponto, bastando começar com alguns princípios (e.g. legais) e boas práticas (e.g. *atomic design* e restante arquitetura preconizada), e que este é um processo incremental, iterativo e criativo.

O desenvolvimento realizado nesta prova de conceito desmitifica o que é assumido por muitas pessoas e empresas - que a acessibilidade é difícil e leva tempo a implementar. A realidade é que pensar em acessibilidade deve fazer parte do processo de desenvolvimento de um *website* desde o início do mesmo. Isso leva a que a sua implementação seja pensada e calculada de forma a que todos os envolvidos beneficiem. Na realidade, fica mais difícil de atingir um nível de aceitação considerável de acessibilidade quando os projetos já se encontram numa fase mais madura. Assim, pensar em acessibilidade desde o início dos

projetos deve ser uma decisão estratégica de modo a facilitar a sua aplicação de forma faseada e iterativa, conseguindo cumprir as diretrizes associadas ao tema.

No caso deste projeto, que está inserido numa realidade *e-commerce*, é considerado que todos os seus componentes e descobertas sejam aplicadas em todos os projetos e clientes associados à equipa, o que é um aspeto bastante positivo, no sentido em que se torna num projeto base e piloto para outros e que por isso vai atingir uma maior audiência.

Um grande orgulho para a minha pessoa foi conhecer e interagir com a Léonie Watson (cf. secção 6.3.1), além de a ter conseguido trazer para uma formação à organização. Foi toda uma nova perspetiva ver como alguém invisual lida com o quotidiano, de uma forma completamente diferente e inspiradora. O auge do meu orgulho aconteceu quando a Léonie testou o meu projeto, dando algumas dicas e, o mais importante, aprovou o projeto.

Em suma, considero que este projeto foi um contributo bastante positivo não só para mim, como para toda a equipa.

# Referências

Accessible, S. Demystifying accessible name » Simply Accessible. Retrieved October 5, 2018, from <https://simplyaccessible.com/article/accessible-name/>

Apple. (2016). VoiceOver overview - VoiceOver Getting Started Guide. Retrieved February 24, 2018, from <https://help.apple.com/voiceover/info/guide/10.12/#/vo2681>

Atlassian. Saving changes | Atlassian Git Tutorial. Retrieved October 6, 2018, from <https://www.atlassian.com/git/tutorials/saving-changes>

Buchanan, J., & Gardiner, L. (2003). A comparison of two reference point methods in multiple objective mathematical programming. *European Journal of Operational Research*, 149, 17–34. [https://doi.org/10.1016/S0377-2217\(02\)00487-3](https://doi.org/10.1016/S0377-2217(02)00487-3)

Business of Fashion. (2015). BoF Exclusive | Farfetch Opens E-Commerce Platform to Brands | BoF Exclusive, News & Analysis | BoF. Retrieved February 17, 2018, from <https://www.businessoffashion.com/articles/bof-exclusive/farfetch-e-commerce-fashion-brands-omnichannel>

CDC - Centers for Disease Control and Prevention; (2009). Centers for Disease Control and Prevention, 1–5. [https://doi.org/CDC 24/7](https://doi.org/CDC%2024/7)

Census Bureau, U. (2012). Nearly 1 in 5 People Have a Disability in the U.S., Census Bureau Reports. U.S. Census Bureau. Retrieved from <https://www.census.gov/newsroom/releases/archives/miscellaneous/cb12-134.html>

Connor, C. (2014). Color Contrast And Why You Should Rethink It. Retrieved October 11, 2018, from <https://www.smashingmagazine.com/2014/10/color-contrast-tips-and-tools-for-accessibility/>

Cook, A. M.; Hussey, S. M. . (2002). *Assistive Technologies: Principles and Practice*. Cook & Husseys. Debian Accessibility. (2016). Debian -- Debian-Accessibility. Retrieved February 21, 2018, from <https://www.debian.org/devel/debian-accessibility/index.pt.html>

Dolson, J. (2009). Accessibility: How Many Disabled Web Users Are There? | Practical Ecommerce. Retrieved February 19, 2018, from <https://www.practicalecommerce.com/Accessibility-How-Many-Disabled-Web-Users-Are-There>

Farfetch. (2008). Farfetch. Retrieved February 17, 2018, from <https://farfetch.prezly.com/>

Freedom Scientific. (2018). JAWS Screen Reader - Best in Class. Retrieved February 24, 2018, from <http://www.freedomscientific.com/Products/Blindness/JAWS>

Frost, B. (2013). Atomic Design | Brad Frost. Retrieved February 25, 2018, from <http://bradfrost.com/blog/post/atomic-web-design/>

Frost Brad. (2015). Atomic Design. Retrieved from <http://atomicdesign.bradfrost.com/table-of-contents/>

Keijonen, S. (2017). Building an Inclusive Web: Why Accessibility Matters. Retrieved February 4, 2018, from <https://webdesign.tutsplus.com/articles/building-an-inclusive-web-why-accessibility-matters--cms-29773>

Koen, P. A., Ajamian, G. M., Boyce, S., Clamen, A., Fisher, E., Fountoulakis, S., ... Seibert, R. (2001).

FuzzyFrontEnd: Effective Methods, Tools, and Techniques LITERATURE REVIEW AND RATIONALE FOR DEVELOPING THE NCD MODEL. Retrieved from [http://www.stevens-tech.edu/cce/NEW/PDFs/FuzzyFrontEnd\\_Old.pdf](http://www.stevens-tech.edu/cce/NEW/PDFs/FuzzyFrontEnd_Old.pdf)

Leitner, M.-L., & Strauss, C. (2008). Exploratory case study research on web accessibility. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5105 LNCS). [https://doi.org/10.1007/978-3-540-70540-6\\_70](https://doi.org/10.1007/978-3-540-70540-6_70)

Lynch, P., & Horton, S. (2004). *Web Style Guide*. *Web Style Guide*, 2nd, 1–290. Retrieved from <http://www.webstyleguide.com/>

Mateus, J. (2017). A importância dos questionários de satisfação | Joana Mateus | Pulse | LinkedIn. Retrieved February 25, 2018, from <https://pt.linkedin.com/pulse/importância-dos-questionários-de-satisfação-joana-mateus>

MDN Web Docs. (2017). Accessibility | MDN. Retrieved February 19, 2018, from <https://developer.mozilla.org/en-US/docs/Web/Accessibility>

Moore, Lindsay; Mon, Gonzalo; Skelton, C. (2017). Website accessibility: Understanding the ADA for e-commerce. Retrieved February 11, 2018, from <https://www.digitalcommerce360.com/2017/03/10/website-accessibility-understanding-ada-e-commerce/>

Nielsen, J. . (2000). Why You Only Need to Test with 5 Users. Retrieved February 13, 2018, from <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>

Nomensa. How to structure headings for web accessibility | Nomensa. Retrieved October 3, 2018, from <https://www.nomensa.com/blog/2017/how-structure-headings-web-accessibility>

NV Access. (2018). NV Access - What is NVDA. Retrieved February 24, 2018, from <https://www.nvaccess.org/>

Sambhanthan, A., & Good, A. (2012). Implications for Improving Accessibility to E-Commerce Websites in Developing Countries. *International Journal of Knowledge-Based Organizations*, 2(2), 1–20. <https://doi.org/10.4018/ijkbo.2012040101>

Sartoretto, Mara; Bersch, R. (2017). Tecnologia Assistiva. Retrieved February 12, 2018, from <http://www.assistiva.com.br/tassistiva.html>

Schenker, M. (2016). Why and How to Improve Ecommerce Website Accessibility. Retrieved January 28, 2018, from <https://www.shopify.com/partners/blog/why-and-how-to-improve-ecommerce-website-accessibility>

Shneiderman, B. . (2016). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*.

Sohaib, O., & Kang, K. (2016). Assessing web content accessibility of E-commerce websites for people with disabilities. In *25th International Conference on Information Systems Development, ISD 2016* (pp. 466–475). Retrieved from <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1135&context=isd2014>

Staveley, A. (2011). The “4+1” View Model of Software Architecture - DZone Java. Retrieved February 25, 2018, from <https://dzone.com/articles/41-view-model-software>

Stevens, J. (2016). How to Make Websites User Friendly and Accessible for Everybody - HostingFacts.com. Retrieved February 12, 2018, from <https://hostingfacts.com/website-accessibility-guide/>

The World Bank. (2017). Disability Inclusion. Retrieved February 12, 2018, from <http://www.worldbank.org/en/topic/disability>

Tonolli, José Carlos; Bersche, R. (2006). Introdução ao Conceito de Tecnologia Assistiva e Modelos de Abordagem da Deficiência. | Bengala Legal. Retrieved February 18, 2018, from <http://www.bengalalegal.com/tecnologia-assistiva>

U.S. Department of Justice. (2017). Introduction to the ADA. <https://doi.org/10.1145/1653616.1647422>  
United States government. (2008). Official Guide to Government Information and Services | USAGov. Retrieved February 25, 2018, from <https://www.usa.gov/>

United States Social Security Administration. (2016). The United States Social Security Administration. Retrieved February 24, 2018, from <https://www.ssa.gov/>

University of Waterloo. The Principles of Inclusivity. Retrieved December 12, 2018, from <https://uwaterloo.ca/organizational-human-development/staff-workshops/inclusivity-ecertificate/principles-inclusivity>

W3C. ARIA19: Using ARIA role=alert or Live Regions to Identify Errors | Techniques for WCAG 2.0. Retrieved October 6, 2018, from <https://www.w3.org/TR/WCAG20-TECHS/ARIA19>

W3C. (2008). Web Content Accessibility Guidelines (WCAG) 2.0. Retrieved February 13, 2018, from <https://www.w3.org/TR/WCAG20/>

W3C. (2018). Web Content Accessibility Guidelines (WCAG) 2.1. Retrieved February 22, 2018, from <https://www.w3.org/TR/WCAG/#background-on-wcag-2>

WAI-W3C. (2012). Diversity of Web Users - How People with Disabilities Use the Web. Retrieved February 13, 2018, from <https://www.w3.org/WAI/intro/people-use-web/diversity>

Watson, L. (2005). What is a screen reader? | Humanising Technology Blog. Retrieved February 14, 2018, from <https://www.nomensa.com/blog/2005/what-screen-reader>

Watson, L. (2009). How to write an accessibility statement. Retrieved from <https://www.nomensa.com/blog/2009/writing-an-accessibility-statement>

WC3. (2012). Social Factors in Developing a Web Accessibility Business Case for Your Organization ° Web Accessibility Initiative ° W3C. Retrieved February 18, 2018, from <https://www.w3.org/WAI/bcase/soc.html>

WebAIM. (2017). WebAIM: Screen Reader User Survey #7 Results. Retrieved February 24, 2018, from <https://webaim.org/projects/screenreadersurvey7/>

Wodehouse, C. (2017). How Human-Computer Interaction Helps Guide Better UI Design. Retrieved February 18, 2018, from <https://www.upwork.com/hiring/development/human-computer-interaction-ui-design/>